ANEXO I

ANEXO 1.1

Sistemas Automáticos de Medición de Volúmenes

a. Sistema de captura de datos de paquetes estáticos comerciales

DM3500-IR

El DM3500 [5] es una unidad compacta de dimensionamiento que mide automáticamente la longitud, el ancho y altura de paquetes a medida que se transportan sobre una cinta transportadora a través de la superficie de lectura de código de barras. La unidad de dimensionamiento autónoma está diseñada para la instalación simple a través de los sistemas de transporte existentes. En la figura 1.3 se observa la máquina de medición descrita.

- Exactitud: 0.25cm (altura), 0.51cm (ancho y largo)
- Velocidad de faja: 0.61m/s
- Rango de medición: 152.4cm (ancho), 373.38cm (largo) y 91cm (alto).
- Tecnología: arreglo de cámaras CCD y diodos laser
- Orientación: No es necesario orientar el objeto
- Interface: RS232 y Ethernet (TCP/IP)
- Tipo de sistema: dinámico

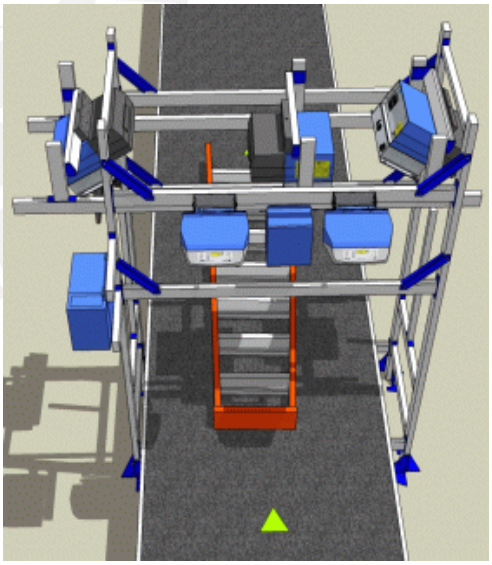


Fig.1.3. *Dispositivo de mediciónDM3500-IR de ACCU-Sort System.*

CSN910 FlexFlow™

La figura 1.4 muestra este dispositivo que representa una alternativa de solución para la captura de datos no atendida que escanea la identificación y mide las dimensiones de paquetes que se desplazan sobre un transportador. Posee una velocidad de medición y capacidad de procesamiento adecuadas, control total del contaje de paquetes y una correcta clasificación y manipulación del material durante de todo el proceso [6].

***Fig.1.4**. Máquina de medición CS910 FlexFlow de Metler Toledo*.

- Exactitud: 2mm (altura), 5mm (ancho y largo)
- Velocidad de faja: 3m/s
- Rango de medición: 120cm (ancho), 2500cm (largo) y 700cm (alto).
- Tecnología: laser
- Rendimiento: 15000 objetos por hora
- Interface: RS232 y Ethernet (TCP/IP)
- Orientación: No es necesario orientar el objeto
- Tipo de sistema: dinámico

TLX MultiCapture™

Es una solución global de captura de datos en movimiento diseñada para su integración en los sistemas de cinta transportadora de manipulación de paquetes guiada o no guiada. Los paquetes son separados automáticamente si la distancia entre ellos no garantiza la medición, pesaje e identificación de los mismos. El sistema está disponible en dos versiones; una totalmente automática y otra con opción de validación donde el operario puede introducir los datos que faltan del paquete [7], La figura 1.5 ilustra el sistema de medición descrito

***Fig.1.5.** Dispositivo de medición TLX MultiCapture.*

Entre sus especificaciones tenemos:

- Precisión: +/-2 mm altura +/- 5mm anchura y longitud
- Tamaño máx. del objeto: 1200 x 900 x 900 mm
- Tamaño mín. del objeto: 150 x 50 x 25 mm
- Rendimiento: 90 m/min

CSN210 MassFlow™

Es un sistema de vigilancia para adquirir automáticamente las dimensiones y la identificación de las mercancías movilizadas en un transportador. El haz de luz paralelo explora sobre cada elemento, captando todos los detalles, sin riesgo de sombreado. El diseño es modular, por ende, significa que el sistema se puede integrar completamente con escalas en movimiento y lectores de código de barras para adaptarse a las aplicaciones más exigentes de manipulación de embalaje [8]. Una ilustración del sistema se observa en la figura 1.6.

Fig.1.6.*Sistema de Medición CSN210 MassFlow*

Entre sus especificaciones tenemos:

- Precisión: ± 5 mm
- Tamaño máx. del objeto: 3000 x 1800 x 920 mm
- Tamaño mín. del objeto: 50 x 50 x 25 mm
- Velocidad de la faja: 1.3 m/s [14]

CubiScan 200TS

La ventaja más representativa de este dispositivo es la integración en movimiento, dimensionamiento y sistema de pesaje para maximizar el rendimiento y la eficiencia de los procesos de la industria. Usa tecnología de luz infrarroja, una representación gráfica de esta máquina se observa en la figura 1.7 [9].

- Exactitud: 5mm
- Velocidad de faja: 0.5m/s a 1.16m/s
- Rango de medición: 153cm (largo), 122cm (ancho) y 92cm (alto).
- Tecnología: sensores infrarrojos.
- Interface: RS232.
- Orientación: No es necesario orientar el objeto.
- Tipo de sistema: Dinámico

Fig. 1.7. Dispositivo de medición CubiScan200TS

b. Sistema de captura de datos de paquetes estáticos comerciales

ExpressCube 265R

La capacidad de medición y pesaje de este sistema la hace ideal para el comercio electrónico y aplicaciones de transporte de carga. La figura 1.8 brinda una representación gráfica de tal dispositivo [10].

- Exactitud: 5mm
- Rango de medición: 60cm (largo), 65.5m (ancho) y 94cm (alto).
- Peso máximo: 70kg
- Interface: RS232 y USB
- Tipo de sistema: estático

Fig. 1.8. Dispositivo de medición ExpressCube 265R

CSN810 TableTop™

Está diseñado para medir las dimensiones de objetos rectangulares colocados sobre una superficie nivelada. Se monta en alto y ahorra, así, espacio en el suelo; exige poca reorganización, o ninguna en absoluto, de las operaciones existentes y permite moverse sin impedimentos por debajo de él. En la figura 1.9 se observa el dispositivo descrito [11]

Fig.1.9. *Máquina de medición CSN810 TableTop*

Entre sus especificaciones tenemos las siguientes:

- Precisión: ±5mm
- Tamaño máx. del objeto: 1200mm x 900mm x 900mm
- Tamaño mín. del objeto: 50mm x 50mm x 50mm

ANEXO 1.2

Variables Inmersas en la Operación de los Sistemas Automáticos de Medición de Volúmenes.

1.2.1. Variables Externas.

La figura 1.10 muestra las variables externas que ejercen influencia y son afectadas por el desarrollo del sistema de medición, tanto en el nivel organizacional, especifico y general.

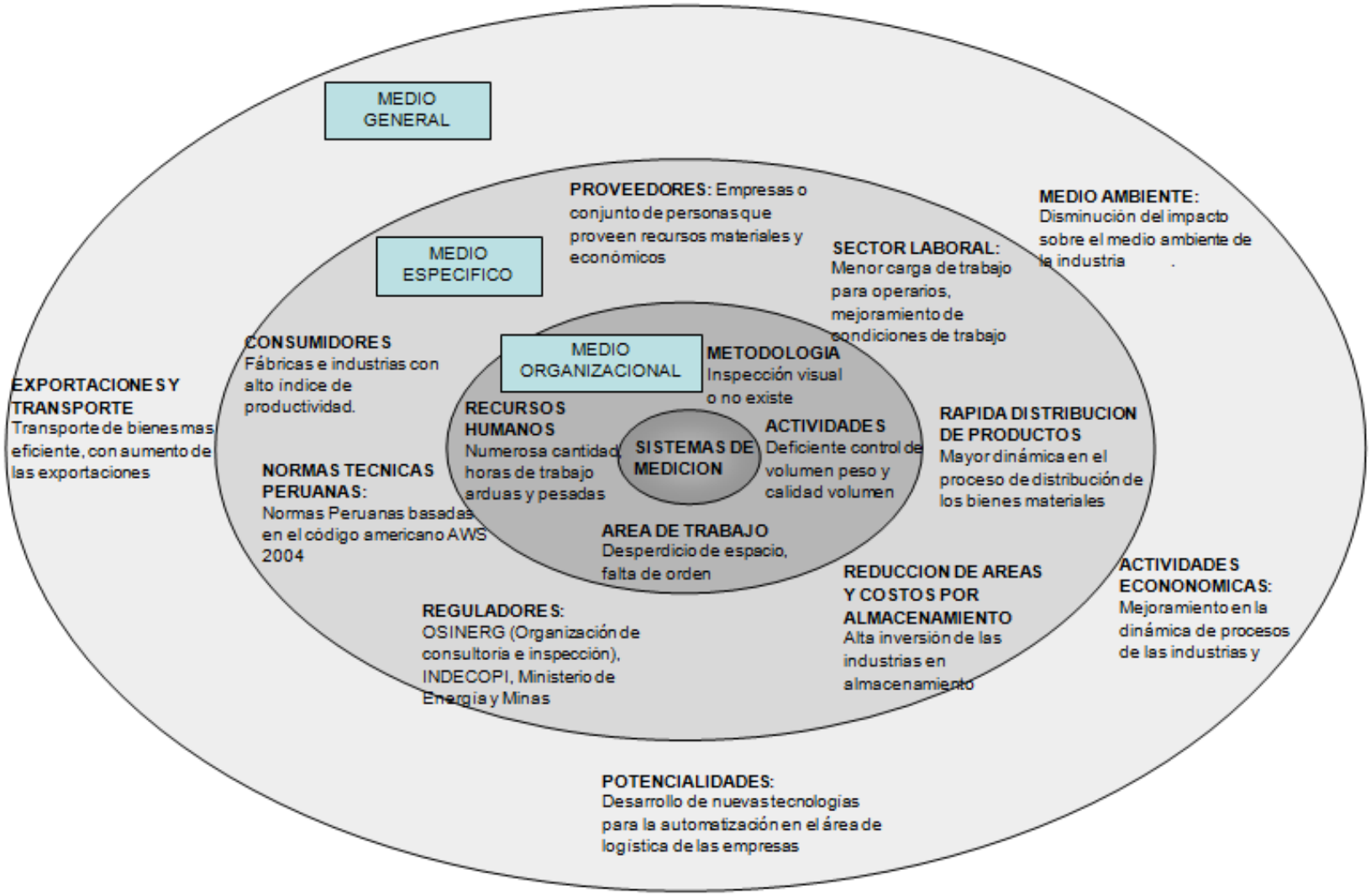


Fig. 1.10. *Variables Externas que afectan el desarrollo del sistema a implementarse.*

1.2.2. Variables Internas

El diagrama de flujo de los procesos de almacenamiento, empaquetamiento y transporte de productos se muestra en la figura 1.11, donde además se señalan las variables internas que afectan sus desarrollos.

Asimismo la tabla 1.1 brinda mejor información y explicación de cómo las variables internas identificadas anteriormente ejercen influencia en ciertas actividades de los procesos mencionados en el párrafo anterior.

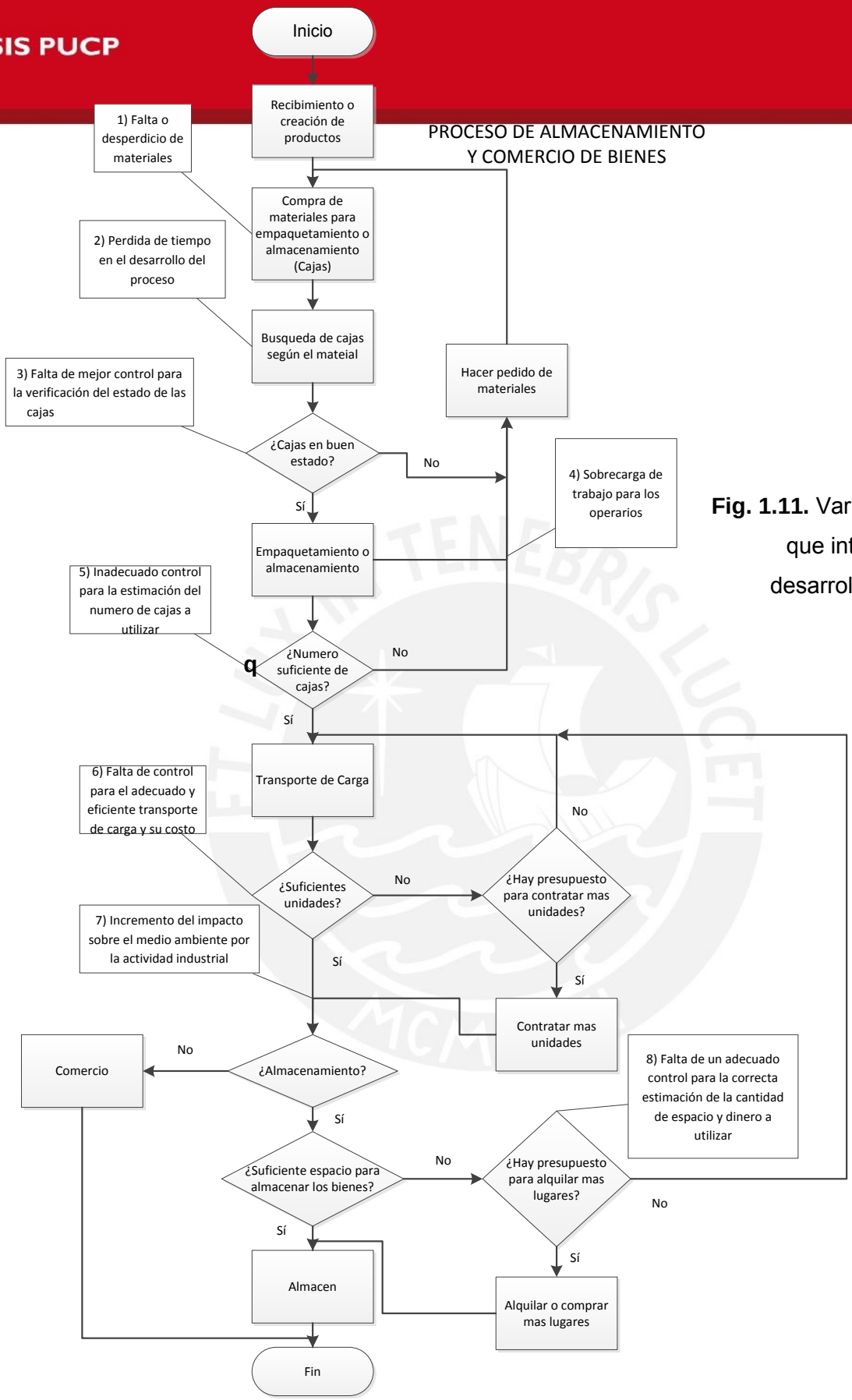


Fig. 1.11. Variables internas que intervienen en el desarrollo del sistema.

Tabla 1.1. *Descripción de los problemas originados por variables internas.*

HECHOS	PROBLEMAS Y CAUSAS
1) Falta o desperdicio de materiales.	La falta de un adecuado sistema de control para el dimensionamiento de los productos y la cantidad de ellos, produce un aumento en el costo de compre de materiales y pérdida de tiempo en el flujo de procesos de la empresa.
2) Pérdida de tiempo en el desarrollo del proceso.	En el proceso de empaquetamiento de los productos, los operarios pierden mucho tiempo buscando las cajas adecuadas para ellos, más aun si lo hacen sin alguna clasificación anterior, lo que genera retraso e interrupción en el desarrollo del proceso.
3) Falta de control para verificar el estado de las cajas.	Sin un sistema de monitoreo de las dimensiones de los cajas, los productos serán mal empaquetados y pueden ser dañados, originando baja calidad de éste ante los compradores.
4) Sobrecarga de trabajo para los operarios.	La búsqueda de los empaques adecuados para cada producto, el deterioro o falta de ellos será verificado por los trabajadores de manera directa, lo que produce sobrecarga en su labor, aumento de las jornadas, que se vuelven arduas y rutinarias. Esto, a su vez, causa estrés y fatiga en los operarios, lo que constituirá en la disminución de la productividad de la empresa.
5) Inadecuado control para la estimación del número de cajas a utilizar.	Ineficiente control para el cálculo de las cajas a utilizar significará el aumento de costo del material y mano de obra, así como retardo en el desarrollo del proceso.
6) Falta de control para el adecuado y eficiente transporte de la carga y su costo.	Sin la implementación de un sistema de medición y peso de los productos, originará perdidas en la economía de la empresa, por la contratación de nuevas unidades para cumplir con el requerimiento, o la interrupción del flujo del proceso si no hay unidades de transporte
7) Incremento del impacto sobre el medio ambiente por la actividad industrial.	El aumento de las unidades de transporte causara incremento de contaminación del medio ambiente, por la liberación de gases, combustión, ruido, etc.
8) Falta de un adecuado control para la correcta estimación de la cantidad de espacio y dinero a utilizar.	La falta de implementación de un sistema de medición originara en ineficiente uso de los espacios asignados para el almacenamiento e incremento de costos por la compra o alquiler de nuevas zonas.

Fuente: Elaboración propia

Sensor ultrasónico: U-GAGE™ S18U **Banner Engineering**

- Rango de censado: 30 a 300 mm
- Tiempo de respuestas: 2.5ms o 30 ms
- Salida: 4-20mA
- Alimentación: 10 a 30V dc - 65mA máx.
- Frecuencia ultrasónica: 300 kHz.
- Resolución: 1mm

Figura A.7. *Sensores ultrasónicos*

Diodo Infrarrojo GP2Y0A21YK0F: **Sharp**

- Medición de distancias
- Rango de censado: 10 a 80 cm
- Alimentación: 4.5 a 5.5v – 30mA máx.
- Salida: 0.5v a 2.25v no lineal
- Tiempo de respuesta: 50ms

Figura A.8. *Sensor infrarrojo.*

Sensor Fotoeléctrico: Q12AB6FF50 **Banner Engineering**

- Modo operación: difuso
- Lógica: PNP o NPN (detector de presencia)
- Rango de censado: hasta 50 mm
- Diámetro del haz de luz: 0.5 mm @ 16 mm
- Alimentación: 10 a 30V dc 20Ma máx.
- Tiempo de respuestas: 700us

Figura A.9. *Sensor fotoeléctrico.*

Sensor Óptico Modelo GP2Y0D810Z0F SHARP

- Utiliza un diodo emisor infrarrojo y un fotodiodo.
- Voltaje de alimentación: 2.7 a 6.2V.
- Rango de detección: 80 a 130mm.
- Voltaje de salida digital: nivel bajo 0.6V como máximo y nivel alto igual al voltaje de alimentación menos 0.6V como mínimo.
- Tiempo de respuesta: máximo 5.65ms, típico 3.84ms

Figura A.10. *Sensor fotoeléctrico.*

MARCAS COMERCIALES DE MICROCONTROLADORES

❖ **ATmega 16 (*Atmel*)**

- 131 instrucciones (Arquitectura RISC)
- 32x8 registros de propósito general
- 16 KBytes de Memoria Flash dentro del sistema
- 512 Bytes de Memoria EEPROM
- 1Kbyte de SRAM

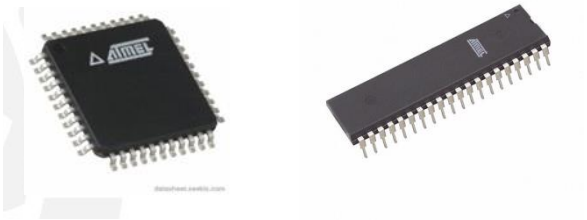


Figura A.11. *Empaques ATmega*

- 2 Temporizadores/Contadores de 8 bits con pre-escaladores y comparadores separados,
  1Temporizadores/Contadores de 16 bits con pre-escalador, modo de comparación y captura.
- 4 Canales PWM
- 8 Canales ADC de 10 bits
- Comunicación serial USART
- 32 Puertos de I/O programables

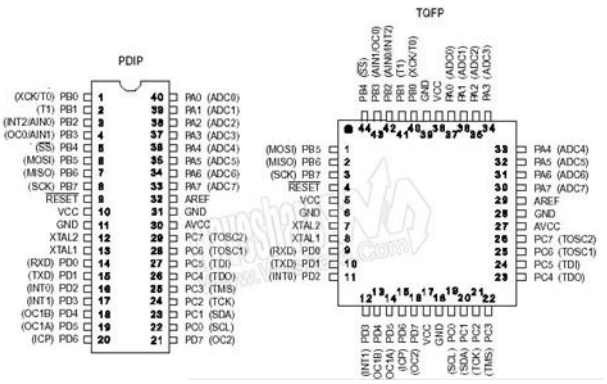


Figura A.12.*Distribución de los pines.*

- Voltaje de operación: 4.5V – 5.5V para ATMega16, 2.5V – 5.5V para ATMega16L
- Grado de velocidad: 0 – 16Mhz para ATMMega16, 0 – 8Mhz para ATMega16L

❖ **PIC 16F877 (*Microchip*)**

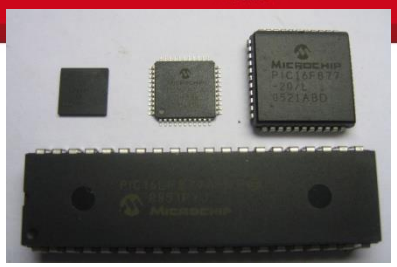


Figura A.13.*Empaques de PIC's.*

- 35 instrucciones (Arquitectura RISC)
- 8K x 14 palabras de Memoria FLASH.
- 368 x 8 Bytes de Memoria RAM
- 256x 8 Bytes de Memoria EEPROM
- 2 Temporizadores/Contadores de 8 bits con pre-escaladores y comparadores separados.
- 1 Temporizadores/Contadores de 16 bits con pre-escalador, modo de comparación y captura.
- 8 Canales ADC de 10 bits
- Comunicación serial USART

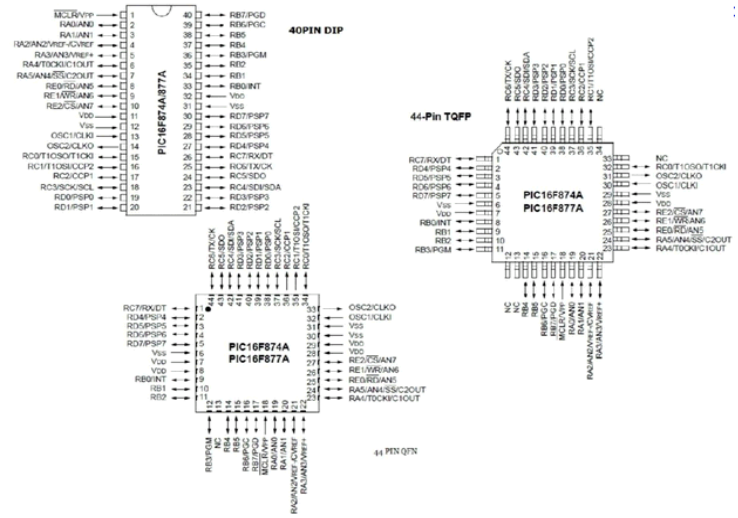


Figura A.14.*Distribución de los pines de un PIC*

ANEXO 3

ANEXO 3.1.1

Cálculo del torque:

Para hallar este parámetro se requerirá una carga máxima sobre la faja cuando ésta se encuentre en movimiento, es decir, se hallará el torque necesario para la carga máxima. Se aplicará el concepto de torque o momento de una fuerza respecto a un punto de referencia. Se hallará el torque necesario para mover la polea principal, para esto se hará uso de un bloque cuya masa es de 480gramos que se suspenderá a 17cm de la polea principal de la faja transportadora como se muestra en la figura 3.3.

$$Torque = Fuerza\ x\ Distacia$$

$$Fuerza = Peso = (masa)\ x\ (gravedad) = 0.48\ kg\ x\ 9.81\ m/s2$$

$$Torque\ necesario = 4.7088\ N\ x\ 0.17\ m = \mathbf{0.8\ N.m}$$

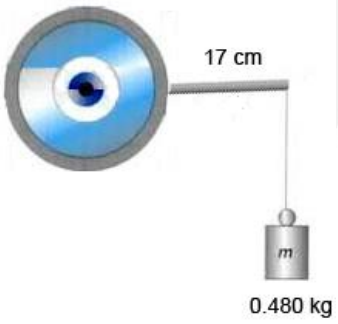


Figura 3.3. *Descripción gráfica del cálculo del torque.*

Cálculo de la velocidad de giro del motor (RPM):

Para el cálculo de la velocidad que deberá tener el motor, se tomará en cuenta la relación que existe entre la rapidez lineal y angular. La velocidad máxima a la que debería moverse la faja transportadora es de 0.7 m/s y el diámetro de la polea es 5 cm. Con estos datos podemos deducir lo siguiente:

$$Velocidad\ angular = \frac{Velocidad\ lineal}{radio}$$

$$RPM = \frac{Velocidad\ lineal}{radio}\ x\ \frac{30}{\pi} = \frac{Velocidad\ lineal}{diámetro}\ x\ \frac{60}{\pi}$$

$$RPM = \frac{0.7\ m/s}{0.05\ m}\ x\ \frac{60}{\pi} = 267.3796\ RPM \cong \mathbf{268\ RPM}$$

Cálculo de la potencia:

El cálculo de la potencia de salida requerida del motor se puede realizar conociendo los dos parámetros ya mencionados *[16]*.

$$Potencia\ mecánica\ útil = Torque\ x\ RPM\ x\ \frac{\pi}{30}$$

$$Potencia\ = 0.8\ N.m\ x\ 268\ RPM\ x\ \frac{\pi}{30} = \mathbf{22.45\ W}$$

ANEXO 3.1.2

Alternativas de solución para el actuador

En la tabla 3.1 presenta un cuadro que muestra la comparación de las características más importantes de un motor DC sin escobillas y un motor DC con escobillas.

Tabla 3.1. *Comparación entre motores DC sin y con escobillas.*

Característica	***Motor DC sin escobillas*** 	***Motor DC con escobillas*** 
Conmutación	La conmutación es electrónica mediante sensores de efecto Hall.	La conmutación es por escobillas.
Mantenimiento	Mínimo	Periódico
Durabilidad	Mayor	Menor
Curva Velocidad - Par	Plana. Operación a todas las velocidades con carga definida.	Moderada. Se reduce el par debido al incremento de la fricción en las escobillas.
Eficiencia	Alta. No existe caída de tensión debido a la ausencia de escobillas.	Moderada.
Potencia - Tamaño	Alta. Mejor disipación de calor ya que los devanados están en el estator.	Moderada. El calor es disipado en el interior aumentando la temperatura.
Rango de velocidad	Alto. No existen limitaciones mecánicas por parte de las escobillas o el conmutador.	Baja. El límite se impone debido al uso de las escobillas.
Ruido eléctrico	Bajo.	Los arcos en las escobillas generan ruido causando EMI en equipos cercanos.
Control	Complejo y costoso.	Simple y barato.
Requisitos de control	Necesidad de un controlador para operar el motor y variar la velocidad.	Solo requiere un controlador si se desea variar la velocidad.

Adaptado del libro "Electrónica de Potencia: circuitos, dispositivos y aplicaciones" de Muhammad H, Rashid [28].

También se analizará la posibilidad de usar un motor AC de inducción trifásico pues el sistema podría ser utilizado en un ambiente industrial. La comparación de las características más resaltantes frente a un motor DC sin escobillas, se detalla en la tabla 3.2.

Tabla 3.2. *Comparación entre motores DC sin escobillas y motores AC de inducción.*

Característica	***Motor DC sin escobillas*** 	***Motor AC de inducción*** 
Curva velocidad y Torque	Plana. Permite operación a cualquier velocidad con carga nominal.	No lineal. A menor velocidad existe menor torque.
Potencia de salida	Alta. Es pequeño pues posee imanes permanentes en el rotor.	Moderada. Debido a los devanados en el estator y rotor.
Corriente de Arranque	Nominal. No es necesario un circuito especial de arranque.	Hasta 7 veces la nominal. Se debe contar con un circuito para el arranque.
Requisitos de control	Es necesario un controlador externo para ponerlo en marcha.	Solo se requiere un controlador si se desea variar la velocidad.
Deslizamiento	No sufre deslizamiento entre la frecuencia del estator y rotor.	Existe deslizamiento pues la frecuencia del rotor es menor que la del estator.

Adaptado de la nota de aplicación "Brushless DC (BLDC) Motor Fundamentals". Microchip Tecnology Inc. [28]

ANEXO 3.2.1

Alternativas de solución para el interruptor de potencia

Se tiene muchos tipos de transistores de potencia que cumplen de papel de interruptor de conmutación. Algunos de los más usados para este tipo de aplicaciones se muestran en la tabla 3.3, la cual ayudará a conocer un poco más de este tipo de interruptores y, de esta manera, elegir el más adecuado.

Tabla 3.3. *Comparación de interruptores de potencia.*

Característica	***BJT***	***MOSFET***	***IGBT***
			
Variable de control	Corriente	Voltaje	Voltaje
Frecuencia de Conmutación	Media (Hasta 25kHz)	Muy Alta (Hasta casi 1MHz)	Alta (Hasta 100kHz)
Especificaciones máximas de voltaje	Hasta 1.5Kv	Hasta 1kV	Hasta 3.5kV
Especificaciones máximas de corriente	Hasta 700 A	Hasta 100 A	Hasta 2kA
Ventajas	-Baja caída en estado encendido. -Mayor capacidad de voltaje en estado apagado.	-Baja pérdida de conmutación. -Poca potencia disipada en compuerta. -Circuito simple de control de compuerta.	-Poca potencia disipada en la compuerta. -Bajo voltaje en estado encendido.
Limitaciones	-Disipación de potencia en activación de base. -Requiere mayor corriente de base para sostener el estado activo. -Altas disipaciones de conmutación.	-Menor capacidad de voltaje en estado apagado. -Alta caída de voltaje en conducción.	-Menor capacidad de voltaje en estado apagado.

Adaptado de la nota de aplicación "Brushless DC (BLDC) Motor Fundamentals". Microchip Tecnology Inc. [28]

ANEXO 3.2.2

Aislamiento Eléctrico:

Existe una etapa digital que se caracteriza principalmente por la generación de la onda PWM con frecuencia y ciclo de trabajo que se define en un primer momento. Luego se tendrá la posibilidad de variar el ciclo de trabajo a través de una interfaz de entrada externa como el teclado alfanumérico logrando así variar el voltaje promedio en los terminales del motor y, por ende, la velocidad de giro del mismo.

La onda PWM será generada por un microcontrolador que se encontrará en una tarjeta impresa separada. A continuación, se usará un optoacoplador para aislar eléctricamente la etapa digital de la etapa de potencia mediante un diodo LED que satura o corta un fototransistor a través de emisión de luz. Para esto, cuando se quiera que el fototransistor se sature solo se debe hacer circular corriente por el diodo LED del optoacoplador y dejar de hacerlo cuando se quiere cortar al fototransistor.

Como la señal PWM tiene una amplitud de 0V cuando está en baja y 4.2V como mínimo cuando está en alta, se puede aprovechar estos estados para controlar la emisión de luz del diodo LED del optoacoplador tal como se muestra en la figura 3.5.

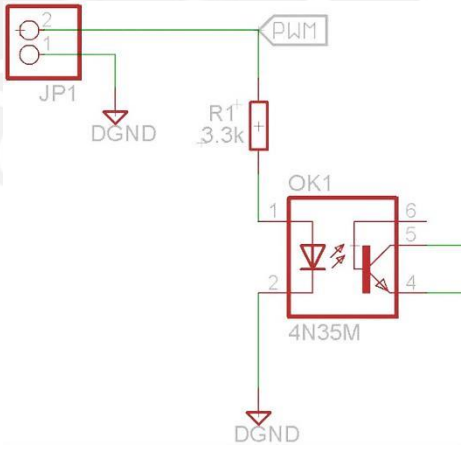


Figura 3.5. *Circuito esquemático del bloque digital del variador de velocidad.*

Cálculos:

El objetivo es generar una determinada cantidad de corriente para emitir luz al fototransistor del optoacoplador. Trabajaremos con los parámetros sugeridos en la hoja de datos del optoacoplador. De toda la gama de optoacopladores comerciales se eligió el 4N35 pues posee un mayor porcentaje de CTR (*Current Transfer Ratio*) lo que nos proporciona mayor corriente de colector cuando el fototransistor se satura. Según la hoja de datos del 4N35, el valor típico de la caída de voltaje en el diodo LED V_F = 1.18V cuando circula a través de éste una corriente I_F =1mA. En base a esto podemos hallar R_1.

$$\mathbf{R1} = \frac{4.2V - 1.18V}{1mA} => R1 = 3.02\text{K}\Omega => \mathbf{R1}(\text{valor comercial}) = 3.3K\Omega$$

Con este valor de R1, se modifica ligeramente el valor de la corriente.

$$\mathbf{IF} = \frac{4.2V - 1.18V}{3.3K\Omega} => IF = 0.915\text{mA} \cong 1\text{mA}$$

ANEXO 3.2.3

Etapa de Potencia:

Se analizará los dos estados de la señal PWM, tanto en alta como en baja para ver el comportamiento de algunos de los componentes involucrados y poder calcular sus valores.

Bloque de Acondicionamiento de la variable de control

- *Cuando la señal PWM está en 5V*:

En este estado el fototransistor se satura y su voltaje colector-emisor es igual a 0.3V. Luego debemos acondicionar la variable de control, es decir, el voltaje pues se usará un MOSFET como interruptor de potencia. El objetivo principal es que el voltaje que llegue al pin *gate* sea aproximadamente 12V. En el bloque de conmutación se explicará el motivo.

Se debe aprovechar la saturación del fototransistor para que diseñar un circuito que genere 12V al pin *gate* del MOSFET. Para esto se usará una distribución de transistores y resistencias que cumplan el objetivo. En la figura 3.6 se propone el uso de un transistor PNP (Q1) que se saturará cuando el fototransistor también lo haga. La corriente máxima que se puede circular por el colector del fototransistor es aproximadamente 10mA pues el optoacoplador posee un 100% de CTR. El CTR es la relación de transferencia de corriente entre el diodo LED y el fototransistor.

$$\text{CTR} = \frac{Ic}{Id} \text{x } 100\%$$

Para saturar el transistor Q1 debemos de cumplir algunos requisitos. Para esto, se debe recurrir a la hoja de datos del transistor 2N3906 para establecer los parámetros eléctricos de saturación.

- Voltaje de Saturación Colector Emisor:
  - ✓ Ic = -10mA, Ib = -1mA @ típico -0.25V
  - ✓ Ic = -50mA, Ib = -5mA @ tipico-0.4V
- Voltaje de Saturación Colector Emisor:
  - ✓ Ic = -10mA, Ib = -1mA @ mínimo -0.65V
  - ✓ Ic = -50mA, Ib = -5mA @ típico -0.95V

Entonces, se fijará la corriente del colector del fototransistor en 1mA que a su vez será la corriente de base de transistor Q1.

Tomando los valores máximos de saturación V_{EC} = 0.25V y V_{EB} = 0.85V @ I_C = 10mA y I_B = 1mA; y el voltaje de saturación V_{CE} = 0.3V del fototransistor, se hallan las resistencias R_2, R_3 Y R_4.

$$\mathbf{R2} = \frac{12V - 0.85V - 0.3V}{1mA} = 10.85\text{k}\Omega \Rightarrow \mathbf{R2}(\text{valor comercial}) = 10\text{k}\Omega$$

Con este valor de R_2 se puede hallar la corriente real que circula por el colector del fototransistor.

$$\mathbf{Ic} = \frac{12V - 0.85V - 0.3V}{10k\Omega} = 1.085\text{mA} \approx 1\text{mA}$$

La resistencia R_3 es necesaria debido a que existe diferencia de potencial entre sus terminales cuando el transistor Q1 se satura. La corriente que se genere debido a esta diferencia de voltaje debe ser aproximadamente igual en comparación a la corriente que circula por el colector de fototransistor. Se asumirá que esta corriente aproximadamente 0.9mA ≅ 1mA. De esta manera, solo exigiremos una corriente de base del transistor Q_1 I_B de 0.1mA. Según este criterio se hallará la resistencia R_3.

$$\mathbf{R3} = \frac{12V - (12V - 0.85V)}{0.9mA} = 0.94\text{k}\Omega \Rightarrow \mathbf{R3}(\text{valor comercial}) = 1\text{k}\Omega$$

La corriente real será:

$$\mathbf{I} = \frac{12V - (12V - 0.85V)}{1k\Omega} = 0.85\text{mA}$$

Para hallar R_4 se tomará en cuenta la corriente del colector de Q1 en saturación I_{CQ1}= 1mA ya que I_B = 0.1mA; de esta manera cumplimos la condición I_{CQ1} = 10I_B para garantizar la saturación. Entonces se tendrá la siguiente ecuación:

$$\mathbf{R4} = \frac{12V - 0.25V}{1mA} = 11.75\text{k}\Omega \Rightarrow \mathbf{R4}(\text{valor comercial}) = 12\text{k}\Omega$$

La corriente real no variará tanto en comparación a la asumida:

$$\mathbf{IC} = \frac{12V - 0.25V}{12k\Omega} = 0.97916\text{mA} \approx 1\text{mA}$$

Luego de la etapa del transistor PNP, debemos de llevar la señal de 12V hacia la entrada del MOSFET, para esto se tiene un diodo de conmutación de alta velocidad que conducirá cuando el transistor Q1 se sature. En la hoja de datos del diodo 1N4148 se puede apreciar que el tiempo que demora en cambiar de modo de conducción a modo de corte. Alguna de éstas y otras características son las siguientes.

- T_{rr}= 4ns (Tiempo máximo de cambio de estado)
- V_F= 0.72V @ I_F = 5mA (Ver gráfica para mayores detalles)
- $I_{FMÁX}$= 200mA
- V_R = 75V (Voltaje continuo máximo en inversa)

Debido a la alta impedancia de entrada del MOSFET, se asume que la corriente que circula a través del diodo es mucho menor en comparación a la que circula por el colector del transistor Q1 ya además que es la necesaria para que el diodo conduzca.

Finalmente la señal que llega al pin *gate* del MOSFET es un poco menos de 12V debido a las pequeñas caídas de tensiones; este valor sería aproximadamente:

$$V = 12V - 0.25V - 0.72V = 11.03V$$

- *Cuando la señal PWM está en 0V*:

En este caso, el fototransistor se encuentra en estado de corte, pues no circula corriente a través del LED interno del optoacoplador. De la misma manera, al no tener un camino hacia un menor potencial por el cual circule la corriente de base del transistor Q1, se produce el estado de corte del mismo.

Bloque Interruptor de Potencia:

- *Cuando la señal PWM está en 5V*:

Esta etapa consta básicamente de la excitación de la puerta de un MOSFET a través de voltaje aplicado en ese terminal. El voltaje umbral para la conducción del MOSFET es $V_{GS\ (TH)}$= 4.0V y admite un voltaje máximo de 20.0V por dicho terminal. Sin embargo, el voltaje que se aplica a la puerta es 11.03V aproximadamente debido a que la resistencia dinámica interna del MOSFET entre los terminales drenador y surtidor R_{DS} disminuye a medida que se aumenta en voltaje aplicado en la puerta V_G.

Con un $V_G > V_{GS\ (TH)}$, aseguramos que la caída de voltaje en los terminales drenador y surtidor sea el menor posible y que el motor reciba el voltaje completo.

MOSFET Canal N Activo:

V_{DD} = 20V, Voltaje de control (V_G) = 11.03V > $V_{GS\ (TH)}$ = 4.0V $R_{(MOTOR)}$ = 7Ω

Si V_{GS}= 10V (valor más cercano a 11.03V) → De las hojas de datos del MOSFET se obtiene: $R_{DS\ (ON)}$ = 15mΩ y la relación $I_{DS} = 50V_{DS}$ en el supuesto caso que el MOSFET se encuentre en la región óhmica cuando se active. De la figura 3.7 se puede deducir dicha relación.

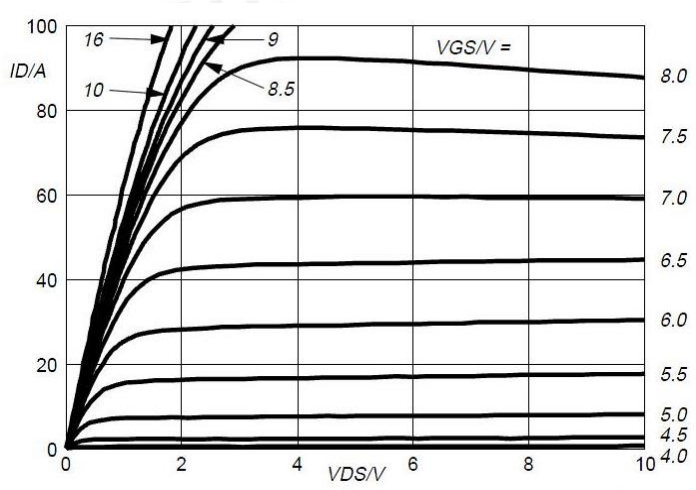


Figura 3.7. *Gráfica V_{DS} vs I_{DS} del MOSFET IRFZ44N.*

Se toma la parte lineal de la curva situada entre V_{DS} [0 - 1.6V] e I_D [0 - 80A] para plantear la siguiente ecuación:

$$\mathbf{Id} = 50\mathbf{Vds}$$

Reemplazando V_{DS} = (0.02)*I_{DS} en la siguiente ecuación del circuito proveniente de la figura 3.7.

$$\mathbf{Ids} = \frac{20V - Vds}{7\Omega + 15m\Omega}$$

$$\mathbf{Ids} = 2.84\mathrm{A}$$

De la misma relación podemos encontrar el V_{DS}:

$Vds = (0.02)Vds \Rightarrow Vds = 0.02 \times 2.84 = 0.0568V$

Luego aplicamos la condición:

$$\mathbf{Vds} < Vgs - Vgs(th)$$

$$0.0568\text{V} < 11.03V - 4.0V \rightarrow 0.0568V < 7.3V$$

Se verifica entonces que el MOSFET se encuentra en la REGIÓN ÓHMICA.

- *Cuando la señal PWM está en 0V*:

El voltaje de control en la puerta del MOSFET V_G= 0V. Entonces se tiene:

$$\boldsymbol{Vgs} < Vgs(\boldsymbol{th})$$

$$0V < 2.3V \rightarrow \boldsymbol{Ids} = 0A$$

Entonces el MOSFET se encuentra en la REGIÓN DE CORTE.

Debemos de considerar los tiempos transitorios de conmutación en el MOSFET y asegurar que sean los más bajos posibles *[32]*. Se analizará el tiempo de conmutación de apagado del MOSFET en 2 momentos significativos:

- Cuando el MOSFET tiende a cambiar de estado pero aún sigue activo.

$$\boldsymbol{\tau 1} = Rg(Cgd1 + Cgs)$$

C_{GD}: Capacitancia entre los terminales puerta y drenador $V_{DS\ (ON)}$= 56.8mV

De las gráficas proporcionadas en las hojas de datos se puede obtener los siguientes valores:

Ciss = C_{DG} + C_{GS}, (El valor de C_{DG} depende del voltaje V_{DS}) →Ciss = 2250pF

$$\boldsymbol{\tau 1} = 22\Omega.(2250pF) = 49.5ns$$

- Cuando el MOSFET llega a la zona de corte y V_{DS} = 20.0V

Ciss = C_{DG} + C_{GS}→ Ciss = 1400pF

$$\boldsymbol{\tau 2} = 22\Omega.(1400pF) = 30.8ns$$

El tiempo de total de transición en apagado es: τ1+ τ2 = 80.3ns

Este tiempo de apagado que emplea el MOSFET se puede considerar despreciable en comparación al periodo de la señal de control del terminal *gate* de 10 ms. Para este caso, el MOSFET podrá cambiar de estado completamente tomando sus respectivos tiempos de establecimiento tanto para el apagado como para el encendido.

Bloque de Motor DC:

- *Cuando la señal PWM está en 5V*

Debido a que el MOSFET se encuentra en estado de conducción, el motor DC recibe su voltaje nominal entre sus terminales y empieza a girar. Como la variación de su velocidad es por PWM, el motor recibe un voltaje promedio en relación al ciclo de trabajo de la señal PWM proporcionada por el microcontrolador.

- *Cuando la señal PWM está en 0V*:

El MOSFET se encuentra en estado de corte y no permite el flujo de corriente a través del motor. El motor tiende a parar su marcha. Además, existe otro efecto relacionado con la activación de cargas inductivas por conmutación de interruptores. Cuando se corta el flujo de corriente del motor en un tiempo muy corto, las bobinas tienden a generar voltajes muy elevados como respuesta a esa acción. Este efecto puede ocasionar daños al interruptor de potencia.

Para liberar la energía atrapada en las bobinas, se coloca un diodo de efecto volante o *"Free-Wheeling Diode"* el cual entra en estado de conducción cuando el MOSFET se encuentre en estado de bloqueo y crea un camino por el cual descargar la corriente almacenada en las bobinas y llevarla hacia un condensador mostrado en el circuito esquemático. En la figura 3.8 se explica de gráficamente el efecto mencionado anteriormente.

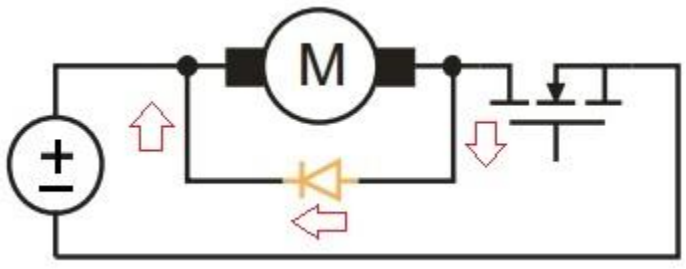


Figura 3.8.*Acción del diodo volante para cargas inductivas [33].*

Cálculo del disipador:

Para poder calcular el disipador necesario para esta aplicación, es necesario calcular las pérdidas totales del MOSFET operando en conmutación*[34]*.

En primer lugar se calculan las *pérdidas en conducción*, es decir, cuando el MOSFET esté cerrado y, por lo tanto haya circulación de corriente. Para este caso se considera que el MOSFET se comporta como una resistencia de valor $R_{(ds)ON}$

$$\boldsymbol{Pc} = Id^2 * D * Rds(on)$$

$$\boldsymbol{Pc} = (2.85)^2 * 100\% * 15m\Omega => \boldsymbol{Pc = 0.122\,W}$$

De la misma manera se toma en cuenta las *pérdidas en conmutación* que se producen cuando el semiconductor pasa del estado de bloqueo o corte a conducción y viceversa.

$$\boldsymbol{Psw} = \frac{1}{2} * Vd * Io * f * (tr + tf)$$

$$\boldsymbol{Psw} = \frac{1}{2} * 20 * 2.85 * 100 * (75ns + 40ns) => \boldsymbol{Psw = 0.32mW}$$

De esta manera, la potencia total consumida en el MOSFET será:

$$\boldsymbol{Pt} = Pc + Psw => \boldsymbol{Pt = 0.12232W}$$

Finalmente procedemos al cálculo del disipador para el MOSFET con empaque TO-220.

$$R\Theta ja = R\Theta jc + R\Theta cs + R\Theta sa$$

Donde:

$$R\Theta ja = \frac{Tj - Ta}{Pt} = \frac{100 - 25}{0.12232} = 613.14ºC/W$$

Si se asume $R\Theta_{JC}$ = 1.5ºC/W y $R\Theta_{CS}$ = 0.5ºC/W

$$R\Theta sa = R\Theta ja - R\Theta jc - R\Theta cs => \boldsymbol{R\Theta sa} = 613.14 - 1.5 - 0.5$$

$$\boldsymbol{R\Theta sa = 611.14ºC/W}$$

Este resultado sugiere usar un disipador que posea una resistencia térmica no mayor a 611.14ºC/W. Debido a la poca potencia que disipa el MOSFET, se requiere un disipador con una resistividad térmica muy alta, lo que significa que cualquier disipador que posea una resistividad menor a la mencionada puede ser usado. Se tomará como referencia el disipador 6296B de la empresa AAVID THERMALLY con H = 50.8mm y resistencia térmica 3.6ºC/W mostrado en la figura 3.9 *[35]*.

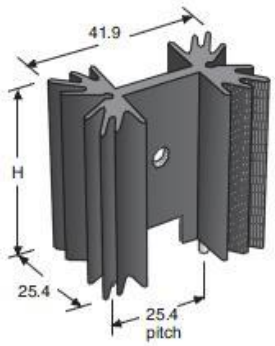


***Figura 3.9**. Disipador vertical para empaque TO-220.*

Anexo 3.3.1

Cálculo del tiempo de respuesta máximo para la detección de 5mm:

$$T= \frac{Precisión}{Velocidad} = \frac{5mm}{0.7m/s} = 7.14ms \text{ @ 0.7m/s}$$

$$T_{máx} = 7.14ms$$

Cálculo frecuencia mínima de muestreo del sensor de presencia para realizar mediciones con precisión de 5mm: 280.12 Hz

$$f = \frac{1}{T\max} = \frac{1}{7.14ms} = 140.06hz \qquad fs \geq 2f \Rightarrow fs \geq 280.12Hz$$

Tiempo de respuesta máximo del sensor de presencia para realizar mediciones con precisión de 5mm de una dimensión de una caja:

$$tr = \frac{1}{F\max} = \frac{1}{280.12Hz} = 3.57ms$$

Cálculo del tiempo de respuesta máximo para la detección de 1mm:

$$T= \frac{1mm}{0.7m/s} = 1.42ms \text{ @ 0.3m/s, entonces elegimos}$$

Cálculo frecuencia mínima de muestreo del sensor de presencia para realizar mediciones con precisión de 1mm: 280.12 Hz

$$f = \frac{1}{T\max} = \frac{1}{1.42ms} = 704.23hz \qquad fs \geq 2f \Rightarrow fs \geq 1408.46Hz$$

Tiempo de respuesta máximo del sensor de presencia para realizar mediciones con precisión de 1mm de una dimensión de una caja:

$$tr = \frac{1}{F\max} = \frac{1}{1408.46Hz} = 0.71ms$$

Cálculo frecuencia mínima de muestreo del ADC y sensores de distancia:

$$tm = \frac{Longitud.Minima.Caja}{Velocidad}$$

$$\frac{5cm}{0.7m/s} = 71.43ms \qquad \frac{5cm}{0.3m/s} = 166.67ms$$

$$fs \geq 2\,fm \Rightarrow fs \geq 2x\frac{1}{71.43ms} \Rightarrow fs \geq 28hz$$

Cálculo de sensibilidad del ADC y número de bits necesarios:

Valor Teórico:

- Voltaje de Referencia 5V.
- 8 canales de ADC de 10 bits

Tenemos 20cm para un voltaje de 0 a 5V 1mm x $\frac{5V}{200mm} = 25mV$

1LSB=25mV

Número de bits necesarios para digitalizar la señal para alcanzar la precisión requerida (1mm):

1LSB=23.8mV= $= \frac{5V}{2^N} \Rightarrow N \geq 7.71$, entonces el mínimo número de bits necesarios para digitalizar la señal del sensor de distancia será 8.

Tabla 3.4. *Cuadro comparativo de alternativas de solución de Sensores de distancia*

Características	**Sensor Ultrasónico Modelo S18UUAR, 5 cables, 2m de longitud**	**Sensor Fotoeléctrico Modelo DT20 Hi**
Fabricante	Banner Engineering	Sensor Intelligence
Alimentación	10 a 30Vdc	10 – 30VDC
Parámetros de salida	4-20mA	4-20mA modo PNP o NPN
Tiempo de respuesta	2.5ms , cable negro conectado a 12Vcc (modo rápido de funcionamiento)	Mayor a 2.5ms y menor a 15ms
Rango de medición	30 a 300mm	3 a 60mm
Configuración y calibración	Modo manual y automática mediante TEACH	Modo manual
Linealidad [1]	2.5 ms response: ±1 mm 30 ms response: ± 0.5 mm	+/-2mm
Protección contra cortocircuito y polaridad inversa en la salida	Si	No
Inmunidad al ruido	Norma CE, diseño de equipos eléctricos dentro de ciertos límites de voltaje, compatibilidad electromagnética en las telecomunicaciones e inmunidad ante disturbios de esta clase.	Norma CE*[56]*
Frecuencia del ultrasonido	300khz @ 2.5ms	-
Precio	$261	$680

Adaptado de "U-GAGE™ S18U Series Sensors with Analog Output" de Banner™ [37] y "DT20 HiDistance Sensor "de Sensor Intelligence [38].

Anexo 3.3.3

Tabla 3.5. *Cuadro comparativo de alternativas de solución de Sensores de presencia.*

Características	Sensor Fotoeléctrico Modelo Q12AB6FF50	Sensor de Posición Fotodiodo AD230-8-TO52
Fabricante	Banner Engineering	PacificSilicon Sensor
Modo de operación	Fixed-field, bipolar, operación por brillo	Directo
Voltaje de Alimentación	10 – 30Vdc	Potencia disipada: 100mW@22°C
Tiempo de Respuesta	700us	Tiempo de subida: 0.18ns
Rango de Medición	0.5 - 5mm	0.23mm de área activa
Tamaño del haz de luz	0.5mm @ 16mm de distancia, 6.5mm @50mm de distancia	2mm
Señales de salida	- Apagado: 0V @ 10uA - Encendido: V_{sat}=1.45V @ 50mA	Corriente máxima de operación Desde 0.3nA (Corriente oscura máxima 1.5nA) hasta 1mA
Inmunidad al ruido	Norma CE, diseño de equipos eléctricos dentro de ciertos límites de voltaje, compatibilidad electromagnética en las telecomunicaciones e inmunidad ante disturbios de esta clase.	Energía equivalente de ruido: $10^{-14} W/Hz$
Circuito de Protección de la fuente	Contra polarización inversa y voltajes transitorios	No
Circuito de Protección de la Salidas	Contra pulsos falsos y cortocircuitos	No
Precio	$76	1: $63.84 5: $57.12 10: $52.08 50: $47.88

*Adaptado de "*WORLD-BEAM® Q12*" de Banner™ [39] y "DT20 Hi Distance Sensor "de Sensor Intelligence [40].*

Anexo 3.3.4

Diseño del circuito del acondicionamiento de señales del Sensor de Distancia (Sensor Ultrasónico)

El sensor posee 5 terminales como se observa en la figura 3.10, los detalles de las conexiones se especifican a continuacion:

- Cable marrón se conecta a la alimentacion del sensor, en este caso 12V
- Cable azul se conecta al terminal negativo de la fuente de alimentacion.
- Cable blanco es la salida del sensor, la carga se conecta entre este y el cable azul(terminal negativo)
- Cable negro se utiliza para definir la velocidad de operación del sensor, rapida conectada de 5 a 30V o lenta 0 a 2V. En este caso se utilizará en modo rapido conectado a 12V.
- Cable plomo utilizado para la configuracion automatica del sensor mediante modo TEACH.
- Cable de metal que es un escudo para la atenuacion del ruido y proteccion contra cortocircuitos y cambios de polaridad del sensor.

Según la hoja de datos de este dispositivo, las señales de salida del sensor son mejores cuando la carga total de resistencia es $R = \frac{V_{in} - 3}{0.02}\Omega$

- Corriente de salida del sensor: I_s = 4 a 20mA
- Resistencia de carga: $R = \frac{12 - 3}{0.02} = 450\Omega$

  Donde R= R3 + R4

- Voltaje de salida del sensor: Vs = I_s x 0.44KΩ $\Rightarrow$ Vs= 1.76 – 8.8V
- Elección de OPAMPS, en la tabla 3.6 se muestra las características más importantes de 2 OPAMPS que podrían utilizarse en la implementación del circuito de acondicionamiento.

Tabla 3.6. *Comparación de características más resaltantes de OPAMPS que se pueden emplear.*

Características	UA741C	LM324
Voltaje de Alimentación	+/-15V	+/-16V
Rango de voltaje de entrada en modo común	+/-15V	-0.3 a 28.3V
Desplazamiento de voltaje de entrada	1 a 6mV	típico 2mV, máximo 7mV
Ganancia en rechazo de modo común	70-90DB	($Rs = 10K\Omega$): 70DB
Voltaje de Salida	Mínimo +/-12V@, R_L = 10KΩ típico +/-14V Mínimo +/-12V@ R_L<= 10KΩ Mínimo +/-10V@, R_L= 2kΩ, típico +/-13V	Limite alto a V_{cc} = 30V, $RL = 2K\Omega$: 26V Limite bajo a V_{cc} = 5V, $RL = 2K\Omega$: típico 5mV, máximo 20mV
Máxima corriente de desplazamiento en la entrada	0.2Ma	9Na
Tiempo de Respuesta (*SlewRate*)	0.5V/µs	5V/µs

Adaptado de "LM124 LM224 - LM324 LOW POWER QUAD OPERATIONAL AMPLIFIERS" de On Semiconductor [41] y " A741, A741Y GENERAL-PURPOSE OPERATIONAL AMPLIFIERS" de Texas Instruments [42].

.

- Utilización de OP-AMP's LM324 como buffer para trasladar el voltaje al microcontrolador y no producir efecto de carga, elegido debido a las siguientes características:
- Voltaje de entrada al buffer LM 324: V_{in} = 0.22kΩ x Is $\Rightarrow$ V_{in} = 0.88V a 4.4V
- Diodos de conmutación rápida 1N4148 para la protección contra cortocircuitos, sobretensiones o inversión de polaridad del microcontrolador, posee las siguientes características: [45]
    - Voltaje de Ruptura: 100V
    - Máxima Corriente umbral continuo: 450mA
    - Voltaje Umbral a I_f = 5mA: mínimo 0.62V, máximo 0.72V

- Voltaje de entrada al microcontrolador: $V_{\mu c}$ = 0.88V a 4.4V
- Potencia MAXIMA disipada por las resistencias:

$$P_R = Is^2 xR \Rightarrow P_R = (20mA)^2 \times 0.22K\Omega = 88\text{mW}$$

- Control remoto de los Sensores ultrasónicos: TEACH

Características: [37]

- TEACH remoto configurable a través del cable plomo.
- Impedancia de entrada del TEACH: $12K\Omega$
- Ajuste de límites de medición máximo y mínimo de manera remota, y manual.

Modo de funcionamiento y requerimientos:

- El sensor requiere un estado lógico de 0V durante 0.05s para configurar los límites máximos y mínimos de medición.
- Cuando el microcontrolador genera 5V, el cable plomo del sensor estará en alta impedancia.
- Cuando el microcontrolador genera 0V, el conector plomo del sensor estará cortocircuitado con la tierra, voltaje en el cable plomo será 0V.

Para la realización de los cálculos, se considerará en los transistores un $\beta_{sat} = 10$, Voltaje colector- emisor en saturación $V_{ce} = 0.2V$, Voltaje base-emisor $V_{be} = 0.7V$ a 25°C.

- Voltaje de Microcontrolador igual a 5V, transistores T1 y T2 en corte (Igualmente T3 y T4 en el otro circuito), V_{sensor} es 6V.
- Voltaje de Microcontrolador igual a 0V, transistores T1 y T2 saturados (Igualmente T3 y T4 en el otro circuito),

En este estado hallamos el valor de todas las resistencias:

- En la malla de Polarización de T1y T3 definimos la corriente de base $I_{B1} = 0.1mA$, entonces tenemos :

$$R9 = R12 = \frac{V_{uC} - V_{be}}{I_B}\Omega = \frac{5 - 0.7}{0.1mA}\Omega = 43K\Omega$$

Para fines prácticos elegimos la resistencia comercial de $47K\Omega$

Recalculando:

$$I_{B1} = \frac{V_{uC} - V_{be}}{47K\Omega}mA = \frac{5 - 0.7}{47K\Omega}mA = 0.0915mA$$

$$I_{CE1} = \beta \times I_{B1} = 10 \times 0.0915mA = 0.915mA$$

Potencia disipada por la resistencia:

$$P_R = Is^2 xR \Rightarrow P_R = (0.0915mA)^2 \times 47K\Omega = 0.393mW$$

Se utilizarán resistencias de 0.25W

- Malla de polarización de T2 y T4, definimos una $I_B = 1mA$

$$R10 = R13 = \frac{V_{uC} - V_{CE1} - V_{BE}}{I_{B2}}\Omega = \frac{5 - 0.2 - 0.7}{0.1mA}\Omega = 41K\Omega$$

Se elegirá la resistencia comercial 39K

Recalculando:

$$I_{B2} = \frac{V_{uC} - V_{CE1} - V_{BE}}{39K\Omega}mA = \frac{5 - 0.2 - 0.7}{39K\Omega}mA = 0.1051mA$$

$$I_{CE2} = \beta \times I_{B2} = 10 \times 0.1051mA = 1.051mA$$

- En la malla de carga de T1:

$$R11 = R14 = \frac{V_{uC} - V_{CE1}}{I_{CE1} - I_{B2}}\Omega = \frac{5 - 0.2}{9.15 - 0.1051mA}\Omega = 530.685\Omega$$

Se utilizará la resistencia comercial de $560\,\Omega$

Anexo 3.3.5

Diseño del circuito del acondicionamiento de señales del Sensor de Presencia (Sensor Fotoeléctrico)

Los cables de conexión del sensor se observan en la figura 3.12, cuyas funciones y caracteristicas se explican a continuación:

- La alimentación del sensor conectada al cable marrón, para este caso 12V.
- Terminal negativo de la alimentación conectado al cable azul.
- El sensor tiene 2 tipos de salidas(Bipolar):
- Fuente de Corriente o PNP, donde la carga está conetada al cable negro.(terminal positivo) y al cable azul (terminal negativo).
- Disipador de corriente o NPN, donde la carga está conectada al cable marrón(terminal positivo a 12V) y al cable negro.(terminal negativo).

Para el desarrollo de este trabajo se utilizará la configuracion PNP, cuando el sensor detecte la presencia de algún objeto se satura y produce una caída de potencial Vsat = 0.45V@ 50mA, de lo contrario actúa como circuito abierto y tiene un voltaje de salida igual 0V.

- Voltaje de salida del Sensor cuando está saturado: $V_s = A_{limentación} - V_{sat}$

  Vs = 12 – 0.45 = 11.55V

- Corriente máxima que entrega el sensor: 50mA @ 0.45V voltaje de saturación del sensor. Elegimos establecer una corriente de 2mA para un menor consumo de energía y no afecta en gran medida el voltaje de saturación del señor.
- Cálculo de resistencias de carga, para ello se elige una corriente de operación del sensor de 2mA:

  R7+R8= $= \frac{VSalidaSensor}{Isensor} = \frac{11.55}{2mA} = 5.75K\Omega$, elegimos R7=R8= $3.3K\Omega$

  Cálculo de la nueva corriente del sensor:

$$Isensor = \frac{VSalidaSensor}{R7 + R8} = \frac{11.55}{6.6K\Omega} = 1.75mA$$

- Potencia consumida por las resistencias: $P_R = Is^2 xR \Rightarrow P_R = (1.75mA)^2 \times 3.3K\Omega = 10.11\text{mW}$

Se elegirán resistencias de 0.25W

- Utilización de OPAMPS LM324 como buffer para trasladar el voltaje a la salida y no producir efecto de carga [41].
- Voltaje de entrada al buffer: V_{in} = 3.3kΩ x Is $\Rightarrow$ Vin = 4.62V y Vin = 0.1V
- Utilización de diodos de conmutación rápida 1N4148, para protección del microcontrolador ante cortocircuitos y voltajes altos o inversos [45].
- Voltaje de entrada al microcontrolador: Vuc = 4.62V y 0.1V

Tabla 3.7. *Cuadro comparativo de características más resaltantes de una pantalla LCD de caracteres y una gráfica.*

Características	Pantalla LCD de caracteres MOP-AL204A-BYFY-25E-3IN	Pantalla Grafica GDM12864
Tipo	Transflectivo negativo con módulo *backlight*	Transflectivo con módulo *backlight*
Voltaje de alimentación (VDD)	4.5 – 5V	4.75 – 5.25V
Angulo de visión (Fluido STN)	(76mmx25.2mm): Mínimo: -20°C Máximo: 35°C	Mínimo: -40° Máximo: 35° @Cr>2
Tamaño del caracter	2.95 x 4.75mm	0.39 x 0.39mm
Radio de contraste	3	6
Tiempo (ciclo):	Escritura: 2.41ms Lectura: 2.98ms	Escritura: 2.61ms Lectura: 2.62ms
Tiempo de respuesta	$T_{subida} = T_{bajada} = 250ms$	Mínimo; 250ms Máximo: 300ms
Corriente de operación (IDD)	1 – 10ma	1mA
Voltaje de salida	VDD	Mínimo: 0.7Vdd Máximo: VDD
Temperatura de operación	-20 a 70 °C	-20 a 70°C
Conexión de pines	Total pines 16: • 8 pines de datos • 3 pines de control RS, R/W, CE • 2 pines de alimentación Vas y VDD	Total pines 20: • 8 pines de datos. • 6 pines de control RES, CS1, CS2, R/W, D/I, E

	- 1 pin que regula el contraste Bo - 2 pines de *backlight*: ánodo (+) y cátodo (-)	• 3 pines de alimentación VDD y Vas y Ve(variable) • 1 pin que regula el contraste Bo • 2 pines de *backlight*: ánodo (+) y cátodo (-).
Memoria CG-RAM	80B	107B y 108B
Precio	$24.01	$29.95

Adaptado de "MOP-AL204A Parallel Display Specifications" de Texas Matrix Orbital [46] y "User's Guide GDM12864HLCM (Liquid Crystal Display Module)" de XIAMEN OCULAR OPTICS CO [47].

Anexo 3.4.2

Para regular el contraste de la pantalla se implementa el circuito mostrado en la figura 3.14, donde V_0 se conecta a un potenciómetro de 10K y los otros 2 terminales de éste a 5V y a tierra respectivamente.

Asimismo, para controlar el brillo o *backlight,* el pin BL- se conecta de forma similar a V_0 a un potenciómetro de 5K con los otros 2 terminales de éste conectados a 5V y tierra, y además BL+ a fuente, como se observa en la figura 3.15.

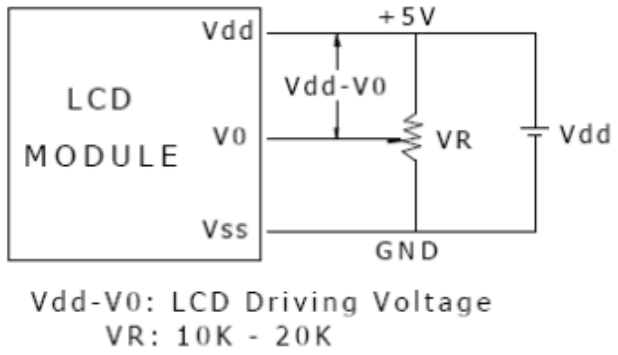


Fig. 3.14. *Circuito de regulación del contraste*

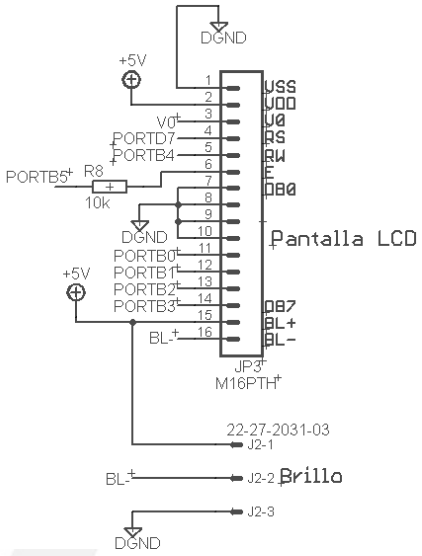


Fig. 3.15. *Circuito de regulación de brillo*

Tabla 3.8. *Comparación de características de uso de un teclado matricial y pulsadores*

Características	Teclado Matricial	Interruptor pulsador Modelo Kan-38
Resistencia al agua y polvo	Si	Agua y humedad
Tiempo de rebote	$\leq 5ms$	-
Voltaje máximo de operación	24Vdc	60Vdc
Máxima corriente de operación	30Ma	100mA
Resistencia de aislamiento	$100M\Omega$ @100V	$100M\Omega$ @500V
Voltaje máximo soportado por el dieléctrico	$250V_{RMS}$ @60Hz por 1min	500v @60Hz 1 min
Expectativa de vida	1000000 de operaciones	10000-50000
Interface de entrada	8 pines de acceso	16 pines de acceso
Dimensiones	6.9 x 7.6cm	-
Precio	S/.5.00	S/.0.10

Adaptado de 4x4 "Matrix Membrane Keypad" de Parallax[48] y "Interruptor de pulsador modelo no:Kan-38" de JialongElectron [49].

Anexo 3.5.2

En la implementación del teclado matricial, cada salida se conectará en configuración *pull down* con los puertos I/O del microcontrolador, como se ilustra en la figura 3.16.

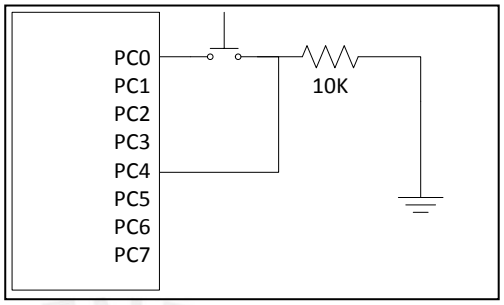


***Figura 3.16**. Modelo de conexión de un pulsador en el teclado matricial*

- Las entradas del microcontrolador son las filas del teclado y están conectados a los puertos PC4 a PC8.
- Las salidas del microcontrolador son las columnas del teclado y corresponden a los puertos PC0 a PC1.
- Si el pulsador está suelto, fluye corriente por la resistencia que es mínima 0.5mA, la entrada TTL del microcontrolador está en alto (5V) y la salida se encuentra en nivel bajo (0V).
- Si el pulsador está presionado, fluye una corriente proveniente de la entrada TTL del microcontrolador (cortocircuito) hacia la salida, por lo tanto ésta se encuentra en nivel alto (5V).

Anexo 3.6.1

Se enviarán 13 Bytes de datos a la computadora los cuales contendrán los valores del largo, alto, ancho en formato ASCII (3 bytes por cada dato) 3 caracteres @ para la separación de tales magnitudes y un byte de para la verificación de correcta transmisión.

$$t_{tx} = \frac{Lcajamaspequeña}{Velocidad\ .\max ima.faja} = \frac{0.05m}{0.7m/s} = 71.43ms$$

$$V_{tx} = \frac{13Bytes \times 8}{71.43ms} = 1456bps$$

La velocidad de transmisión de datos debe ser por lo menos 2 veces esta velocidad para asegurar la correcta transmisión de datos, valor que sería igual a 2912bps

Anexo 3.6.2

Tabla 3.9. *Cuadro comparativo de los tipos de comunicación de datos que pueden emplearse en el circuito de transmisión.*

Características	RS485	RS232
Máxima longitud del cable para la transmisión	Hasta 1200m	Hasta 15m dependiendo de la velocidad de transmisión
Tipo de comunicación	*Half y Full Duplex*	*Half Duplex*
Velocidad de transmisión	Hasta 2.5Mbps	Hasta 9600bps
Atenuación de ruido e interferencias	Mucho mayor	Regular
Uso a nivel industrial	Sí	No
Canales de comunicación	Transmisión: 32 Recepción: 32	Transmisión: 2 Recepción: 2
Voltaje de entrada del transmisor MAX490 **Voltaje de salida del receptor**	**MAX490:** -0.5V a Vcc+0.5V	**MAX232:** -0.3V a Vcc+0.3V
Voltaje de entrada del receptor **Voltaje de salida del transmisor**	**MAX490:** -8V a 12.5V	**MAX232:** Vs(-)-0.3V a Vs(+)+0.3V
Precio	S/. 24.00	S/.5.00

Adaptado de "Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers" [50] y "MAX220–MAX249 MAX220–MAX249 5V-Powered, Multichannel RS-232 Drivers/Receivers" de Maxim Integrated [51].

Anexo 3.6.3

Descripción de Pines de Conexión del circuito Integrado MAX490

- TXD corresponde al pin de transmisión de datos del microcontrolador.
- RXD corresponde al pin de recepción de datos del microcontrolador.
- DI es el pin de entrada del controlador del MAX490, el número 3 en el CI del MAX490, un valor bajo de este establece una salida Y en nivel bajo y Z en alto, asimismo, un nivel alto fuerza un nivel alto en la salida Y y un nivel bajo en Z.
- RO es el pin de recepción de salida, si la entrada de recepción A es mayor que B por 200mV, R0 estará en alto nivel, caso contrario se encontrará en nivel bajo. En el circuito mostrado corresponde al pin 2.
- Y es la salida no invertida del controlador del MAX490, pin 5.
- Z es la salida invertida del controlador del MAX490, pin 6.
- A es la entrada no invertida para la recepción del MAX490, pin 8.
- B es la entrada invertida del para la recepción del MAX490, pin 7.
- Alimentación del circuito integrado Vcc por el pin 1, que variará en el rango de 4.75V y 5.35V
- La tierra del circuito integrado es GND asignado al pin 4.

Anexo 3.6.4

Tabla 3.10. *Comparación de los tipos de protocolos que pueden emplearse en el circuito de recepción de datos.*

Características	Conversor RS485 a RS232 y luego utilizar un adaptador USB a RS232 	Conversor RS485 a USB mediante la utilización circuito integrado FT232 
Interfaces estándares a utilizarse	Recepción mediante interfaz RS485 mediante MAX490, conversión a interfaz RS232 mediante MAX232, y adaptador RS232 con conector DB9 a USB compatible con USB 1.1 y 2.0	Recepción mediante interfaz RS485 mediante MAX490, conversión RS485 a USB mediante FT232.
Complejidad de diseño e implementación	Menos complejo y bajo costo	Más complejo y más costoso
Velocidad máxima en baudios	MAX232: 120Kbps MAX490: 2.5Mbps Adaptador USB-RS232: 921.6Kbaudios	MAX232: 120Kbps MAX490: 2.5Mbps FT232: 300 baudios a 3Mbaudios para el estándar RS485
Compatibilidad con Sistemas Operativos	Windows7, Vista, XP, 2000ME, 98SE Windows Server 2008 R2 Mac OS 10.x, Linux	Windows 7, Vista, XP, 98, 98SE, ME, 2000, Server 2003 y 2008 Windows XP embebidos Mac OS 8/9, OS-X
Buffer de Transmisión y Recepción	Adaptador USB-RS32: FIFO 192 Bytes en el buffer de recepción y 96 Bytes en el de transmisión	Circuito Integrado FT232: FIFO 256 Bytes en el buffer receptor y 128 Bytes en el de transmisión
Precio	MAX232 : S/. 3.00 MAX490 : S/. 23.00 Adaptador USB-RS232: S/.18	MAX490: S/. 23.00 FT232: S/. 57.00

Adaptado de "Conversor USB-RS232 (ZT-RS232B)" de Zone-Tek (HK) CompanyLimited [52] y "FT232R USB UART IC" de Future Technology Devices International Ltd. [53].

Anexo 3.6.5

Descripción de Pines de Conexión del circuito Integrado MAX232

- T1IN pin de transmisión de datos de entrada del MAX232.
- T1OUT pin de transmisión de datos de salida del MAX232.
- R1IN pin de recepción de datos de entrada del MAX232.
- R1OUT pin de recepción de datos para enviarlos como salida del MAX232.
- Pin 2 de conector DB9 utilizado para la recepción de datos, pin 3 usado para la transmisión de datos y pin 5 para la conexión a tierra, la disposición de estos puntos de conexión se puede observar de mejor forma en la figura 3.19.

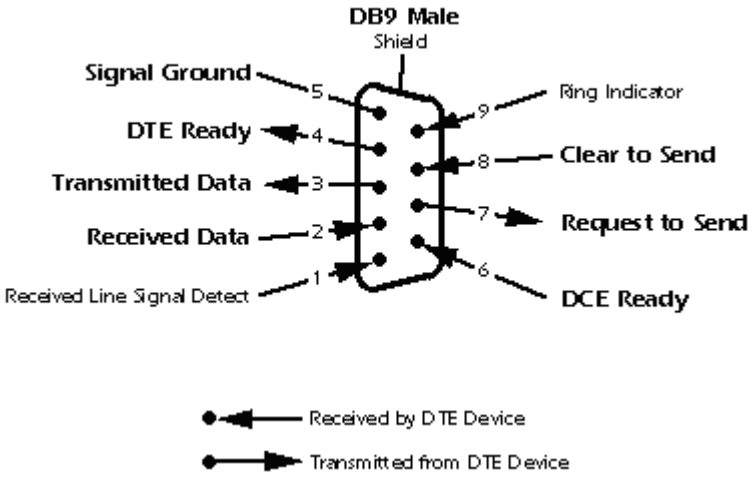
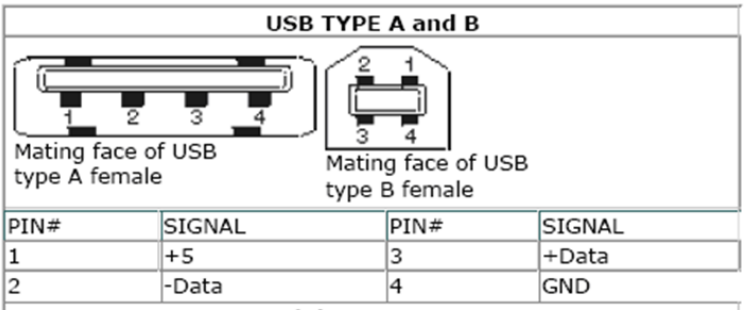


Fig. 3.19. Diagrama de conexiones del conector DB9

Fig. 3.20. *Tipos de Conectores USB*



PIN#	SIGNAL	PIN#	SIGNAL
1	+5	3	+Data
2	-Data	4	GND

Fig. 3.21. *Diagrama de conexión del conector USB Tipo A*

Anexo 3.7.1

Tabla 3.12. *Comparación de microcontroladores ATmega16 y7 PIC16F87.*

	Comunicación de datos	Número de puertos de E/S	Canales ADC	Canales de temporización	Memoria programa	Memoria RAM	Set de instrucciones	Costo
ATmega16 ATMEL	1 USART	32	8 de 10 bits	2 de 8 bits y 1 de 16 bits 4 canales PWM	16KB	1KB	131	S/. 28 Empaque superficial
PIC16F877 MICROCHIP	1 USART	22	8 de 10 bits	2 de 8bits y 1 de 16bits	8KB	1KB	35	S/. 20 Empaque superficial

Adaptado de "8-bit Microcontroller with 16K Bytes In-System Programmable Flash ATmega16-ATmega16L" de Atmel [54] y "PIC16F84A Data Sheet 18-pin Enhanced FLASH/EEPROM 8-bit Microcontroller" de Microchip [55].

Anexo 3.7.

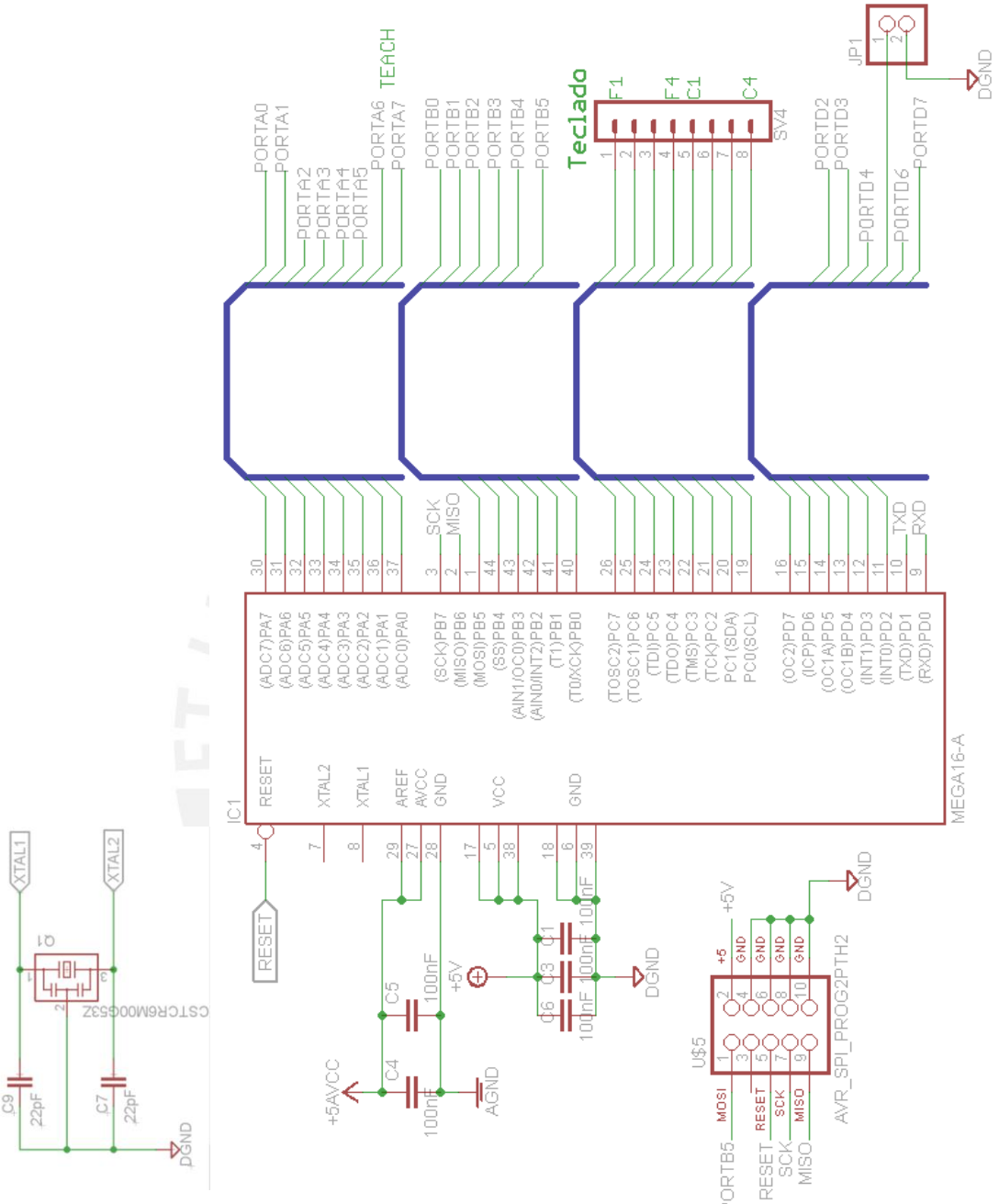


Fig. 3.22. *Diagrama de conexiones del microcontrolador ATMega16*

Descripción:

- Teclado matricial de 4x4 conectado a todos los pines del puerto C configurado como entradas.
- Pantalla LCD de 4x16 caracteres en modo de 4 bits conectado al puerto B configurado como salida de la siguiente forma:

- Pin RS conectado a PD7
- Pin R/W conectado a PB4
- Pines de datos de DB4 a DB8 conectados a los puertos PB0 a PB3
- Pin de activación de la pantalla E conectado a PB5 (Modo de 4 bits).

- Comunicación USART a través del puerto de entrada PD0 como recepción y del puerto de salida PD1 para la transmisión de datos.
- Canales de ADC conectados a los sensores ultrasónicos por los puertos de entrada PA0 y PA1
- Señales generadas por los sensores fotoeléctricos conectados a los pines de entrada PD2 y PD3
- Calibración en modo TEACH (automático) de los sensores ultrasónicos a partir de los puertos de entrada PA2 y PA3.
- Control de velocidad a través del canal PWM del puerto PD5.
- Conexión del programador del microcontrolador en los pines PB5 (MOSI), PB6 (MISO), PB7 (SCK) y RESET.
- XTAL1 y XTAL2 para la inclusión de un cristal resonador externo de 1Mhz.
- Utilización de los temporizadores 0, 1 y2 de la siguiente forma:
  - Temporizador 0, que corresponde al puerto PB3, utilizado para calcular el tiempo que demora una caja desde que activa el sensor fotoeléctrico 1 hasta su llegada al número 2.
  - Temporizador 2, que corresponde al puerto PD7, utilizado para registrar los valores de los datos del ADC cada 2ms.
- Interrupciones externas INT0 y INT1 de los puertos PD2 y PD3, utilizados para detectar la activación de los sensores fotoeléctricos.

Anexo 3.8.1

Tabla 3.13.*Requerimientos eléctricos de la etapa de control.*

Dispositivo	Voltaje (V)	Corriente (mA)	Potencia (mW)
ATmega16A	5.0	15.0	75.0
Pantalla LCD	5.0	2.5	12.5
MAX490	5.0	0.5	2.5
	Total	18.0	90.0

Tabla 3.14.*Requerimientos eléctricos de la etapa de sensores.*

Dispositivo	Voltaje (V)	Corriente (mA)	Potencia (W)
Sensor Ultrasónico	12.0	65.0	(x2) 1.56
Sensor Fotoeléctrico	12.0	20.0	(x2) 0.48
LM324	12.0	3.0	(x4) 0.144
MMBT3906	V_E= 5.0	I_C= 10.0	(x2) 0.1
	Total	98.0	2.2844

Tabla 3.15.*Requerimientos eléctricos de la etapa de potencia.*

Dispositivo	Voltaje (V)	Corriente (mA)	Potencia (W)
Motor DC	20.0	3000	60.0
MMBT3906	V_E= 12.0	I_E= 1.0	(x2) 0.48
4N35	V_C = 12.0	I_C= 1.0	(x2) 0.1
	Total	3002	60.58

Anexo 3.8.2

Tabla 3.17. *Desempeño de una fuente de alimentación lineal versus una conmutada.*

Característica	 ***Fuentes lineales***	 ***Fuentes conmutadas***
Eficiencia	45% - 55%	60% - 95%
Regulación de línea	0.02% - 005%	0.05% - 0.1%
Regulación de carga	0.02% - 0.1%	0.1% - 1.0%
Voltaje de rizado en la salida	0.5mV - 2.0mV_{RMS}	10mV - 100mV_{PP}
Proporción potencia – masa	22W/kg	88W/kg
Ruido (EMI)	Despreciable	50mV - 200mV_{PP}
Complejidad	Simplicidad en el diseño. Además de la etapa de rectificación, requiere solo un circuito integrado y capacitores.	Se agrega una etapa de control por PWM. Presencia de transformadores, inductores, transistores y filtros.
Costo	Para potencias menores a 20W, el precio de los componentes es igual al de una fuente conmutada. En el mercado local: S/.80 aprox.	El elaborado diseño y la inclusión de más elementos activos y pasivos encarecen el sistema. En páginas web: $ 50 el precio barato.

Adaptado de "Linear & Switching Voltage Regulator Handbook" de ON Semiconductor™ y "AN-556 Introduction to Power Supplies" de Texas Instruments [57] y [58].

Anexo 3.8.3

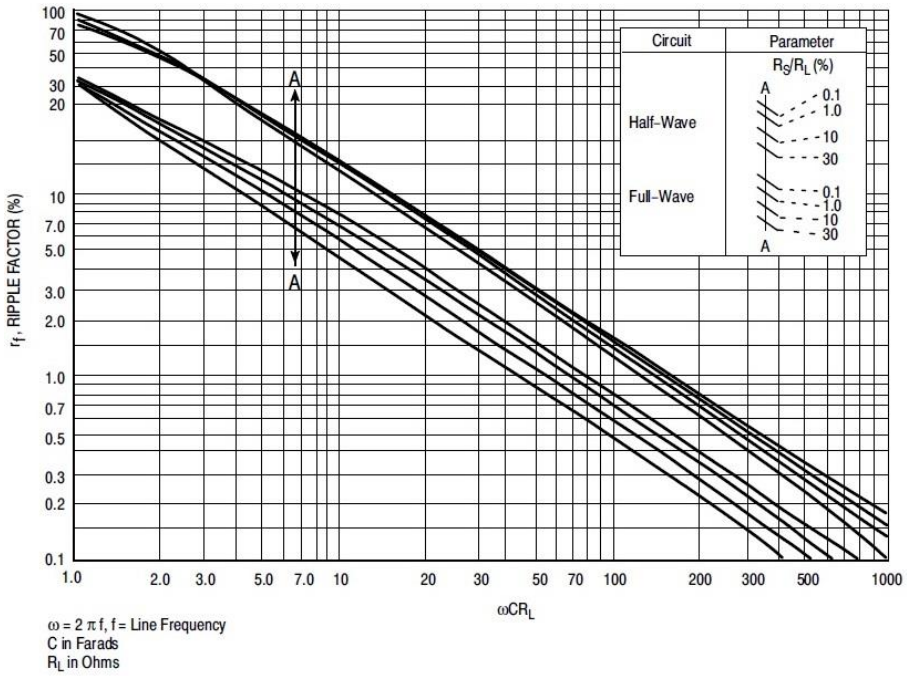


Figura 3.23.*Gráfica $r_f$vs wCR_L utilizada para hallar un rango de wCR_L.*

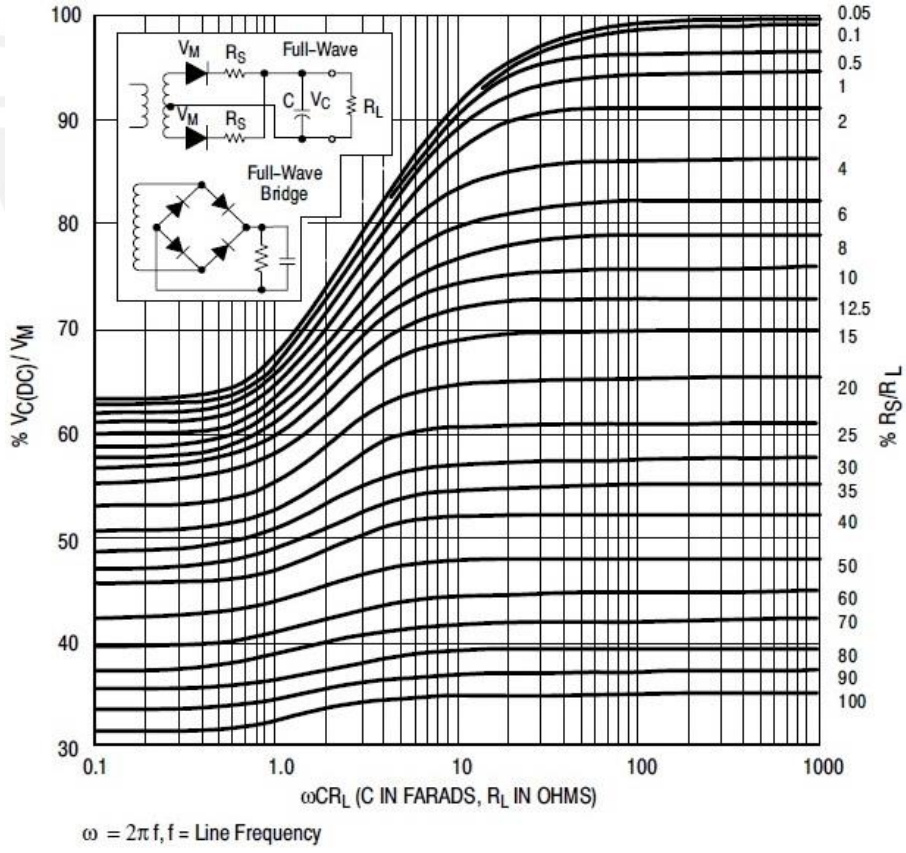


Figura 3.24. *Gráfica utilizada para hallar el parámetro R_S/R_L en %.*

Anexo 3.8.4

Diseño del filtro capacitivo de entrada:

Primero se determina el valor de la resistencia de carga R_L y se fija Vc (dc).

$$Rl = \frac{Vc(dc)}{Icarga} = \frac{9.0}{1.0} = 9.0\Omega$$

Se adopta un valor de R_S de: R_S= 10%*(R_L) =10%*(9.0) = 0.9Ω

Establecemos un voltaje de rizado de 1.0V_{pp} y hallamos un factor de rizado:

$$rf = \frac{Vripple(pp)}{2\sqrt{2}(Vc(dc))}\ x\ 100\% = \frac{1.0V}{2\sqrt{2}(9)}\ x\ 100\% = 3.92\%$$

Con este factor en la figura 3.23 hallamos un valor estimado para wCR_L. Elegimos wCR_L = 16. Luego hallamos C_L:

$$C = \frac{16}{wRl} = \frac{16}{2\pi * 60 * (9)} = 4715.7\mu F => Valor\ comercial = \mathbf{4700\mu F}$$

Considerando el nuevo valor de C, se halla el nuevo valor de wCR_L.

$$wCRl = 2 * \pi * 60 * 4700\mu * 9 = 15.94$$

Luego, en la figura 3.24 se calcula la relación entre el voltaje dc a la salida del regulador y el valor máximo de voltaje en la salida del filtro.

$$75\% = \frac{Vc(dc)}{Vmax} = \frac{9}{Vmax} => Vmax = 12\ V$$

Este valor es el máximo valor de voltaje a la salida del transformador que se usaría. Para entenderlo en términos más simples, se pasará a magnitudes de voltajes eficaces.

$$Vrms = \frac{Vmax}{\sqrt{2}} => Vrms = \frac{12}{\sqrt{2}} = 8.48V$$

El valor R_S es valor de la resistencia que posee el devanado secundario del transformador que se utiliza. El voltaje en el secundario del transformador debe ser aproximadamente 8.48$V_{RMS.}$

Fuente de alimentación de 12V:

Diseño del filtro capacitivo de entrada:

Primero se determina el valor de la resistencia de carga R_L y se fija Vc (dc).

$$Rl = \frac{Vc(dc)}{Icarga} = \frac{15.0}{1.0} = 15\Omega$$

Se adopta un valor de R_S de: R_S= 5%*(15) = 5%*(15) = 0.75Ω

Establecemos un voltaje de rizado de 2.0V_{pp}y hallamos un factor de rizado:

$$rf = \frac{Vripple(pp)}{2\sqrt{2}(Vc(dc))}\ x\ 100\% = \frac{2.0V}{2\sqrt{2}(15)}\ x\ 100\% = 4.71\%$$

Con este factor en la figura 3.23 hallamos un valor estimado para wCR_L. Elegimos wCR_L = 13. Luego hallamos C:

$$C = \frac{13}{wRl} = \frac{13}{2\pi * 60 * (15)} = 2298\mu F => Valor\ comercial = \mathbf{2200\mu F}$$

Considerando el nuevo valor de C, se halla el nuevo valor de wCR_L.

$$wCRl = 2 * \pi * 60 * 2200\mu * 15 = 12.44$$

Luego, en la figura 3.24 se calcula la relación entre voltaje dc a la salida del regulador y el valor máximo de voltaje en la salida del filtro.

$$82\% = \frac{Vc(dc)}{Vmax} = \frac{15}{Vmax} => Vmax = 18.29\ V$$

Este valor es el máximo valor de voltaje a la salida del transformador que se usaría. Para entenderlo en términos más simples, se pasará a magnitudes de voltajes eficaces.

$$Vrms = \frac{Vmax}{\sqrt{2}} => Vrms = \frac{18.29}{\sqrt{2}} = 12.93V$$

El valor R_S es valor de la resistencia que posee el devanado secundario del transformador que se utiliza. El voltaje en el secundario del transformador debe ser aproximadamente 12.93 $V_{RMS.}$

Cabe resaltar que la salida de 5 voltios para la alimentación de los sensores, proviene de la fuente de alimentación de 5 voltios diseñada seguida de una etapa de filtrado por medio de una bobina y un condensador con los valores mostrados. Dicho filtro es el recomendado para aislar la etapa analógica de la digital según la hoja de datos del ATmega16 [54].

Cálculo del disipador:

Para poder calcular el disipador en un regulador de voltaje es necesario conocer la diferencia de voltaje en la entrada y la salida del mismo, así como la corriente que proveerá al sistema que se desea alimentar. En primer lugar se debe conocer la potencia total disipada en el regulador de voltaje, para esto se hace un análisis con la siguiente ecuación *[61]*.

$$\boldsymbol{Pd} = (Vin - Vout) * IL + (Vin * Ig)$$

Donde I_L es la corriente de carga e I_g es la corriente de fuga del regulador.

Para el regulador LM7805:

$$\boldsymbol{Pd} = (12.3 - 5) * (0.018) + (12.3 * 0.008) => \boldsymbol{Pd = 0.23W}$$

Para el regulador LM7812:

$$\boldsymbol{Pd} = (15.27\text{V} - 12\text{V}) * (98\text{mA}) + (15.27\text{V} * 8\text{mA}) => \boldsymbol{Pd = 0.44W}$$

El siguiente parámetro a calcular es la temperatura máxima permitida en el componente y viene dado por la diferencia de la temperatura máxima en la juntura del empaque en el regulador (T_j) y la temperatura del medio ambiente donde está expuesto el dispositivo (T_a). Se asigna el parámetro T_j= 100ºC como factor de seguridad pese a que el regulador posee un $T_{j\,(máx)}$= 125ºC.

$$\boldsymbol{Tja(max)} = Tj(max) - Ta(max) => Tja(max) = 100ºC - 25ºC = 75ºC$$

Luego se calcula el máximo valor de resistencia térmica permitida entre la juntura del empaque y el medio ambiente para el caso del regulador LM7805.

$$\boldsymbol{\theta ja} = \frac{Tr(max)}{Pd} = \frac{75ºC}{0.23W} = 326ºC/W$$

Debido a que la resistividad térmica Θ_{JA} calculada es mucho mayor que la máxima permitida según la hoja de datos del regulador (Θ_{JA} = 65ºC/W), no es necesario el uso de un disipador.

De manera similar se realiza el análisis para el regulador LM7812. Se calcula la máxima resistencia térmica (Θ_{JA}).

$$\Theta ja = \frac{Tr(max)}{Pd} = \frac{75ºC}{0.44W} = 170.4ºC/W$$

Ya que la resistencia térmica hallada es mayor que la máxima permitida según la hoja de datos no es necesario el uso de un disipador. Como precaución se hace uso del mismo modelo del disipador usado en el MOSFET en conmutación para ambos reguladores de voltaje.

Anexo 3.8.5

Diseño del filtro capacitivo de entrada:

Tomamos en cuenta la nomenclatura mencionada anteriormente para el mismo cálculo.

Primero se determina el valor de la resistencia de carga R_L y se fija Vc (dc).

$$Rl = \frac{Vc(dc)}{Icarga} = \frac{24.0}{3.0} = 8.0\Omega$$

Se adopta un valor de R_S de: R_S= 10%*(R_L) = 10%*(8) = 0.8Ω

Establecemos un voltaje de rizado de 2.0V_{pp} y hallamos un factor de rizado:

$$rf = \frac{Vripple(pp)}{2\sqrt{2}(Vc(dc))}\ x\ 100\% = \frac{2.0V}{2\sqrt{2}(24)}\ x\ 100\% = 2.94\%$$

Con este factor en la figura 3.23 hallamos un valor estimado para wCR_L. Elegimos wCR_L = 16. Luego hallamos C:

$$C = \frac{15}{wRl} = \frac{15}{2\pi * 60 * (8)} = 4974.4\mu F => Valor\ comercial = \mathbf{4700\mu F}$$

Considerando el nuevo valor de C, se halla el nuevo valor de wCR_L.

$$wCRl = 2 * \pi * 60 * 4700\mu * 8 = 14.17$$

Luego, en la figura 3.24 se calcula la relación entre voltaje dc a la salida del regulador y el valor máximo de voltaje en la salida el del filtro.

$$84\% = \frac{Vc(dc)}{Vmax} = \frac{24}{Vmax} => Vmax = 28.57V$$

Este valor es el máximo valor de voltaje a la salida del transformador que se usaría. Para entenderlo en términos más simples, se pasará a magnitudes de voltajes eficaces.

$$Vrms = \frac{Vmax}{\sqrt{2}} => Vrms = \frac{28.57}{\sqrt{2}} = 20.20V$$

El voltaje en el secundario del transformador debe ser aproximadamente 20.20 V_{RMS}.

De la misma manera, se acoplará una etapa de regulación mediante el circuito integrado LM317, el cual proporcionará un voltaje ajustado a 20V. El voltaje a la salida del regulador se obtiene de acuerdo a la siguiente expresión:

$$\boldsymbol{Vout} = 1.25\,(1 + \frac{R2}{R1}) + Iadj(R2)$$

Donde el valor de la resistencia R_1 se fija en 240Ω como sugerencia de la hoja de datos. Sin embargo, se optará por el valor comercial de 220Ω. El valor V_{OUT} es el deseado, es decir, 20 voltios. Con estos datos se puede hallar R_2.

$$20V = 1.25\left(1 + \frac{R2}{220}\right) + 50\mu(R2) => R2 = 3.271K\Omega$$

$$\boldsymbol{R2}\ (valor\ comercial) = 3.3K\Omega$$

Para que ésta última fuente de alimentación pueda suministrar 4 amperios se hace uso del transistor de potencia PNP MJ2955 mostrado en la figura 3.26, el cual soporta un flujo de corriente continua de hasta 15 amperios por su colector. Se usa este transistor como un "*bypass*", es decir un camino alterno por el cual una gran magnitud de corriente pueda circular ya que el regulador LM317 puede suministrar solo hasta 1.5 amperios.

Figura 3.26. *Transistor PNP MJ2955 empaque TO-3*

La corriente que exige el sistema circulará por el colector del transistor de potencia cuando por la resistencia de 33Ω circule una determinada magnitud de corriente.

Cuando lo anterior suceda, el transistor estará en la región de saturación y proveerá la corriente necesaria.

Se analizará el caso de máxima corriente que puede proveer el transformador, es decir 4.0 amperios en el circuito esquemático mostrado en la figura 3.27.

El voltaje de saturación del transistor MJ2955 es V_{BE} = 1.8V. Para saturar al transistor se elige $hfe_{(mín.)}$ o β = 20. Cuando el transistor se satura se tiene la siguiente expresión:

$$\boldsymbol{VR2 + Vbe = (Iin - Ib) * R1 + Vdiodo}$$

$$4 * 0.22 + 1.8 = \left(0.25 - \frac{4}{20}\right) * R1 + 0.8 => \boldsymbol{R1} = 37.6 \approx 33\Omega$$

De esta manera la resistencia R_1 actuará como sensor de corriente y cuando circule por el aproximadamente 250mA, el voltaje entre base y emisor del transistor de potencia será el suficiente para saturarlo. La resistencia R_1 disipa una potencia de máxima $P_1 = (I_{IN})^2 \times R_1$ = 2.0625 W y la resistencia R_2 una potencia de $P_2 = (I_C)^2 \times R_2$ = 3.52 W. Se usará resistencias de los valores mencionados para R_1 y R_2 con una capacidad de 5 vatios.

Cálculo del disipador:

Caso: Regulador LM317

El cálculo del disipador en el regulador ajustable LM317 es similar al análisis de lo realizado anteriormente para el regulador LM7805 o LM7812.

$$\boldsymbol{Pd} = (Vin - Vout) * IL + (Vin * Iadj(max))$$

Donde I_L es la corriente de carga e I_{adj} es la corriente en el terminal de ajuste.

$$\boldsymbol{Pd} = (28.284\text{V} - 20\text{V}) * (250\text{mA}) + (28.284\text{V} * 100\mu\text{A}) => \boldsymbol{Pd = 2.07W}$$

Se asigna el parámetro T_j= 100ºC como factor de seguridad pese a que el regulador posee un $T_{j\,(máx)}$= 125ºC.

$$\boldsymbol{Tja(max)} = Tj(max) - Ta(max) => Tja(max) = 100ºC - 25ºC = 75ºC$$

Luego se calcula el máximo valor de resistencia térmica permitida entre la juntura del empaque y el medio ambiente.

$$\Theta ja = \frac{Tr(max)}{Pd} = \frac{75ºC}{2.07W} = 36.23ºC/W$$

Debido a que la resistividad térmica Θ_{JA} calculada es mayor que la máxima permitida según la hoja de datos del regulador (Θ_{JA} = 50ºC/W), no es necesario el uso de un disipador; sin embargo, como precaución se hace uso del mismo modelo de disipador usado en el MOSFET en conmutación ya que ambos la resistencia térmica teórica es cercana a la máxima resistencia térmica permitida.

Caso: Transistor MJ2955

Para el cálculo del disipador en el transistor de potencia se calcula la potencia disipada cuando éste se encuentra saturado y fluye corriente por su colector.

$$\boldsymbol{Pd(max)} = Vce * Ic => \boldsymbol{Pd(max)} = 1.1 * 4 => Pd(max) = \boldsymbol{4.4W}$$

Luego, procedemos a calcular la resistencia térmica entre la juntura del semiconductor y el medio ambiente *[62]*.

$$\boldsymbol{R\Theta ja} = \frac{Tj(max) - Ta(max)}{Pd(max)} = \frac{150ºC - 25ºC}{4.4W} = 28.4ºC/W$$

La resistencia térmica $R\Theta_{JA}$ se puede representar como una suma de otras resistencias térmicas en el dispositivo semiconductor según la siguiente expresión:

$$R\Theta ja = R\Theta jc + R\Theta cs + R\Theta sa$$

Si se asigna $R\Theta_{JC}$ = 1.52ºC/W (según hoja de datos) y $R\Theta_{CS}$ = 0.5ºC/W podemos obtener la resistencia térmica entre el disipador y el medio ambiente.

$$\boldsymbol{R\Theta sa} = R\Theta ja - R\Theta jc - R\Theta cs => \boldsymbol{R\Theta sa} = 28.4 - 1.52 - 0.25$$

$$\boldsymbol{R\Theta sa(max) = 26.63ºC/W}$$

Este resultado sugiere usar un disipador que posea una resistencia térmica no mayor a 26.38ºC/W. Se tomará como referencia el disipador modelo 575603 de la empresa AAVID THERMALLY con una resistencia térmica 7.2ºC/W mostrado en la figura 3.28 *[56]*.

***Figura 3.28**. Disipador horizontal para empaque TO-3.*

Caso: Regulador LM7812

Para el caso del regulador LM7812 se recurre al análisis realizado anteriormente:

$$\boldsymbol{Pd} = (28.284\text{V} - 12\text{V}) * (10\text{mA}) + (28.284\text{V} * 8\text{mA}) => \boldsymbol{Pd = 0.389W}$$

Se asigna el parámetro T_j= 100ºC como factor de seguridad pese a que el regulador posee un $T_{j\,(máx)}$= 125ºC.

$$\boldsymbol{Tja(max)} = Tj(max) - Ta(max) => Tr(max) = 100ºC - 25ºC = 75ºC$$

Luego se calcula el máximo valor de resistencia térmica permitida entre la juntura del empaque y el medio ambiente.

$$\Theta\boldsymbol{ja} = \frac{Tja(max)}{Pd} = \frac{75ºC}{0.389W} = 192.8ºC/W$$

Debido a que la resistividad térmica Θ_{JA} calculada es mucho mayor que la máxima permitida según la hoja de datos del regulador (Θ_{JA} = 65ºC/W), no es necesario el uso de un disipador; sin embargo, como precaución se hace uso del mismo modelo de disipador usado en el MOSFET en conmutación.

Anexo 3.9.1

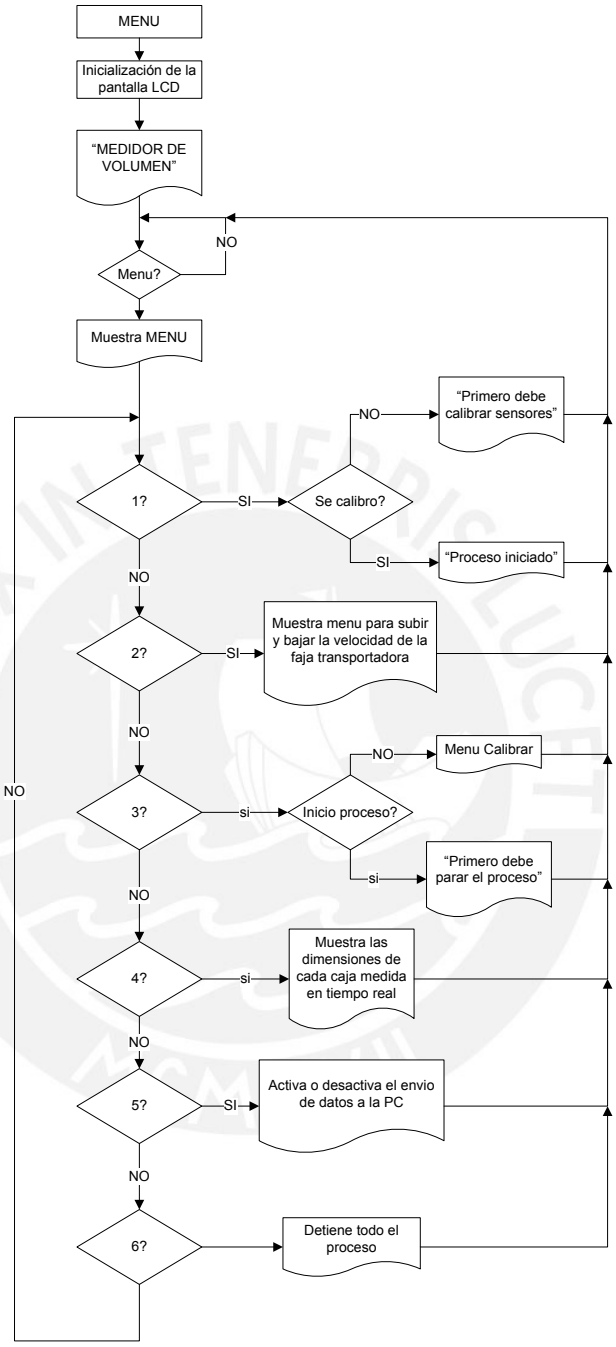
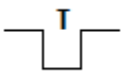
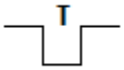


Figura 3.31. *Diagrama de flujo de la tarea menú.*

Anexo 3.9.2

Tabla 3.21. *Métodos de calibración de los sensores ultrasónicos.*

	Procedimiento		Resultado
	Botón 0.04<"clic"<0.8s	Cable remoto 0.04s<T<0.8s	
Modo de programación	Presionar y mantener presionado hasta obtener los resultados	No se requiere ninguna acción, el sensor está listo para aprender el primer limite	OUTPUT LED: encendido rojo Power LED: encendido verde (Buena señal) o encendido rojo (sin señal)
Enseñar el primer limite	Posicionar el objeto para el primer limite	Posicionar el objeto para el primer limite	Power LED:debe estar en verde
	Presionar el botón	Generar un pulso en el cable plomo 	Limite aprendido (el sensor aprende el limita para 4mA) Output LED: rojo parpadeante Limita no aceptado Output LED: encendido rojo
Enseñar el segundo limite	Posicionar el objeto para el segundo limite	Posicionar el objeto para el segundo limite	Power LED:debe estar en verde
	Presionar el botón	Generar un pulso en el cable plomo 	Limite aprendido (el sensor aprende el limita para 20mA) Output LED: amarillo o apagado Limita no aceptado Output LED: rojo parpadeante

Anexo 3.9.3

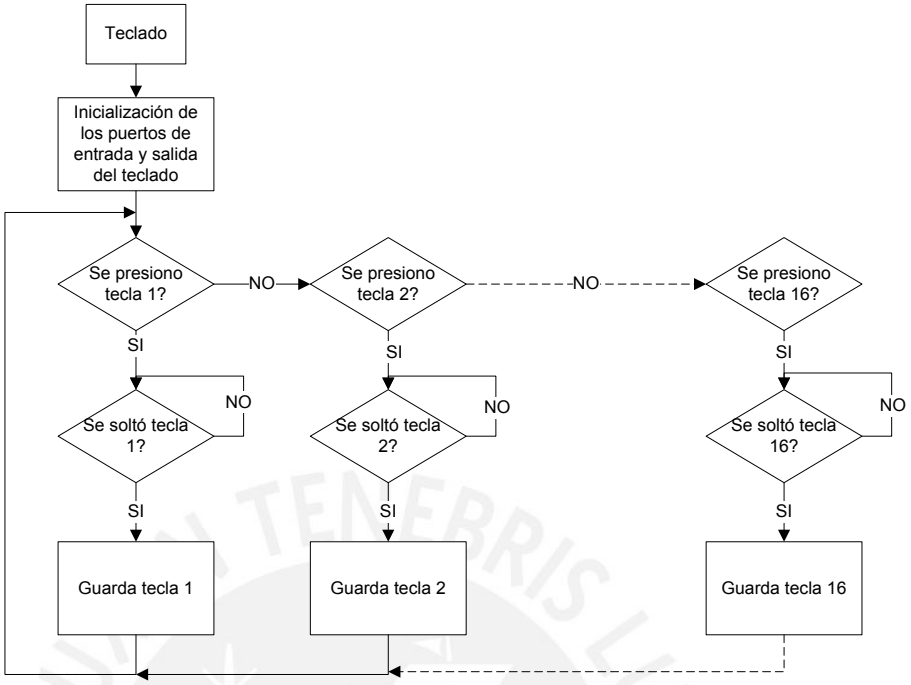


Figura 3.33. *Diagrama de flujo del programa que escanea los pulsadores del teclado matricial.*

Anexo 3.9.4

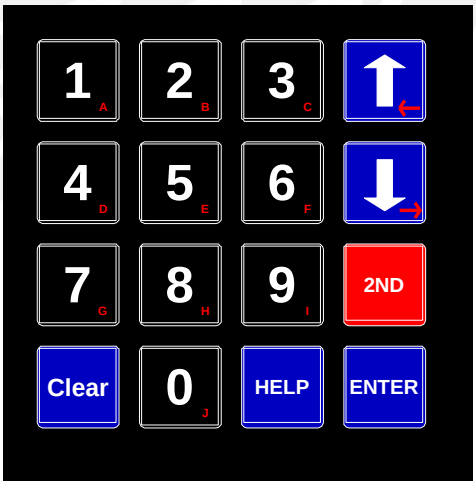


Figura 3.34. *Teclado matricial de 4 filas x 4 columnas*

Anexo 3.9.5

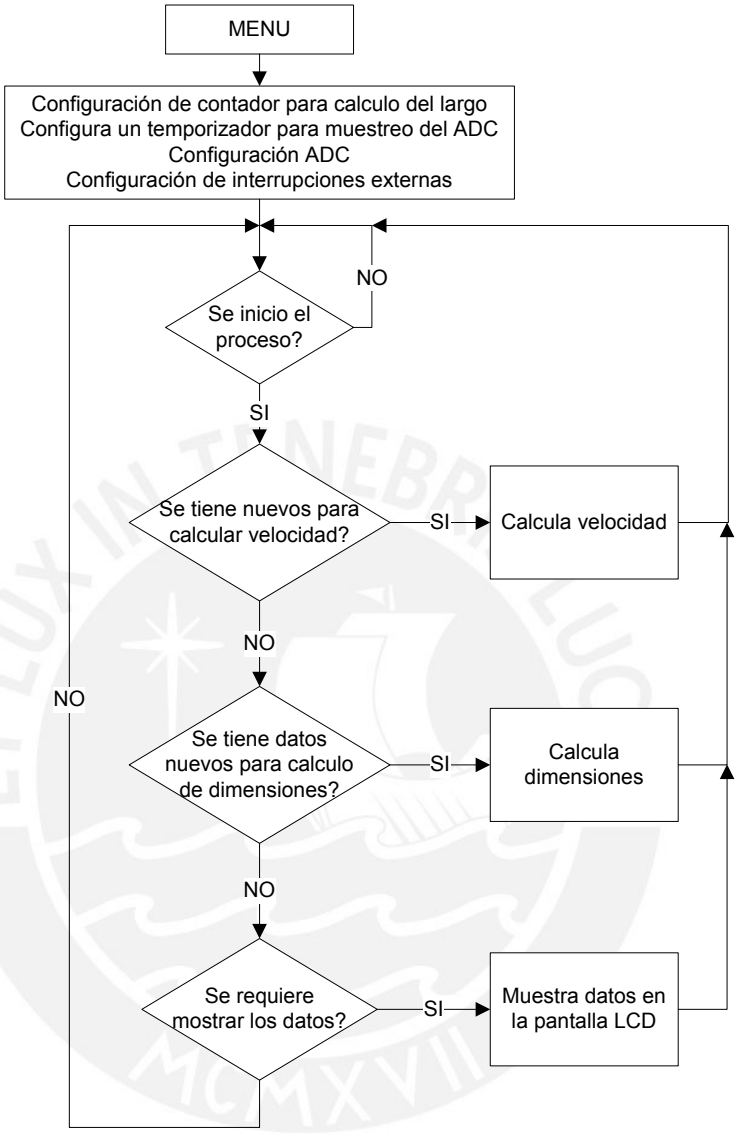


Figura 3.39. *Diagrama de flujo del programa que calcula de la velocidad de la faja transportadora y largo de las cajas.*

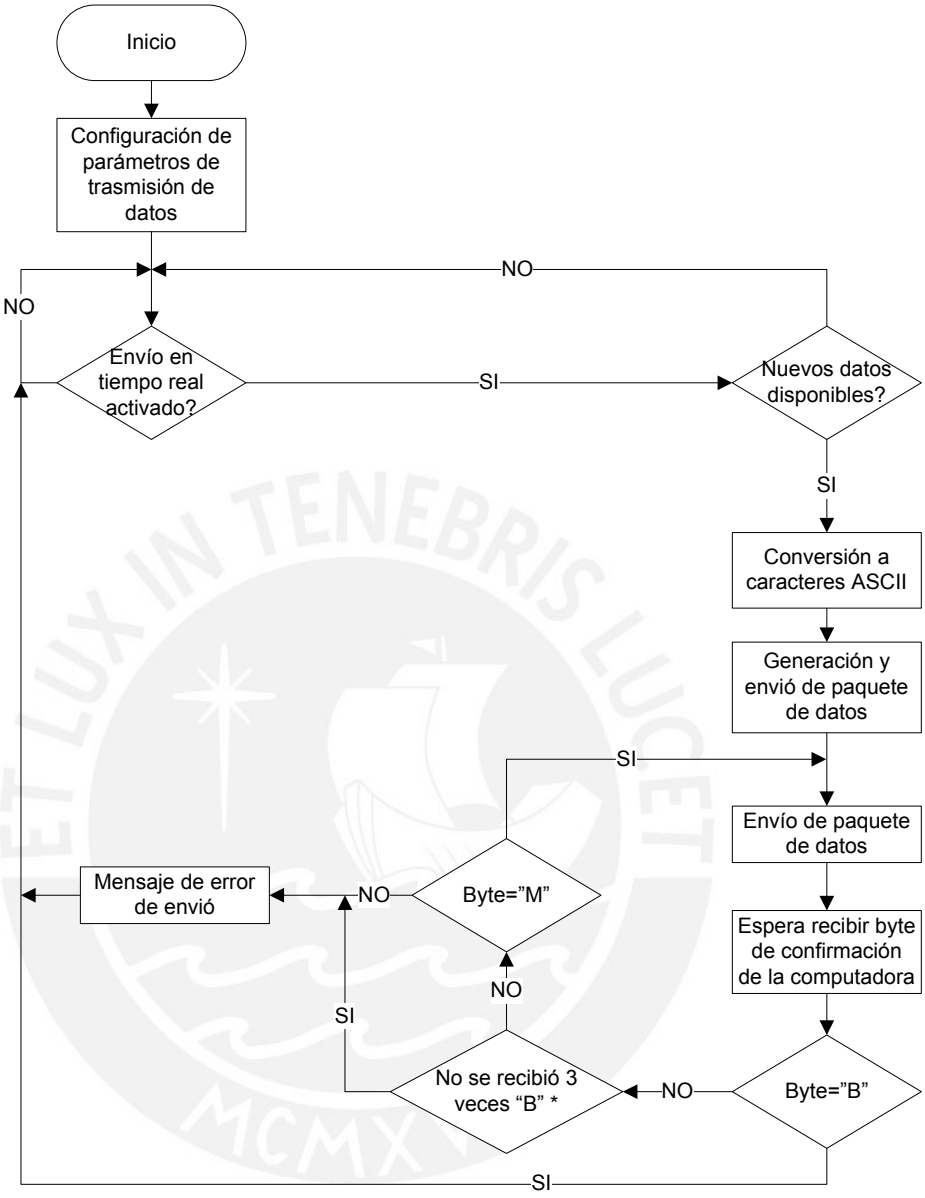


Figura 3.41. *Diagrama de flujo del programa de comunicación de datos*

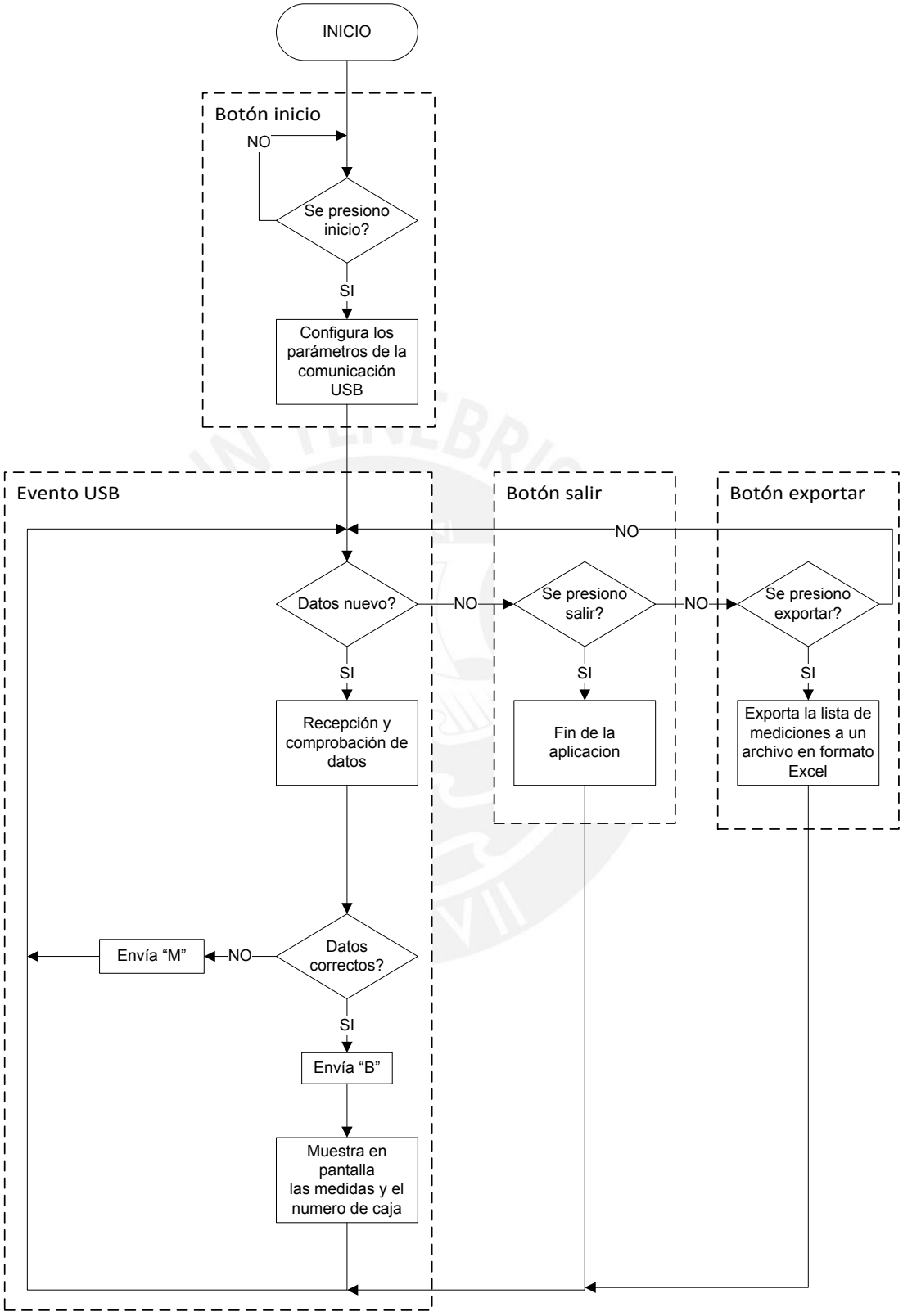


Figura 3.43. *Diagrama de flujo del programa en Visual Basic.*

ANEXO 4

Anexo 4.1

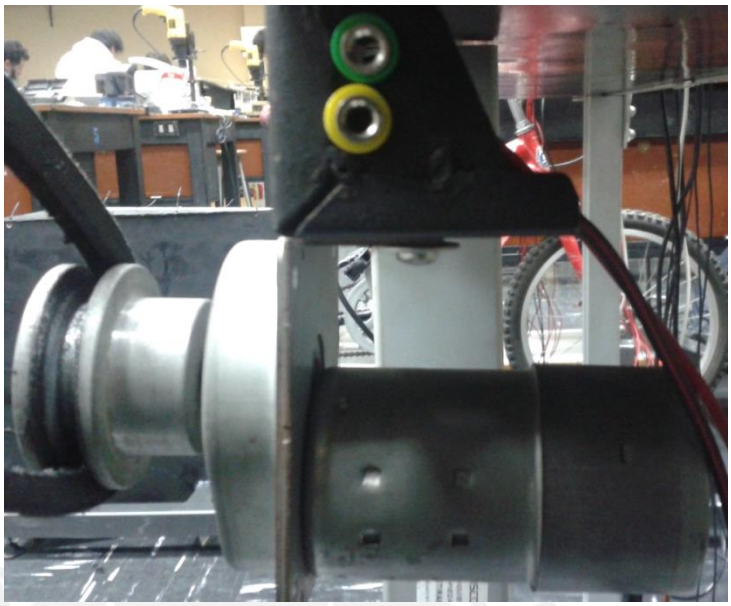


Figura 4.2. *Terminales de conexión y vista lateral del motor.*

Anexo 4.2

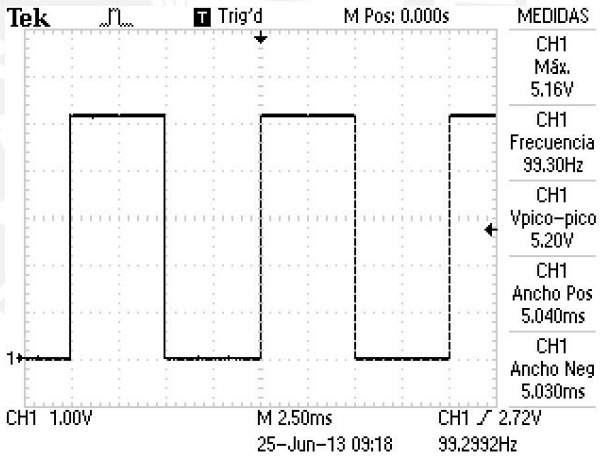


Figura 4.3 *Señal modulada por ancho de pulso con 50% de ciclo de trabajo.*

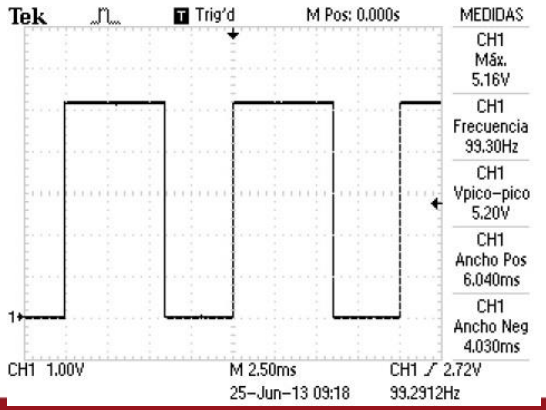


Figura 4.4. *Señal modulada por ancho de pulso con 60% de ciclo de trabajo.*

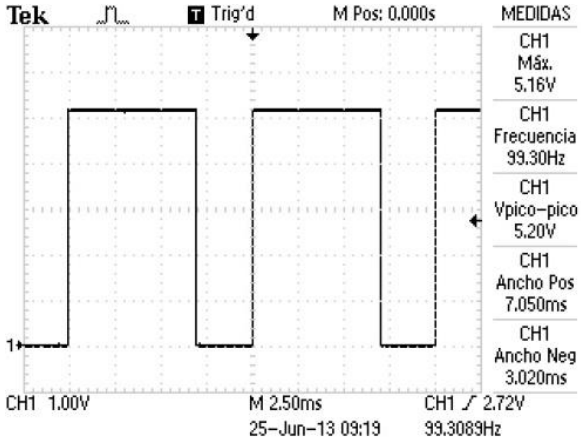


***Figura** pulso*

***4.5.** Señal modulada por ancho de*

con 70% de ciclo de trabajo.

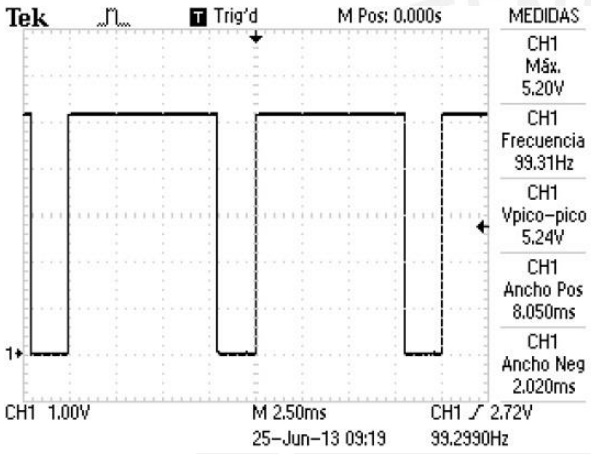


***Figura 4.6.** Señal modulada por ancho de pulso con 80% de ciclo de trabajo.*

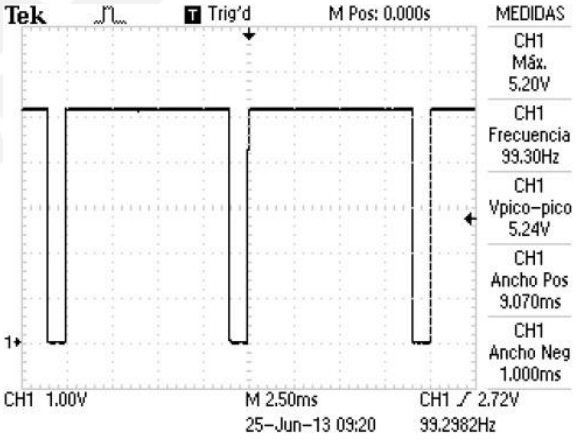


***Figura 4.7.** Señal modulada por ancho de pulso con 90% de ciclo de trabajo.*

Figura 4.8 *Medición de la velocidad usando un tacómetro digital*

Anexo 4.4

Tabla 4.1. *Velocidad de la polea 1 (Motor) y de la polea 2 (faja transportadora) con relación a los diferentes ciclos de trabajo de la señal de control PWM.*

Velocidad (RPM) Ciclo de trabajo 50%		**Velocidad (RPM) Ciclo de trabajo 60%**		**Velocidad (RPM) Ciclo de trabajo 70%**		**Velocidad (RPM) Ciclo de trabajo 80%**		**Velocidad (RPM) Ciclo de trabajo 90%**		**Velocidad (RPM) Ciclo de trabajo 100%**	
Polea 1	***Polea 2***	***Polea 1***	***Polea 2***	***Polea 1***	***Polea 2***	***Polea 1***	***Polea 2***	***Polea 1***	***Polea 2***	***Polea 1***	***Polea 2***
108.8	96.2	139.4	119.1	189.9	154.6	205.6	174.7	229.9	196.7	254.3	215.7
111.6	82.3	146.6	121.5	180.8	152.6	220.3	181.1	228.2	196.5	258.1	215.6
109.9	91.3	138.3	116.5	178.5	151.6	209.1	177.6	230.8	194.8	256.5	220.5
112.5	95.6	138.9	114.9	191.8	163.6	211.5	173.1	237.7	187.8	265.2	218.6
101.3	91.5	143.5	121.8	182.2	160.2	209.1	184.5	236.6	188.1	262.4	219.6
108.3	95.7	144.3	121.7	177.7	152.7	210.7	175.1	230.3	189.3	261.4	210.1
104.8	82.7	147.8	118.6	175.3	156.6	208.5	177.4	231.9	187.9	257.9	219.1
100.9	94.1	138.2	119.5	176.5	150.7	218.4	181.1	237.4	198.9	260.6	220.1
104.7	92.5	130.2	111.3	184.5	157.9	202.5	187.1	235.4	185.7	256.2	222.5
109.2	94.5	138.4	118.5	172.3	159.4	216.2	181.7	232.3	192.1	258.2	215.3
P=107.2	**P=91.64**	P=140.6	**P=118.3**	P=180.9	**P=155.9**	P=211.2	**P=179.3**	P=233.1	**P=191.7**	P=259.1	**P=217.7**

Fuente: Elaboración propia

Anexo 4.5

Tabla 4.7. *Valores registrados en el sensor de ancho y circuito de acondicionamiento a diferentes distancias.*

Distancia del sensor al punto de medición (cm)	Voltaje medido a la salida sensor(V)	Corriente generada por el sensor (mA)	Voltaje medido a la entrada del µC
3	1.668	3.801	0.852
4	1.698	3.870	0.898
5	2.081	4.742	1.069
6	2.421	5.517	1.24
7	2.731	6.224	1.415
8	3.05	6.951	1.587
9	3.416	7.785	1.751
10	3.754	8.555	1.917
11	4.096	9.335	2.084
12	4.401	10.030	2.25
13	4.745	10.814	2.421
14	5.063	11.538	2.595
15	5.422	12.356	2.763
16	5.762	13.131	2.93
17	6.102	13.906	3.091
18	6.507	14.829	3.287
19	6.76	15.406	3.198
20	7.13	16.249	3.633
21	7.45	16.978	3.75
22	7.77	17.707	3.931
23	7.99	18.209	4.102
24	8.41	19.166	4.297
25	8.67	19.758	4.395

Fuente: Elaboración propia.

Anexo 4.6

Tabla 4.8. *Valores registrados en el sensor de alto y circuito de acondicionamiento a diferentes distancias.*

Distancia del sensor al punto de medición (cm)	Voltaje medido a la salida sensor (V)	Corriente generada por el sensor (mA)	Voltaje medido a la entrada del µC
3	1.693	3.858	0.855
4	1.701	3.876	0.859
5	2.025	4.615	1.02
6	2.417	5.508	1.202
7	2.742	6.249	1.374
8	3.081	7.021	1.551
9	3.438	7.835	1.718
10	3.773	8.598	1.878
11	4.092	9.325	2.048
12	4.475	10.198	2.225
13	4.852	11.057	2.406
14	5.146	11.727	2.572
15	5.501	12.536	2.731
16	5.809	13.238	2.907
17	6.18	14.084	3.089
18	6.52	14.859	3.255
19	6.83	15.565	3.424
20	7.19	16.386	3.592
21	7.54	17.183	3.772
22	7.92	18.049	3.915
23	8.15	18.573	4.064
24	8.44	19.234	4.25
25	8.85	20.169	4.364

Fuente: Elaboración propia.

Anexo 4.7

Sensor fotoeléctrico de salida bipolar, configurado también en modo PNP como fuente de corriente, con ello, se registraron las siguientes medidas:

Voltaje de salida del Sensor: 11.26V

Resistencia real de carga del sensor: 7.98kohm

Voltaje de entrada al microcontrolador:

- En alta: 4.96V
- En baja: 0.4V

En la figura 4.16 se observa el tiempo de respuesta del sensor al conmutar de un nivel bajo a uno alto cuando detecta una caja en movimiento. El valor de este tiempo es aproximadamente 9us, mucho menor a lo especificado por el fabricante que es 700us.

La figura 4.17 muestra el valor del voltaje en el tiempo cuando se detecta un objeto, se observa que, al igual que en el caso anterior existe una variación de este al comienzo y transcurso del tiempo.

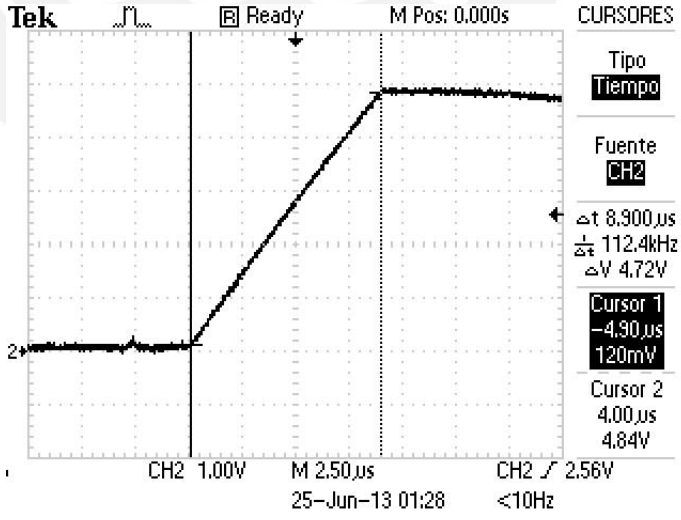


Figura 4.16. *Tiempo de respuesta en flanco de subida del sensor fotoeléctrico 2.*

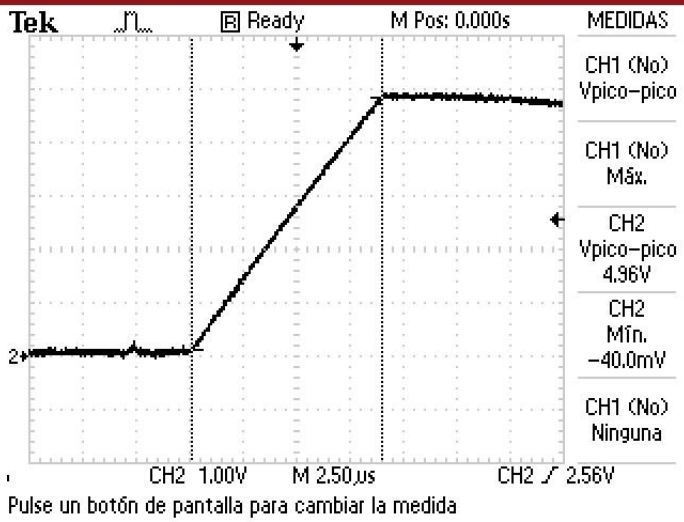


Fig. 4.17. *Medidas registradas a la entrada del microcontrolador con osciloscopio al detectar una caja en movimiento.*

Al evaluar el tiempo de respuesta en el flanco de bajada de este sensor, se encontró que es aproximadamente 228us como se muestra en la figura 4.18, mucho mayor que el identificado en el flanco de subida de éste y del otro sensor, por lo cual, se deduce que introduce error para el cálculo de la medición del largo y la velocidad de la faja transportadora.

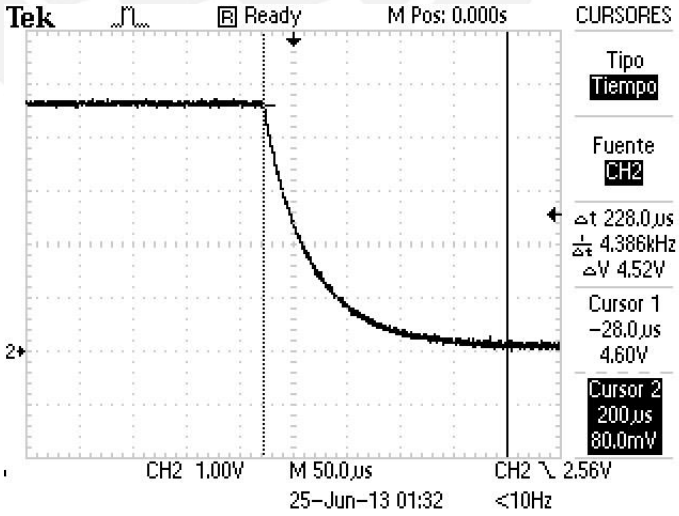


Fig. 4.18. *Tiempo de respuesta en flanco de bajada del sensor fotoeléctrico1.*

Anexo 4.8

CAJA 1: Medidas reales: alto 8.7cm,ancho: 12.1, largo: 20.5cm

Tabla 4.11*. Resultados de medición y porcentaje de error generado en la medición de la caja 1.*

	MEDIDAS			ERROR		
N° CAJA	Alto (cm)	Ancho (cm)	Largo (cm)	Alto (%)	Ancho (%)	Largo (%)
1	8.5	12.3	20.4	-2.30	1.65	-0.49
2	8.5	12.3	20.7	-2.30	1.65	0.98
3	8.5	12.3	20.8	-2.30	1.65	1.46
4	8.5	12.4	20.8	-2.30	2.48	1.46
5	8.5	12.3	21	-2.30	1.65	2.44
6	8.5	12.3	20.7	-2.30	1.65	0.98
7	8.5	12.3	20.8	-2.30	1.65	1.46
8	8.5	12.3	20.6	-2.30	1.65	0.49
9	8.5	12.3	20.4	-2.30	1.65	-0.49
10	8.5	12.3	21	-2.30	1.65	2.44
11	8.5	12.3	19.9	-2.30	1.65	-2.93
12	8.5	12.3	20.6	-2.30	1.65	0.49
13	8.5	12.3	20.9	-2.30	1.65	1.95
14	8.5	12.3	20.8	-2.30	1.65	1.46
15	8.5	12.3	20.4	-2.30	1.65	-0.49

Fuente: Elaboración propia

Los errores máximos y mínimos se muestran en la tabla 4.10.

Tabla 4.12. *Valores máximo y mínimo de error de cada dimensión de la caja 1 expresados en porcentaje.*

	Alto (%)	Ancho (%)	Largo (%)
Error MAX	-2.30	2.48	2.44
Error MIN	-2.30	1.65	-2.93

Fuente: Elaboración propia

Tabla 4.13. *Valores máximo y mínimo de error de cada dimensión de la caja 1 expresados en milímetros.*

	Alto (mm)	Ancho (mm)	Largo (mm)
Error MAX	-2	3	5
Error MIN	-2	2	-6

Fuente: Elaboración propia

CAJA 2: Medidas reales: alto 7.9 cm, ancho 12.9, largo 15.1cm

Tabla 4.14*. Resultados de medición y porcentaje de error generado en la medición de la caja 2.*

	MEDIDAS			ERROR		
N° CAJA	Alto (cm)	Ancho (cm)	Largo (cm)	Alto (%)	Ancho (%)	Largo (%)
1	7.8	13.1	15.1	-1.27	1.55	0.00
2	7.8	13.3	15.3	-1.27	3.10	1.32
3	7.8	13.1	15.5	-1.27	1.55	2.65
4	7.8	13.1	15.6	-1.27	1.55	3.31
5	7.8	13	15.2	-1.27	0.78	0.66
6	7.8	12.9	15.3	-1.27	0.00	1.32
7	7.8	12.9	15.6	-1.27	0.00	3.31
8	7.8	13.1	15.6	-1.27	1.55	3.31
9	7.8	12.9	15.6	-1.27	0.00	3.31
10	7.8	13.3	14.7	-1.27	3.10	-2.65
11	7.7	12.9	15.2	-2.53	0.00	0.66
12	7.7	12.9	14.9	-2.53	0.00	-1.32
13	7.8	12.9	15.5	-1.27	0.00	2.65
14	7.8	12.8	15.3	-1.27	-0.78	1.32
15	7.8	12.8	15.2	-1.27	-0.78	0.66

Fuente: Elaboración propia

Tabla 4.15. *Valores máximo y mínimo de error de cada dimensión de la caja 2 expresados en porcentaje.*

	Alto (%)	Ancho (%)	Largo (%)
Error MAX	-1.27	3.10	3.31
Error MIN	-2.53	-0.78	-2.65

Fuente: Elaboración propia

Tabla 4.16. *Valores máximo y mínimo de error de cada dimensión de la caja 2 expresados en milímetros.*

	Alto (mm)	Ancho (mm)	Largo (mm)
Error MAX	-1	4	5
Error MIN	-2	-1	-4

Fuente: Elaboración propia

CAJA 3: Medidas reales: alto 5 cm, ancho 5cm, largo: 5.2cm

Tabla 4.17*. Resultados de medición y porcentaje de error generado en la medición de la caja 3.*

	MEDIDAS			ERROR		
N° CAJA	Alto (cm)	Ancho (cm)	Largo (cm)	Alto (%)	Ancho (%)	Largo (%)
1	4.4	5.1	5.2	-12.00	2.00	0.00
2	4.4	5.2	5.2	-12.00	4.00	0.00
3	5	5.2	5.2	0.00	4.00	0.00
4	4.4	5.2	5.2	-12.00	4.00	0.00
6	5	5.2	5.1	0.00	4.00	-1.92
7	5	5.2	5	0.00	4.00	-3.85
9	4.4	5.2	5	-12.00	4.00	-3.85
10	5	5.2	5.1	0.00	4.00	-1.92
11	4.9	5.2	5.2	-2.00	4.00	0.00
12	5	5.2	5.2	0.00	4.00	0.00
13	4.4	5.2	5.2	-12.00	4.00	0.00
15	4.4	5.2	5.3	-12.00	4.00	1.92
16	5	5.2	5.2	0.00	4.00	0.00
18	4.8	4.9	5.2	-4.00	-2.00	0.00
19	4.8	5	5.2	-4.00	0.00	0.00

Fuente: Elaboración propia

Tabla 4.18. *Valores máximo y mínimo de error de cada dimensión de la caja 3 expresados en porcentaje.*

	Alto (%)	Ancho (%)	Largo (%)
Error MAX	0.00	4.00	1.92
Error MIN	-12.00	-2.00	-3.85

Fuente: Elaboración propia

Tabla 4.19. *Valores máximo y mínimo de error de cada dimensión de la caja 3 expresados en milímetros.*

	Alto (mm)	Ancho (mm)	Largo (mm)
Error MAX	0	2	1
Error MIN	-6	-1	-2

Fuente: Elaboración propia

Anexo 4.9

Descripción	Cantidad	Precio Unitario (dólares)	Precio Unitario (soles)	Precio Total (soles)
Motor DC con escobillas 20V, 3A 420 RPM	01	-	35.00	35.00
Correa de transmisión	01	-	15.00	15.00
Sensor Ultrasónico U-GAGE S18U Series-Analog (S18UUAR)	02	261.00	722.18	1444.37
Sensor Fotoeléctrico WORLD-BEAM Q12 Series (Q12AB6FF50)	02	76.00	210.29	420.58
Microcontrolador ATmega16 empaque TQFP-44	01	-	28.00	28.00
Cristal resonador de 1Mhz	01		18.00	18.00
Pantalla LCD 4 x 20 con brillo y contraste	01	-	48.00	48.00
Teclado Matricial Alfanumérico	01	-	5.00	5.00
Circuito Integrado MAX490 PDIP-08	02	-	24.00	48.00
Circuito Integrado MAX232 PDIP-16	01	-	6.00	6.00
MOSFETCanal P IRFZ44Nempaque TO-220	01	-	1.50	1.50
Amplificador Operacional LM324 empaque SOP-14	02	-	2.5	2.5
Transistor BJT MMBT3906 empaque SOT-23	02	-	3.5	7.0
Transistor BJT MMBT3904 empaque SOT-23	02	-	3.5	7.0
Transistor BJT 2N3906 empaque TO-92	02	-	0.5	1.0
Optoacoplador 4N35M PDIP-06	01	-	1.0	1.0
Transformador 220VAC - 20VAC 4ª	01	-	25.00	25.00
Transformador 220VAC - 10VAC 1.5A / 8VAC 1.5A	01	-	15.00	15.00
Regulador de voltaje LM7805 empaque TO-220	01	-	1.0	1.0
Regulador de voltaje LM7812 empaque TO-220	02	-	1.0	2.0
Regulador de voltaje LM317 empaque TO-220	01	-	1.5	1.5
Chasis de acero inoxidable	01	-	120.00	120.00
Pulsador de parada de emergencia	01	-	25.00	25.00
Conectores PS/2	04	-	1.0	4.0
Conectores metálicos para chasis 6 pines	02	-	5.0	10.0
Conectores metálicos para chasis 4 pines	02	-	5.0	10.0
Resistencias, condensadores	30	-	0.20	6.0
Diodos	10	-	0.40	4.0
Chasis de la fuente de alimentación	01	-	20.00	20.00
Transistor de potencia PNP para chasis MJ2955	01	-	6.00	6.00
Conversor de puerto serie RS-232 a USB	01	-	18.00	18.00
Trabajo de los tesistas encargados	100 días	-	120.00	12000
			Total	14355.4

Tabla 4.20. *Presupuesto general del sistema implementado.*

ANEXO Nº5: CIRCUITO ESQUEMÁTICO DEL DISEÑO COMPLETO.

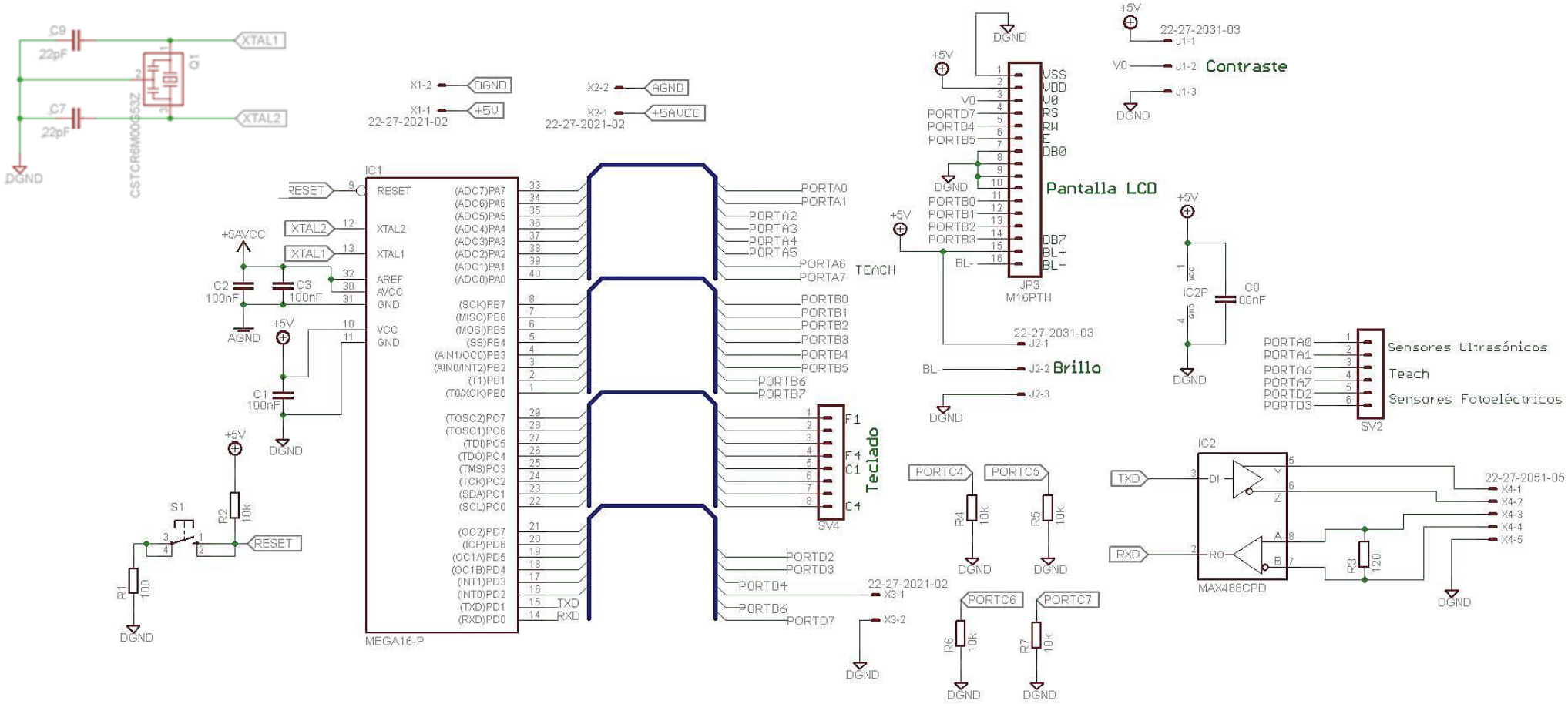


Figura A.1 *Circuito esquemático de la etapa del microcontrolador, pantalla, teclado y transmisión de datos hacia la PC.*

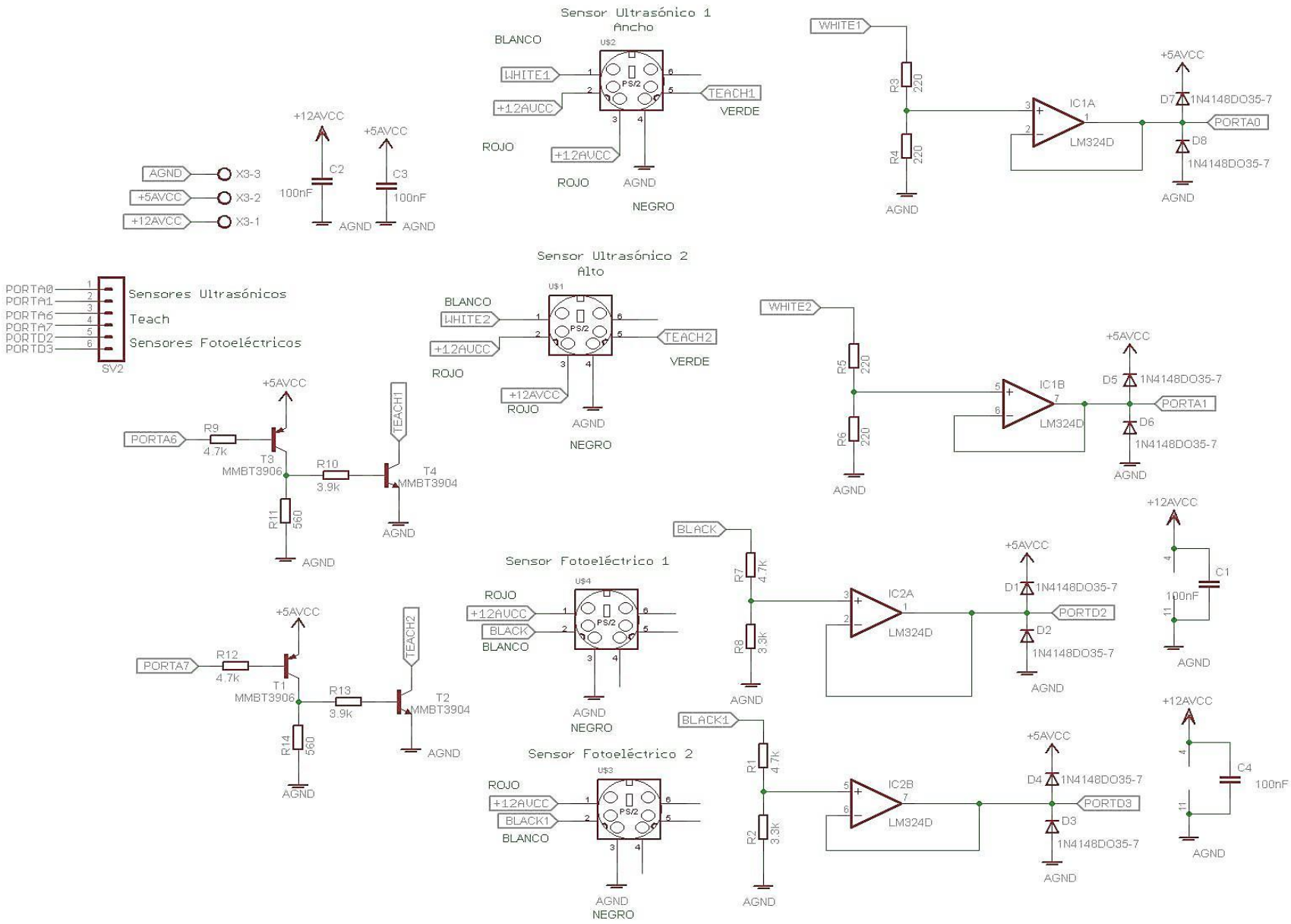


Figura A.2. *Circuito esquemático de acondicionamiento de señales provenientes de los sensores.*

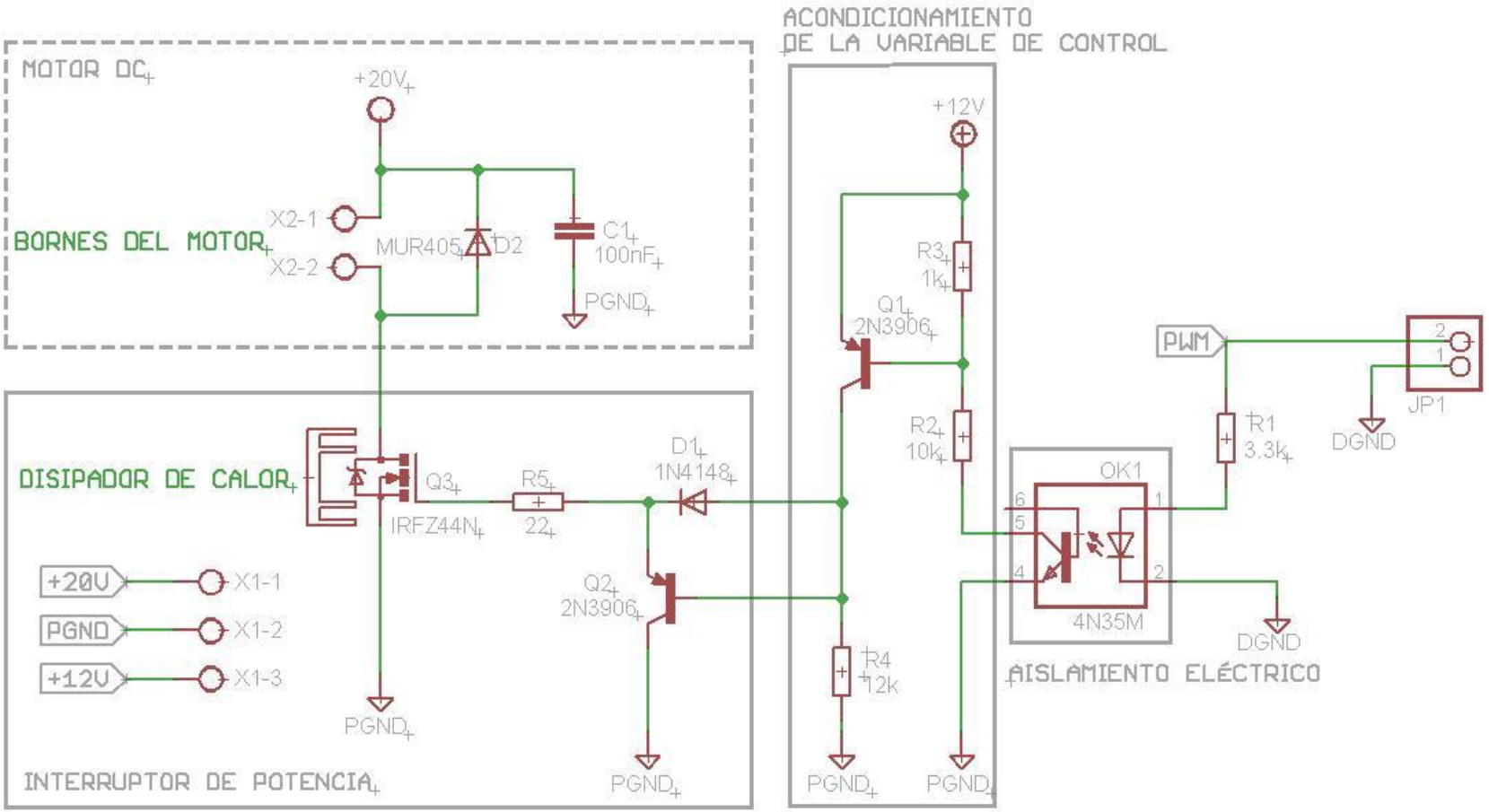


Figura A.3 *Circuito esquemático del circuito excitador del actuador.*

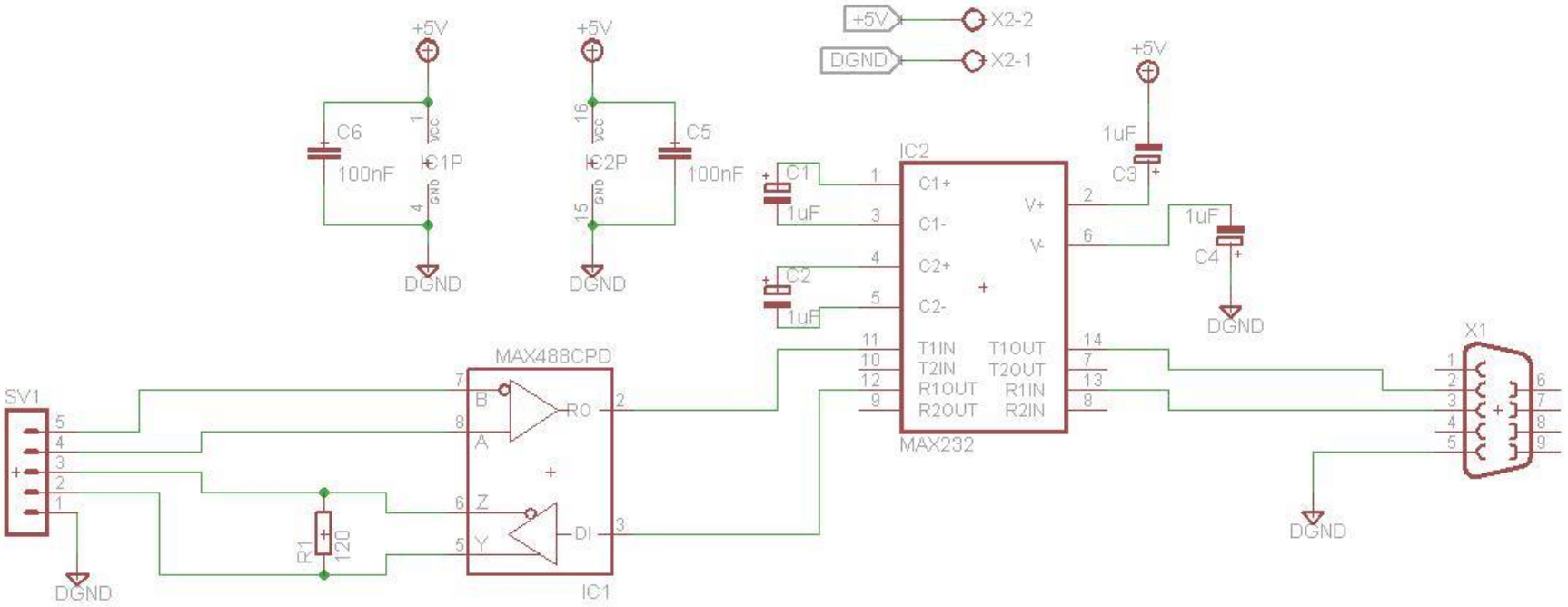


Figura A.4 *Circuito esquemático de la recepción de datos por parte de la PC.*

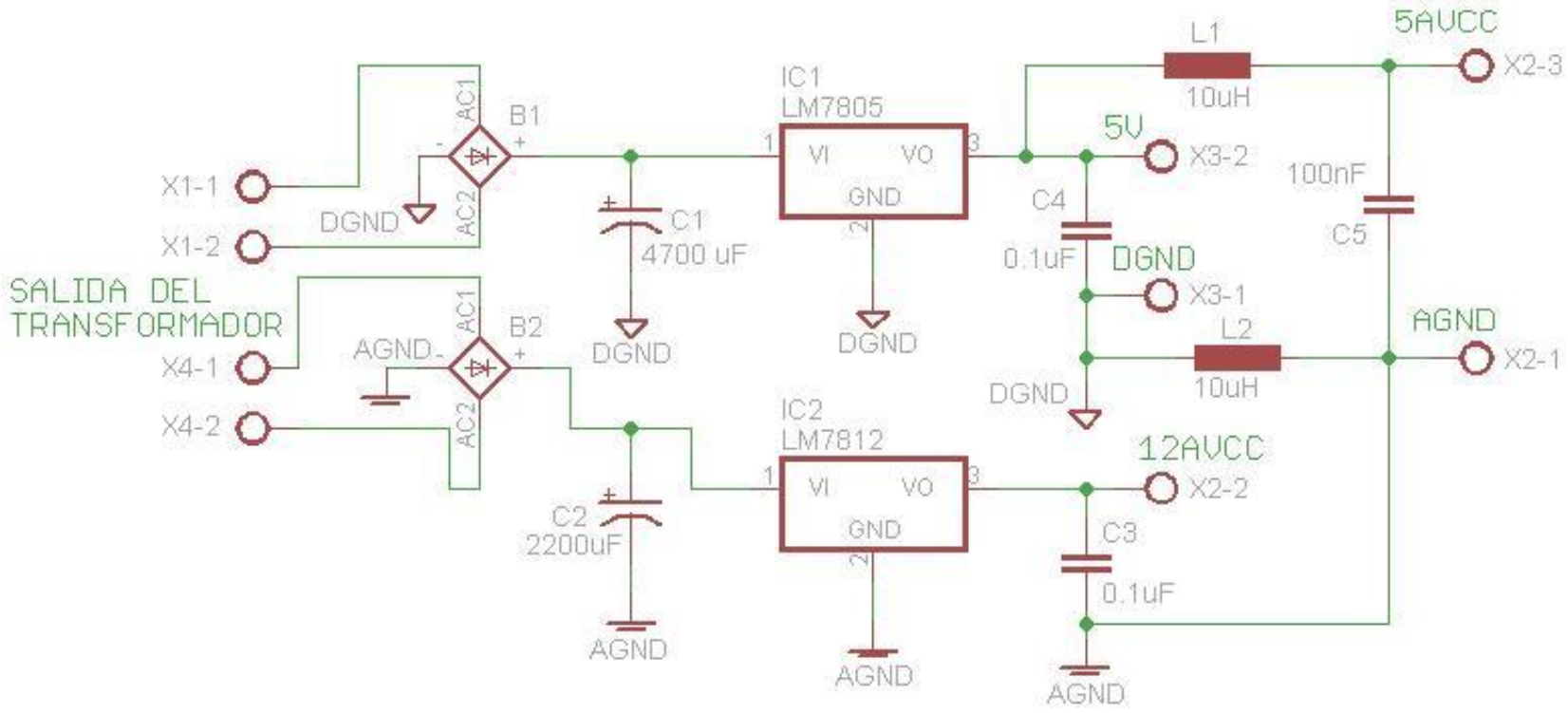


Figura A.5 *Circuito esquemático de la fuente de alimentación Nº 1.*

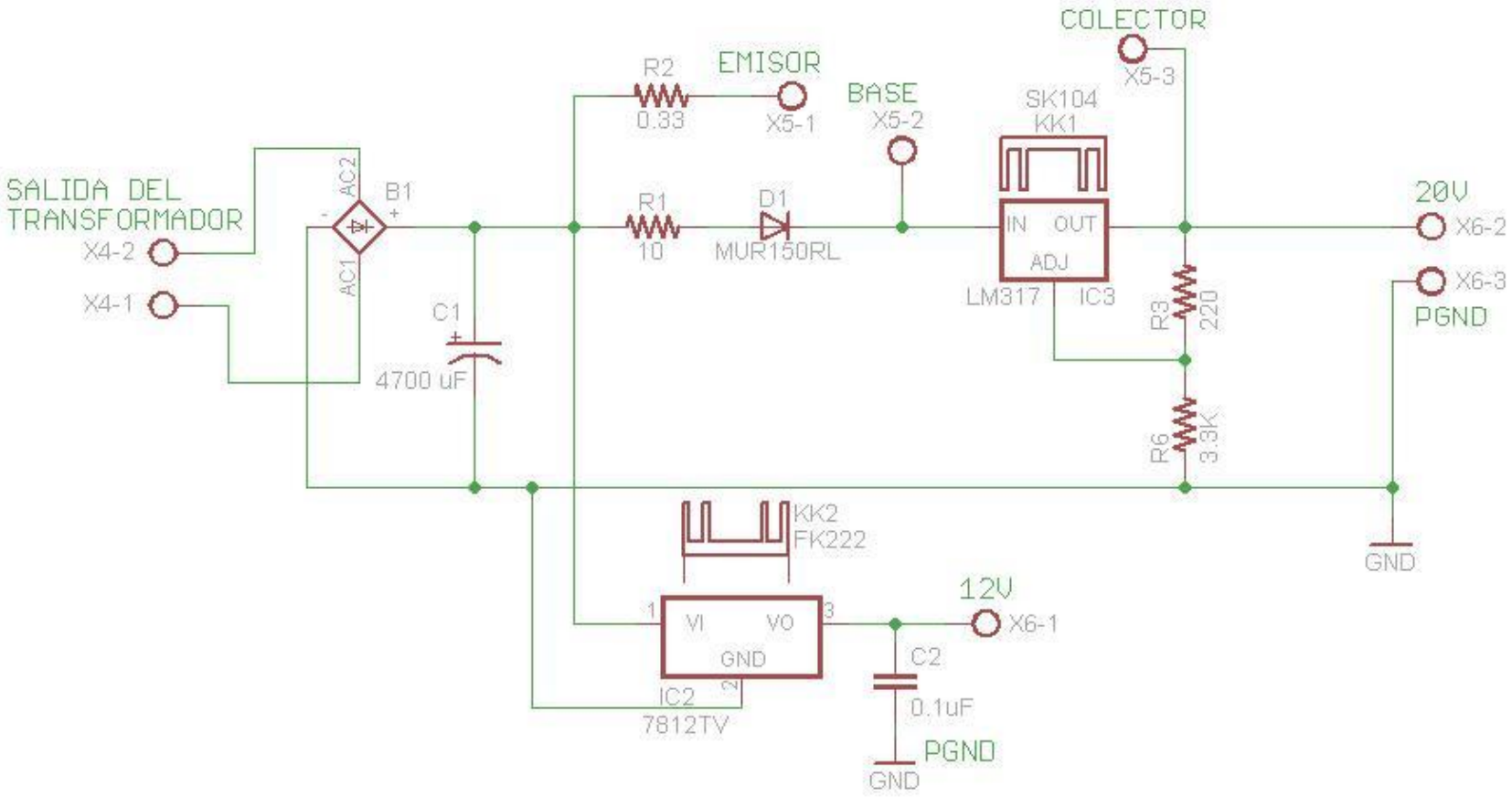


Figura A.6 *Circuito esquemático de la fuente de alimentación Nº 2.*

Switching diode

1N4148 / 1N4150 / 1N4448 / 1N914B

∗This product is available only outside of Japan.

●Applications

High-speed switching

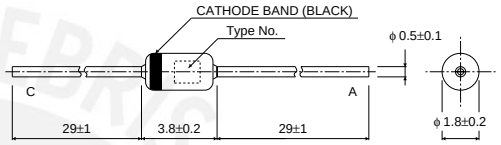
●Features

1) Glass sealed envelope. (GSD)
2) High speed.
3) High reliability.

●Construction

Silicon epitaxial planar

●External dimensions (Units : mm)



ROHM : GSD
EIAJ : –
JEDEC : DO-35

●Absolute maximum ratings (Ta = 25°C)

Type	V_{RM} (V)	V_R (V)	I_{FM} (mA)	I_O (mA)	I_F (mA)	I_{FSM} 1μs (A)	P (mW)	Tj (°C)	Topr (°C)	Tstg (°C)
1N4148	100	75	450	150	200	2	500	200	–65~+200	–65~+200
1N4150	50	50	600	200	250	4	500	200	–65~+200	–65~+200
1N4448 (1N914B)	100	75	450	150	200	2	500	200	–65~+200	–65~+200

●Electrical characteristics (Ta = 25°C)

Type	V_F (V) @0.1mA	@0.25mA	@1mA	@2mA	@5mA	@10mA	@20mA	@30mA	@50mA	@100mA	@200mA	@250mA	BV (V) Min. @5μA	@100μA	I_R (μA) Max. @25°C	V_R (V)	@150°C	V_R (V)	C_t (pF) V_R=0 f=1MHz	t_{rr} (ns) V_R=6V I_F=10mA R_L=100Ω
1N4148						1.0							75	100	0.025	20	50.0	20	4	4
															5.0	75				
1N4150			0.54			0.66			0.76	0.82	0.87		–	50	0.1	50	100.0	50	2.5	4
			0.62			0.74			0.86	0.92	1.0									
1N4448 (IN914B)					0.62								–	100	0.025	20	50.0	20	4	4
					0.72					1.0					5.0	75				

The upper figure is the minimum V_F and the lower figure is the maximum V_F value.

●**Electrical characteristic curves** (Ta = 25°C)

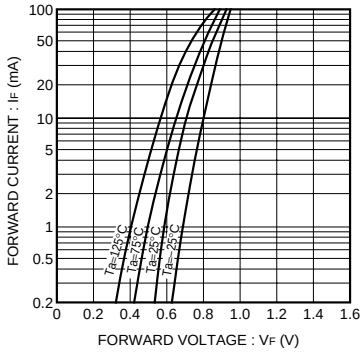


Fig. 1 Forward characteristics

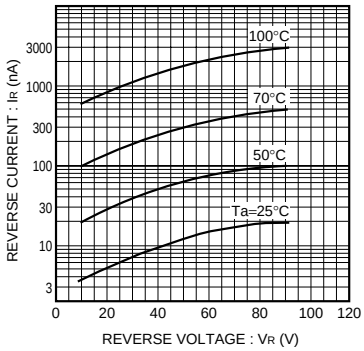


Fig. 2 Reverse characteristics

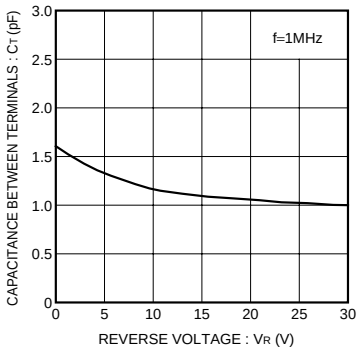


Fig. 3 Capacitance between terminals characteristics

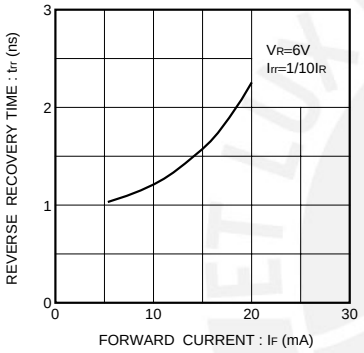


Fig. 4 Reverse recovery time characteristics

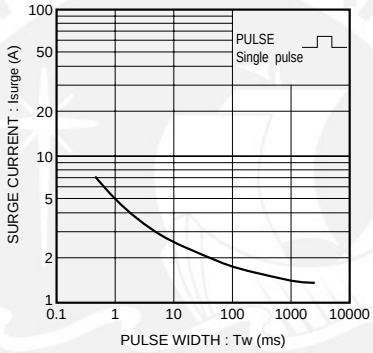


Fig. 5 Surge current characteristics

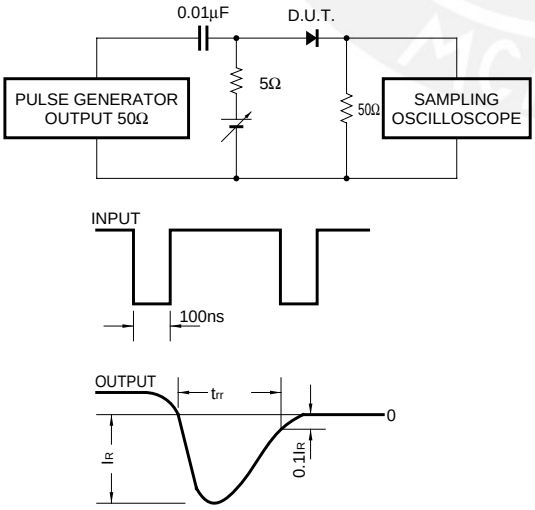


Fig. 6 Reverse recovery time (t_{rr}) measurement circuit

2N3055(NPN), MJ2955(PNP)

Preferred Device

Complementary Silicon Power Transistors

Complementary silicon power transistors are designed for general−purpose switching and amplifier applications.

Features

- DC Current Gain − h_{FE} = 20−70 @ I_C = 4 Adc
- Collector−Emitter Saturation Voltage − $V_{CE(sat)}$ = 1.1 Vdc (Max) @ I_C = 4 Adc
- Excellent Safe Operating Area
- Pb−Free Packages are Available*

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector−Emitter Voltage	V_{CEO}	60	Vdc
Collector−Emitter Voltage	V_{CER}	70	Vdc
Collector−Base Voltage	V_{CB}	100	Vdc
Emitter−Base Voltage	V_{EB}	7	Vdc
Collector Current − Continuous	I_C	15	Adc
Base Current	I_B	7	Adc
Total Power Dissipation @ T_C = 25°C Derate Above 25°C	P_D	115 0.657	W W/°C
Operating and Storage Junction Temperature Range	T_J, T_{stg}	−65 to +200	°C

Maximum ratings are those values beyond which device damage can occur. Maximum ratings applied to the device are individual stress limit values (not normal operating conditions) and are not valid simultaneously. If these limits are exceeded, device functional operation is not implied, damage may occur and reliability may be affected.

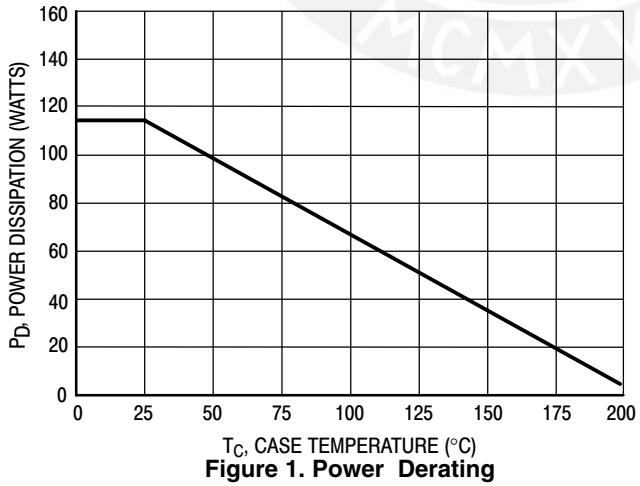


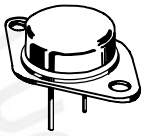
Figure 1. Power Derating

*For additional information on our Pb−Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

ON Semiconductor®

http://onsemi.com

15 AMPERE POWER TRANSISTORS COMPLEMENTARY SILICON 60 VOLTS, 115 WATTS



TO−204AA (TO−3)
CASE 1−07
STYLE 1

MARKING DIAGRAM

xxxx55 = Device Code
xxxx = 2N30 or MJ20
G = Pb−Free Package
A = Location Code
YY = Year
WW = Work Week
MEX = Country of Orgin

ORDERING INFORMATION

Device	Package	Shipping
2N3055	TO−204AA	100 Units / Tray
2N3055G	TO−204AA (Pb−Free)	100 Units / Tray
MJ2955	TO−204AA	100 Units / Tray
MJ2955G	TO−204AA (Pb−Free)	100 Units / Tray

Preferred devices are recommended choices for future use and best overall value.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction−to−Case	$R_{\theta JC}$	1.52	°C/W

ELECTRICAL CHARACTERISTICS (T_C = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
OFF CHARACTERISTICS*				
Collector−Emitter Sustaining Voltage (Note 1) (I_C = 200 mAdc, I_B = 0)	$V_{CEO(sus)}$	60	−	Vdc
Collector−Emitter Sustaining Voltage (Note 1) (I_C = 200 mAdc, R_{BE} = 100 Ω)	$V_{CER(sus)}$	70	−	Vdc
Collector Cutoff Current (V_{CE} = 30 Vdc, I_B = 0)	I_{CEO}	−	0.7	mAdc
Collector Cutoff Current (V_{CE} = 100 Vdc, $V_{BE(off)}$ = 1.5 Vdc) (V_{CE} = 100 Vdc, $V_{BE(off)}$ = 1.5 Vdc, T_C = 150°C)	I_{CEX}	 − −	 1.0 5.0	mAdc
Emitter Cutoff Current (V_{BE} = 7.0 Vdc, I_C = 0)	I_{EBO}	−	5.0	mAdc
ON CHARACTERISTICS* (Note 1)				
DC Current Gain (I_C = 4.0 Adc, V_{CE} = 4.0 Vdc) (I_C = 10 Adc, V_{CE} = 4.0 Vdc)	h_{FE}	 20 5.0	 70 −	−
Collector−Emitter Saturation Voltage (I_C = 4.0 Adc, I_B = 400 mAdc) (I_C = 10 Adc, I_B = 3.3 Adc)	$V_{CE(sat)}$	−	 1.1 3.0	Vdc
Base−Emitter On Voltage (I_C = 4.0 Adc, V_{CE} = 4.0 Vdc)	$V_{BE(on)}$	−	1.5	Vdc
SECOND BREAKDOWN				
Second Breakdown Collector Current with Base Forward Biased (V_{CE} = 40 Vdc, t = 1.0 s, Nonrepetitive)	$I_{s/b}$	2.87	−	Adc
DYNAMIC CHARACTERISTICS				
Current Gain − Bandwidth Product (I_C = 0.5 Adc, V_{CE} = 10 Vdc, f = 1.0 MHz)	f_T	2.5	−	MHz
*Small−Signal Current Gain (I_C = 1.0 Adc, V_{CE} = 4.0 Vdc, f = 1.0 kHz)	h_{fe}	15	120	−
*Small−Signal Current Gain Cutoff Frequency (V_{CE} = 4.0 Vdc, I_C = 1.0 Adc, f = 1.0 kHz)	f_{hfe}	10	−	kHz

*Indicates Within JEDEC Registration. (2N3055)
1. Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2.0%.

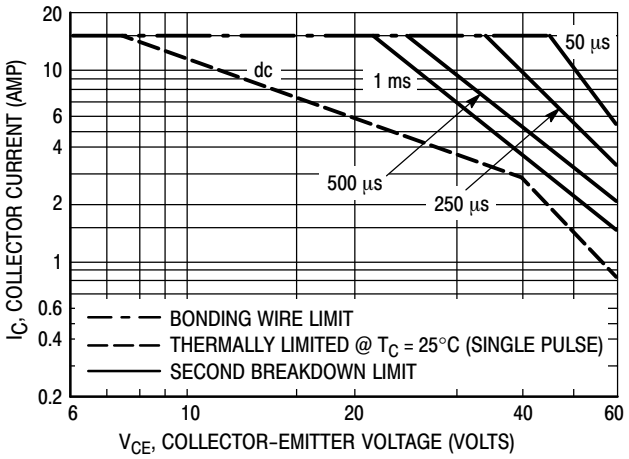


Figure 2. Active Region Safe Operating Area

There are two limitations on the power handling ability of a transistor: average junction temperature and second breakdown. Safe operating area curves indicate $I_C - V_{CE}$ limits of the transistor that must be observed for reliable operation; i.e., the transistor must not be subjected to greater dissipation than the curves indicate.

The data of Figure 2 is based on T_C = 25°C; $T_{J(pk)}$ is variable depending on power level. Second breakdown pulse limits are valid for duty cycles to 10% but must be derated for temperature according to Figure 1.

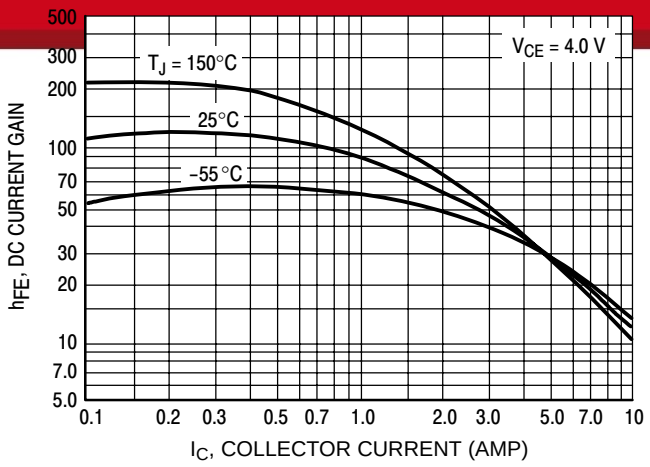


Figure 3. DC Current Gain, 2N3055 (NPN)

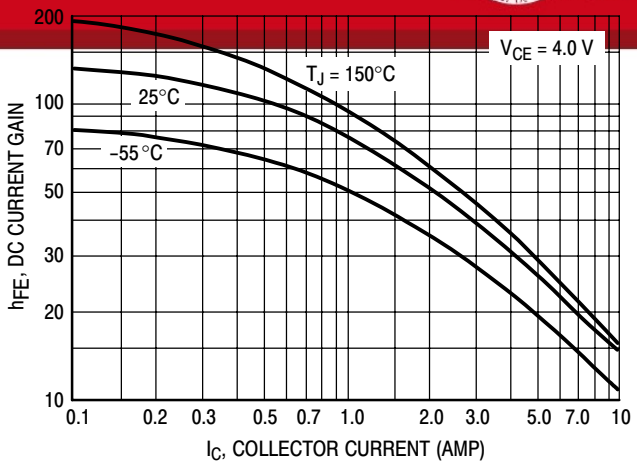


Figure 4. DC Current Gain, MJ2955 (PNP)

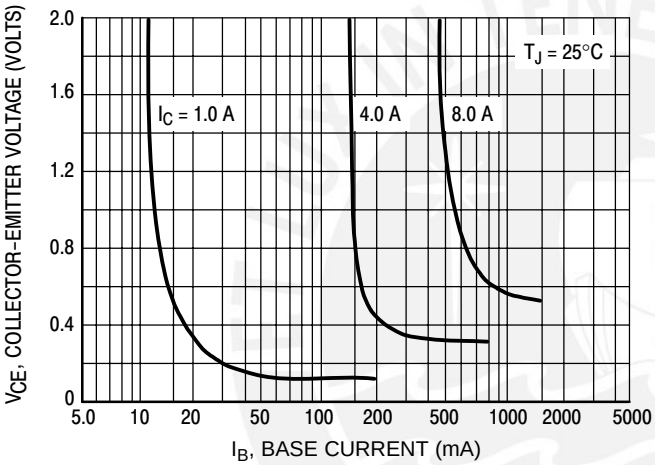


Figure 5. Collector Saturation Region, 2N3055 (NPN)

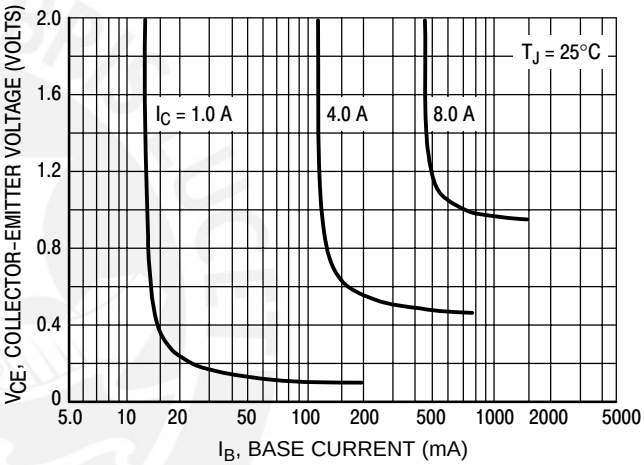


Figure 6. Collector Saturation Region, MJ2955 (PNP)

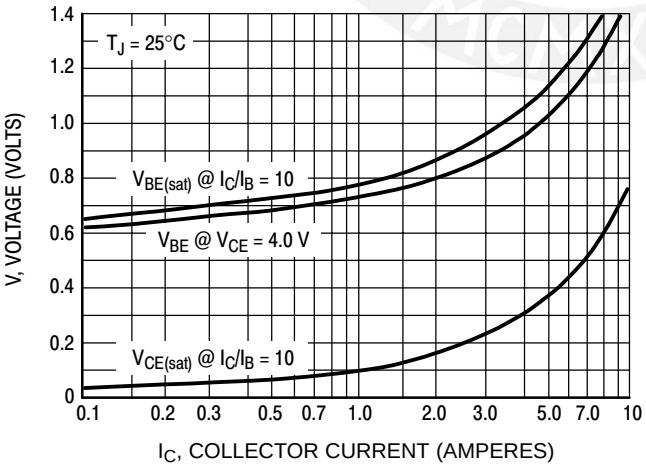


Figure 7. "On" Voltages, 2N3055 (NPN)

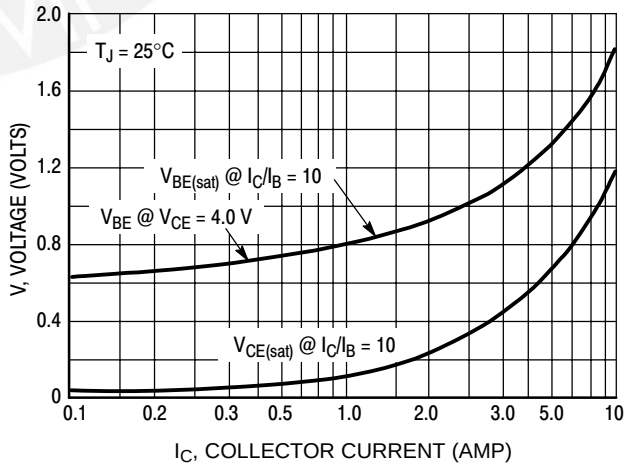
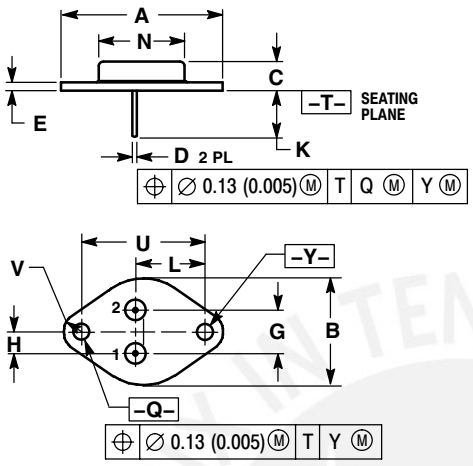


Figure 8. "On" Voltages, MJ2955 (PNP)

PACKAGE DIMENSIONS

TO−204 (TO−3)
CASE 1−07
ISSUE Z



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. ALL RULES AND NOTES ASSOCIATED WITH REFERENCED TO−204AA OUTLINE SHALL APPLY.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.550 REF		39.37 REF	
B	---	1.050	---	26.67
C	0.250	0.335	6.35	8.51
D	0.038	0.043	0.97	1.09
E	0.055	0.070	1.40	1.77
G	0.430 BSC		10.92 BSC	
H	0.215 BSC		5.46 BSC	
K	0.440	0.480	11.18	12.19
L	0.665 BSC		16.89 BSC	
N	---	0.830	---	21.08
Q	0.151	0.165	3.84	4.19
U	1.187 BSC		30.15 BSC	
V	0.131	0.188	3.33	4.77

STYLE 1:
PIN 1. BASE
2. EMITTER
CASE: COLLECTOR

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
P.O. Box 61312, Phoenix, Arizona 85082−1312 USA
Phone: 480−829−7710 or 800−344−3860 Toll Free USA/Canada
Fax: 480−829−7709 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada

Japan: ON Semiconductor, Japan Customer Focus Center
2−9−1 Kamimeguro, Meguro−ku, Tokyo, Japan 153−0051
Phone: 81−3−5773−3850

ON Semiconductor Website: http://onsemi.com

Order Literature: http://www.onsemi.com/litorder

For additional information, please contact your local Sales Representative.

FAIRCHILD SEMICONDUCTOR®

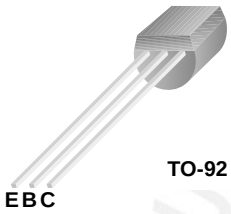
October 2011

2N3904 / MMBT3904 / PZT3904
NPN General Purpose Amplifier

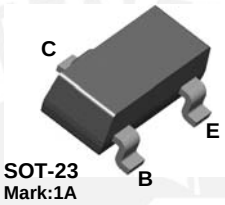
Features

- This device is designed as a general purpose amplifier and switch.
- The useful dynamic range extends to 100 mA as a switch and to 100 MHz as an amplifier.

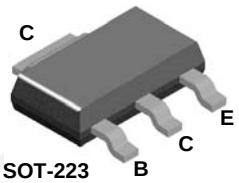
2N3904



MMBT3904



PZT3904



Absolute Maximum Ratings* T_a = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	60	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current - Continuous	200	mA
T_J, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	°C

* These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:

1) These ratings are based on a maximum junction temperature of 150 degrees C.

2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Thermal Characteristics T_a = 25°C unless otherwise noted

Symbol	Parameter	Max.			Units
		2N3904	*MMBT3904	**PZT3904	
P_D	Total Device Dissipation Derate above 25°C	625 5.0	350 2.8	1,000 8.0	mW mW/°C
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3			°C/W
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	357	125	°C/W

* Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06".

** Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 cm^2.

Electrical Characteristics T_a = 25°C unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
OFF CHARACTERISTICS					
$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage	I_C = 1.0mA, I_B = 0	40		V
$V_{(BR)CBO}$	Collector-Base Breakdown Voltage	I_C = 10μA, I_E = 0	60		V
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage	I_E = 10μA, I_C = 0	6.0		V
I_{BL}	Base Cutoff Current	V_{CE} = 30V, V_{EB} = 3V		50	nA
I_{CEX}	Collector Cutoff Current	V_{CE} = 30V, V_{EB} = 3V		50	nA
ON CHARACTERISTICS*					
h_{FE}	DC Current Gain	I_C = 0.1mA, V_{CE} = 1.0V I_C = 1.0mA, V_{CE} = 1.0V I_C = 10mA, V_{CE} = 1.0V I_C = 50mA, V_{CE} = 1.0V I_C = 100mA, V_{CE} = 1.0V	40 70 100 60 30	300	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	I_C = 10mA, I_B = 1.0mA I_C = 50mA, I_B = 5.0mA		0.2 0.3	V V
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	I_C = 10mA, I_B = 1.0mA I_C = 50mA, I_B = 5.0mA	0.65	0.85 0.95	V V
SMALL SIGNAL CHARACTERISTICS					
f_T	Current Gain - Bandwidth Product	I_C = 10mA, V_{CE} = 20V, f = 100MHz	300		MHz
C_{obo}	Output Capacitance	V_{CB} = 5.0V, I_E = 0, f = 1.0MHz		4.0	pF
C_{ibo}	Input Capacitance	V_{EB} = 0.5V, I_C = 0, f = 1.0MHz		8.0	pF
NF	Noise Figure	I_C = 100μA, V_{CE} = 5.0V, R_S = 1.0kΩ, f = 10Hz to 15.7kHz		5.0	dB
SWITCHING CHARACTERISTICS					
t_d	Delay Time	V_{CC} = 3.0V, V_{BE} = 0.5V I_C = 10mA, I_{B1} = 1.0mA		35	ns
t_r	Rise Time			35	ns
t_s	Storage Time	V_{CC} = 3.0V, I_C = 10mA, I_{B1} = I_{B2} = 1.0mA		200	ns
t_f	Fall Time			50	ns

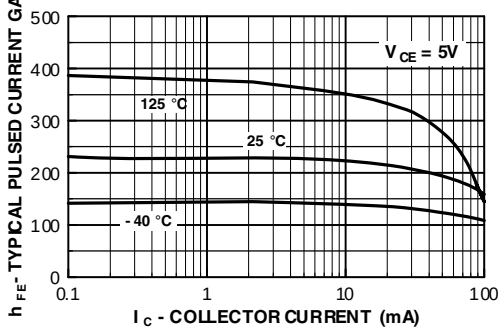
* Pulse Test: Pulse Width ≤ 300μs, Duty Cycle ≤ 2.0%

Ordering Information

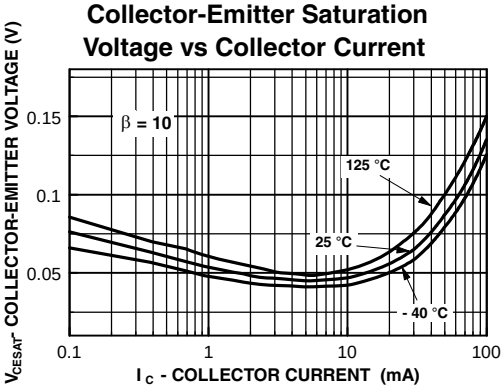
Part Number	Marking	Package	Packing Method	Pack Qty
2N3904BU	2N3904	TO-92	BULK	10000
2N3904TA	2N3904	TO-92	AMMO	2000
2N3904TAR	2N3904	TO-92	AMMO	2000
2N3904TF	2N3904	TO-92	TAPE REEL	2000
2N3904TFR	2N3904	TO-92	TAPE REEL	2000
MMBT3904	1A	SOT-23	TAPE REEL	3000
MMBT3904_D87Z	1A	SOT-23	TAPE REEL	10000
PZT3904	3904	SOT-223	TAPE REEL	2500

Typical Performance Characteristics

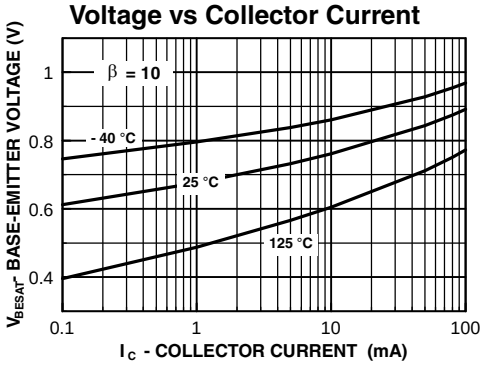
Typical Pulsed Current Gain vs Collector Current



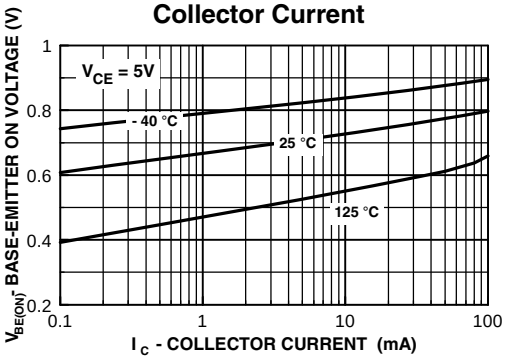
Collector-Emitter Saturation Voltage vs Collector Current



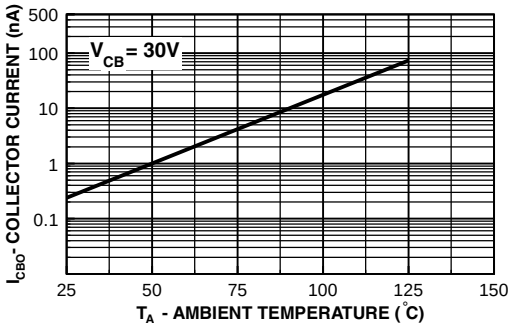
Base-Emitter Saturation Voltage vs Collector Current



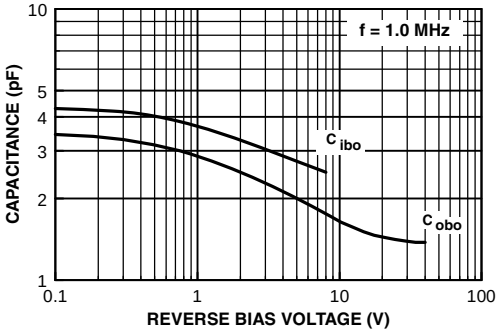
Base-Emitter ON Voltage vs Collector Current



Collector-Cutoff Current vs Ambient Temperature

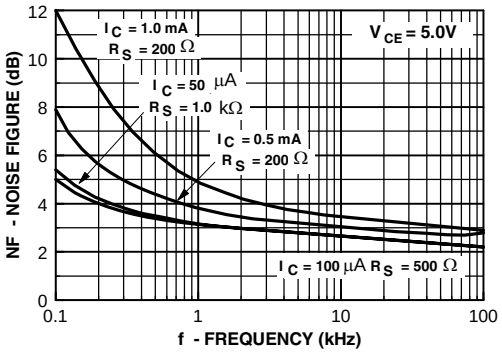


Capacitance vs Reverse Bias Voltage

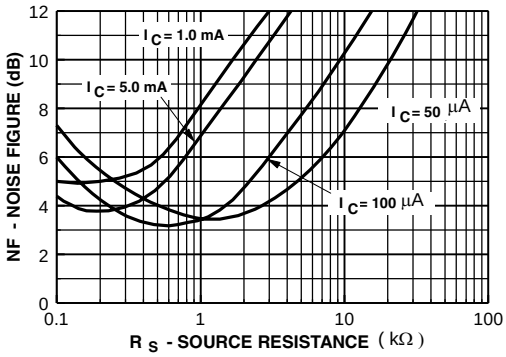


Typical Performance Characteristics (continued)

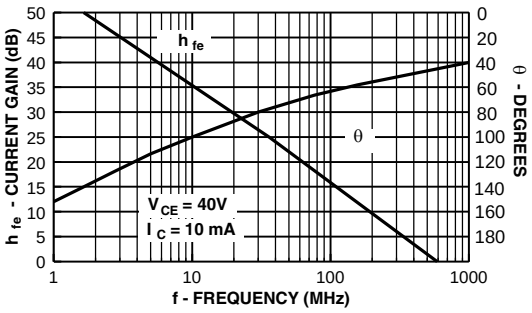
Noise Figure vs Frequency



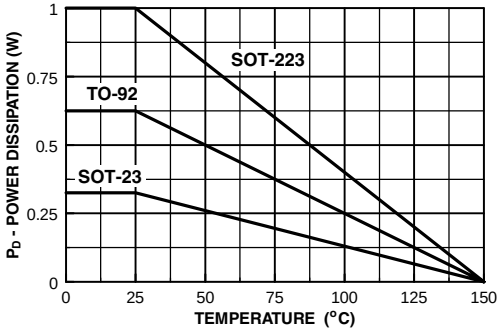
Noise Figure vs Source Resistance



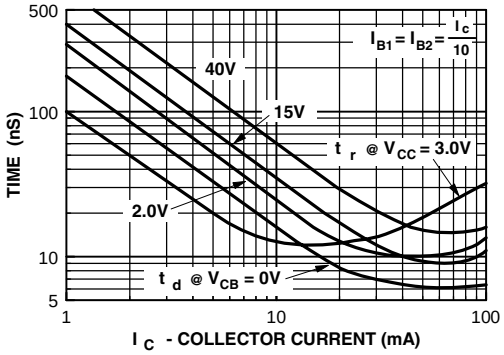
Current Gain and Phase Angle vs Frequency



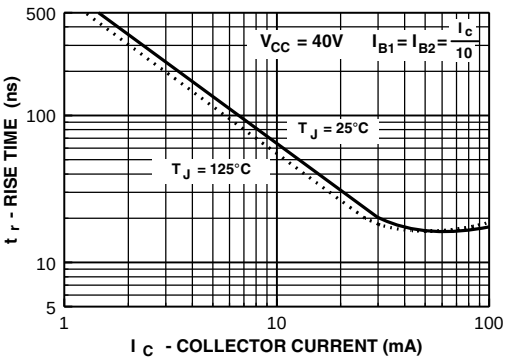
Power Dissipation vs Ambient Temperature



Turn-On Time vs Collector Current

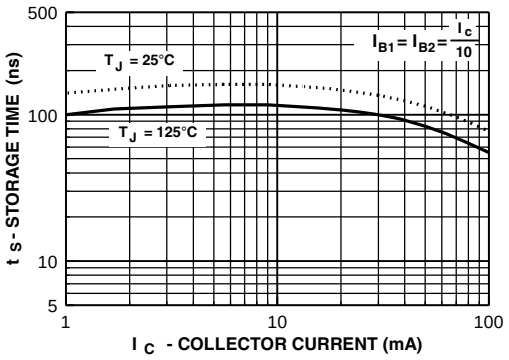


Rise Time vs Collector Current

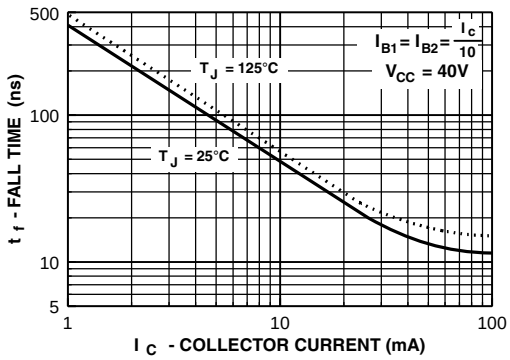


Typical Performance Characteristics (continued)

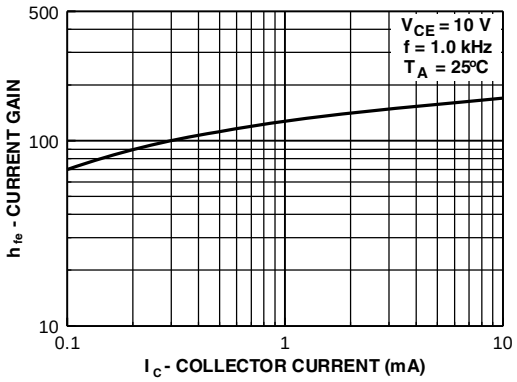
Storage Time vs Collector Current



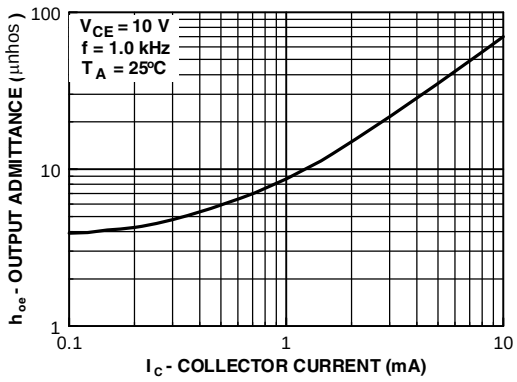
Fall Time vs Collector Current



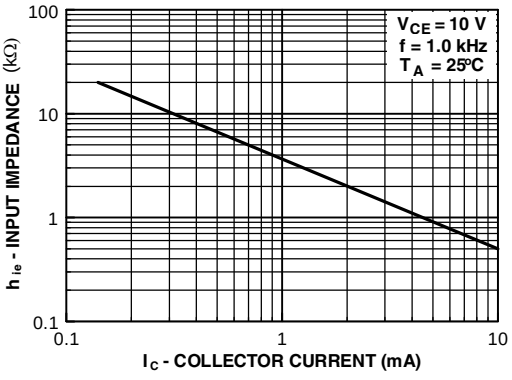
Current Gain



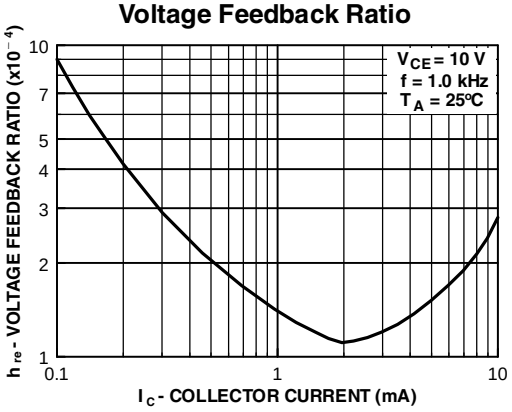
Output Admittance



Input Impedance



Voltage Feedback Ratio



Test Circuits

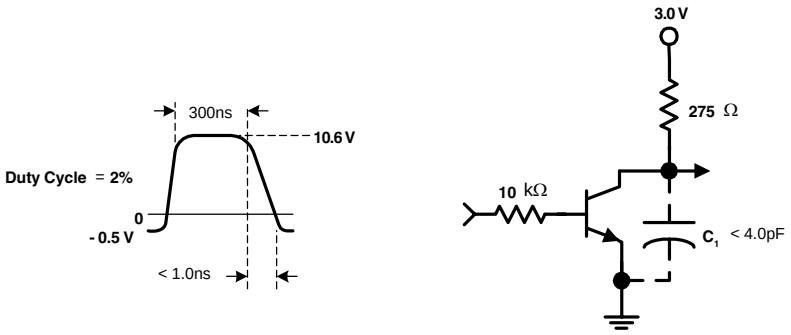


FIGURE 1: Delay and Rise Time Equivalent Test Circuit

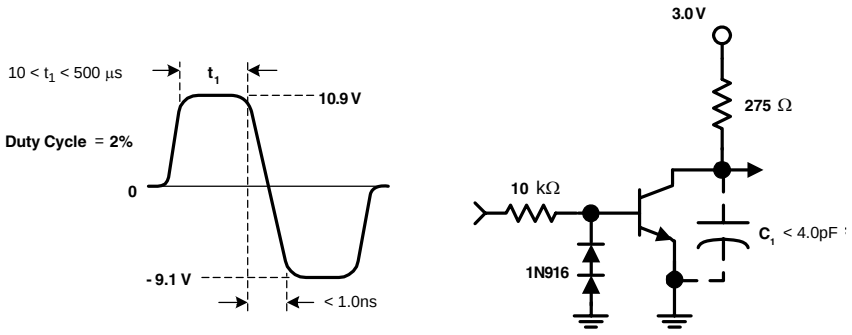


FIGURE 2: Storage and Fall Time Equivalent Test Circuit

FAIRCHILD SEMICONDUCTOR®

TRADEMARKS

The following includes registered and unregistered trademarks and service marks, owned by Fairchild Semiconductor and/or its global subsidiaries, and is not intended to be an exhaustive list of all such trademarks.

2Cool™
AccuPower™
Auto-SPM™
AX-CAP™*
BitSiC®
Build it Now™
CorePLUS™
CorePOWER™
CROSSVOLT™
CTL™
Current Transfer Logic™
DEUXPEED®
Dual Cool™
EcoSPARK®
EfficientMax™
ESBC™
F®
Fairchild®
Fairchild Semiconductor®
FACT Quiet Series™
FACT®
FAST®
FastvCore™
FETBench™
FlashWriter®*
FPS™
F-PFS™
FRFET®
Global Power Resource℠
Green FPS™
Green FPS™ e-Series™
G*max*™
GTO™
IntelliMAX™
ISOPLANAR™
Making Small Speakers Sound Louder and Better™
MegaBuck™
MICROCOUPLER™
MicroFET™
MicroPak™
MicroPak2™
MillerDrive™
MotionMax™
Motion-SPM™
mWSaver™
OptoHiT™
OPTOLOGIC®
OPTOPLANAR®
PDP SPM™
Power-SPM™
PowerTrench®
PowerXS™
Programmable Active Droop™
QFET®
QS™
Quiet Series™
RapidConfigure™
™
Saving our world, 1mW/W/kW at a time™
SignalWise™
SmartMax™
SMART START™
SPM®
STEALTH™
SuperFET®
SuperSOT™-3
SuperSOT™-6
SuperSOT™-8
SupreMOS®
SyncFET™
Sync-Lock™
SYSTEM GENERAL®*
The Power Franchise®
the power franchise
TinyBoost™
TinyBuck™
TinyCalc™
TinyLogic®
TINYOPTO™
TinyPower™
TinyPWM™
TinyWire™
TranSiC®
TriFault Detect™
TRUECURRENT®*
µSerDes™
SerDes
UHC®
Ultra FRFET™
UniFET™
VCX™
VisualMax™
VoltagePlus™
XS™

* Trademarks of System General Corporation, used under license by Fairchild Semiconductor.

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION, OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS. THESE SPECIFICATIONS DO NOT EXPAND THE TERMS OF FAIRCHILD'S WORLDWIDE TERMS AND CONDITIONS, SPECIFICALLY THE WARRANTY THEREIN, WHICH COVERS THESE PRODUCTS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support, device, or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

ANTI-COUNTERFEITING POLICY

Fairchild Semiconductor Corporation's Anti-Counterfeiting Policy. Fairchild's Anti-Counterfeiting Policy is also stated on our external website, www.fairchildsemi.com, under Sales Support.

Counterfeiting of semiconductor parts is a growing problem in the industry. All manufacturers of semiconductor products are experiencing counterfeiting of their parts. Customers who inadvertently purchase counterfeit parts experience many problems such as loss of brand reputation, substandard performance, failed applications, and increased cost of production and manufacturing delays. Fairchild is taking strong measures to protect ourselves and our customers from the proliferation of counterfeit parts. Fairchild strongly encourages customers to purchase Fairchild parts either directly from Fairchild or from Authorized Fairchild Distributors who are listed by country on our web page cited above. Products customers buy either from Fairchild directly or from Authorized Fairchild Distributors are genuine parts, have full traceability, meet Fairchild's quality standards for handling and storage and provide access to Fairchild's full range of up-to-date technical and product information. Fairchild and our Authorized Distributors will stand behind all warranties and will appropriately address any warranty issues that may arise. Fairchild will not provide any warranty coverage or other assistance for parts bought from Unauthorized Sources. Fairchild is committed to combat this global problem and encourage our customers to do their part in stopping this practice by buying direct or from authorized distributors.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative / In Design	Datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	Datasheet contains preliminary data; supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve design.
No Identification Needed	Full Production	Datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve the design.
Obsolete	Not In Production	Datasheet contains specifications on a product that is discontinued by Fairchild Semiconductor. The datasheet is for reference information only.

Rev. I57

FAIRCHILD SEMICONDUCTOR®

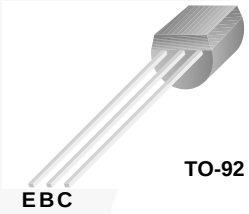
July 2013

2N3906 / MMBT3906 / PZT3906
PNP General Purpose Amplifier

Description

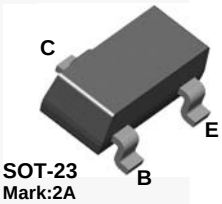
This device is designed for general purpose amplifier and switching applications at collector currents of 10 mA to 100 mA.

2N3906



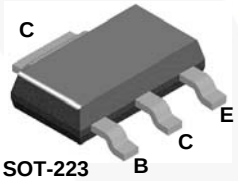
TO-92

MMBT3906



SOT-23
Mark:2A

PZT3906



SOT-223

Ordering Information

Part Number	Marking	Package	Packing Method	Pack Quantity
2N3906BU	2N3906	TO-92	BULK	10000
2N3906TA	2N3906	TO-92	AMMO	2000
2N3906TAR	2N3906	TO-92	AMMO	2000
2N3906TF	2N3906	TO-92	TAPE REEL	2000
2N3906TFR	2N3906	TO-92	TAPE REEL	2000
MMBT3906	2A	SOT-23	TAPE REEL	3000
PZT3906	3906	SOT-223	TAPE REEL	2500

Absolute Maximum Ratings(1)

Stresses exceeding the absolute maximum ratings may damage the device. The device may not function or be operable above the recommended operating conditions and stressing the parts to these levels is not recommended. In addition, extended exposure to stresses above the recommended operating conditions may affect device reliability. The absolute maximum ratings are stress ratings only. Values are at T_A = 25°C unless otherwise noted.

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	-40	V
V_{CBO}	Collector-Base Voltage	-40	V
V_{EBO}	Emitter-Base Voltage	-5.0	V
I_C	Collector Current - Continuous	-200	mA
T_J, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	°C

Note:

1. These ratings are based on a maximum junction temperature of 150 °C.
   These are steady-state limits. Fairchild Semiconductor should be consulted on applications involving pulsed or low-duty cycle operations.

Thermal Characteristics

Values are at T_A = 25°C unless otherwise noted.

Symbol	Parameter	Max.			Units
		2N3906	MMBT3906(2)	PZT3906(3)	
P_D	Total Device Dissipation	625	350	1,000	mW
	Derate above 25°C	5.0	2.8	8.0	mW/°C
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3			°C/W
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	357	125	°C/W

Notes:

2. Device mounted on FR-4 PCB 1.6 inch X 1.6 inch X 0.06 inch.
3. Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead minimum 6 cm^2.

Electrical Characteristics

Values are at T_A = 25°C unless otherwise noted.

Symbol	Parameter	Test Condition	Min.	Max.	Units
OFF CHARACTERISTICS					
$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage[4]	I_C = -1.0 mA, I_B = 0	-40		V
$V_{(BR)CBO}$	Collector-Base Breakdown Voltage	I_C = -10 μA, I_E = 0	-40		V
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage	I_E = -10 μA, I_C = 0	-5.0		V
I_{BL}	Base Cutoff Current	V_{CE} = -30 V, V_{BE} = 3.0 V		-50	nA
I_{CEX}	Collector Cutoff Current	V_{CE} = -30 V, V_{BE} = 3.0 V		-50	nA
ON CHARACTERISTICS					
h_{FE}	DC Current Gain[4]	I_C = -0.1 mA, V_{CE} = -1.0 V	60		
		I_C = -1.0 mA, V_{CE} = -1.0 V	80		
		I_C = -10 mA, V_{CE} = -1.0 V	100	300	
		I_C = -50 mA, V_{CE} = -1.0 V	60		
		I_C = -100 mA, V_{CE} = -1.0V	30		
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	I_C = -10 mA, I_B = -1.0 mA		-0.25	V
		I_C = -50 mA, I_B = -5.0 mA		-0.4	V
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	I_C = -10 mA, I_B = -1.0 mA	-0.65	-0.85	V
		I_C = -50 mA, I_B = -5.0 mA		-0.95	V
SMALL SIGNAL CHARACTERISTICS					
f_T	Current Gain - Bandwidth Product	I_C = -10 mA, V_{CE} = -20 V, f = 100 MHz	250		MHz
C_{obo}	Output Capacitance	V_{CB} = -5.0 V, I_E = 0, f = 100 kHz		4.5	pF
C_{ibo}	Input Capacitance	V_{EB} = -0.5 V, I_C = 0, f = 100 kHz		10.0	pF
NF	Noise Figure	I_C = -100 μA, V_{CE} = -5.0 V, R_S = 1.0 kΩ, f = 10 Hz to 15.7 kHz		4.0	dB
SWITCHING CHARACTERISTICS					
t_d	Delay Time	V_{CC} = -3.0 V, V_{BE} = -0.5 V, I_C = -10 mA, I_{B1} = -1.0 mA		35	ns
t_r	Rise Time			35	ns
t_s	Storage Time	V_{CC} = -3.0 V, I_C = -10 mA, I_{B1} = I_{B2} = -1.0 mA		225	ns
t_f	Fall Time			75	ns

Note:

4. Pulse Test: Pulse Width ≤ 300μs, Duty Cycle ≤ 2.0%.

Typical Performance Characteristics

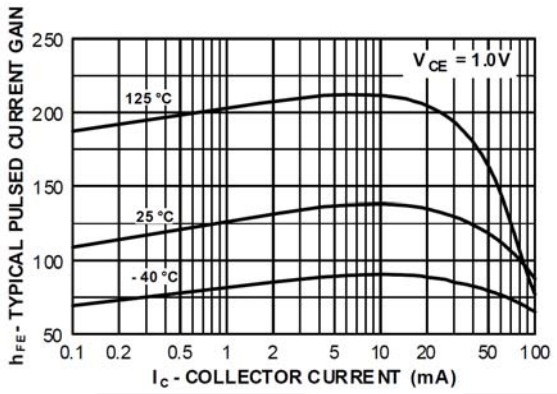


Figure 1. Typical Pulsed Current Gain vs. Collector Current

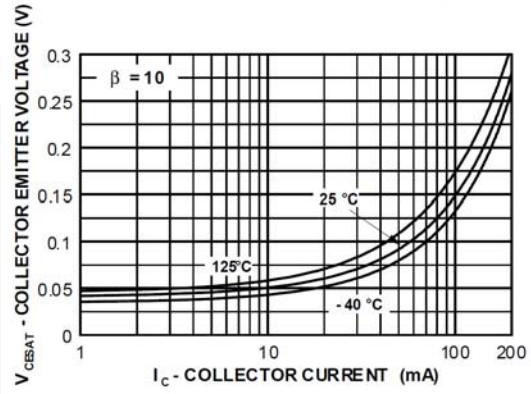


Figure 2. Collector-Emitter Saturation Voltage vs. Collector Current

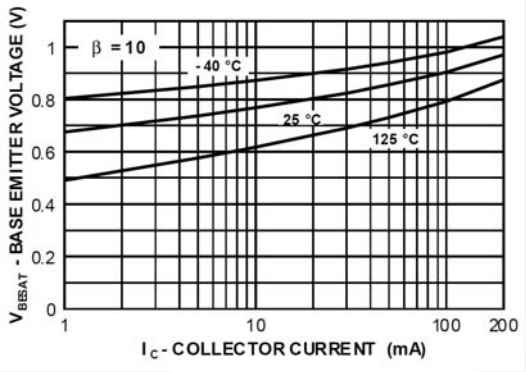


Figure 3. Base-Emitter Saturation Voltage vs. Collector Current

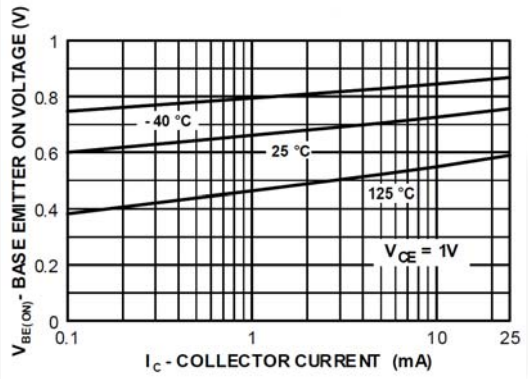


Figure 4. Base-Emitter On Voltage vs. Collector Current

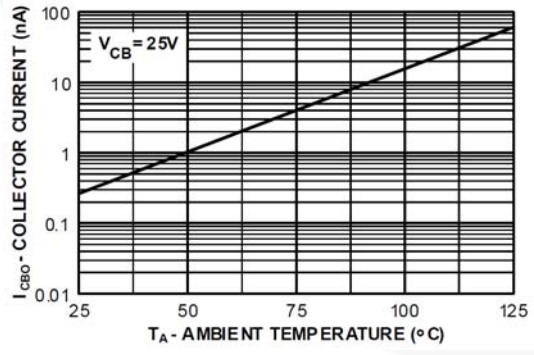


Figure 5. Collector Cut-Off Current vs. Ambient Temperature

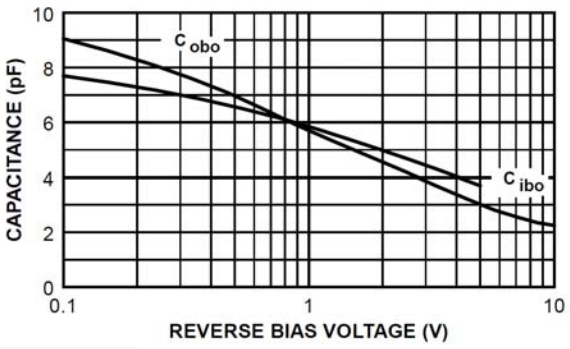


Figure 6. Common-Base Open Circuit Input and Output Capacitance vs. Reverse Bias Voltage

Typical Performance Characteristics (continued)

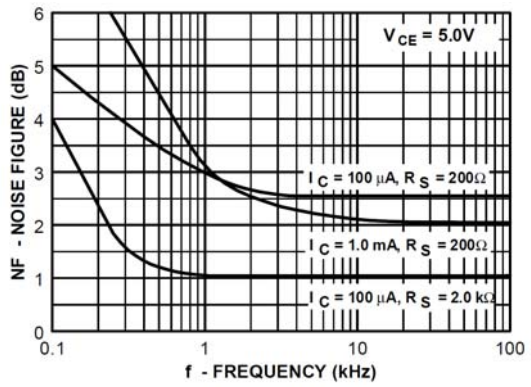


Figure 7. Noise Figure vs. Frequency

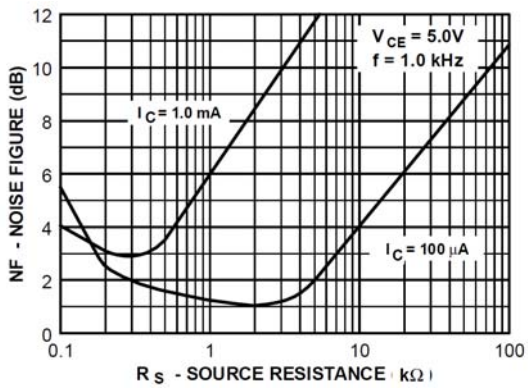


Figure 8. Noise Figure vs. Source Resistance

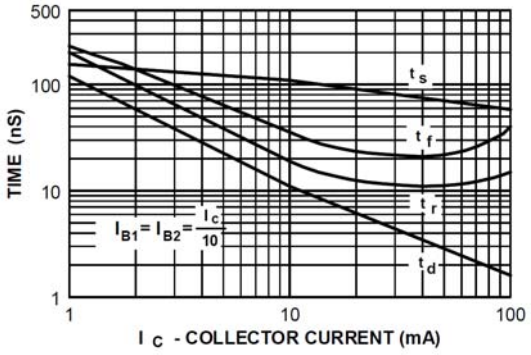


Figure 9. Switching Times vs. Collector Current

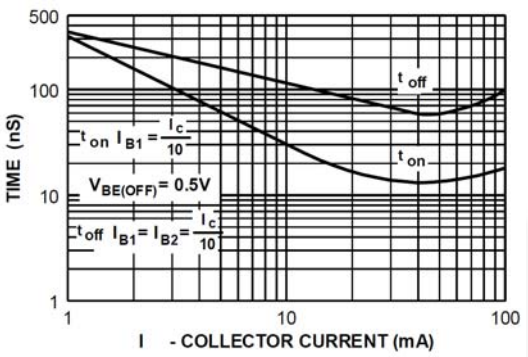


Figure 10. Turn On and Turn Off Times vs. Collector Current

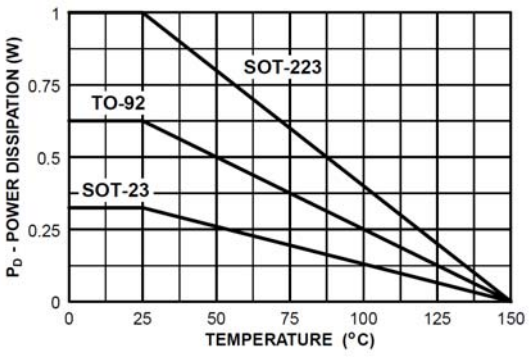


Figure 11. Power Dissipation vs. Ambient Temperature

Typical Performance Characteristics (continued)

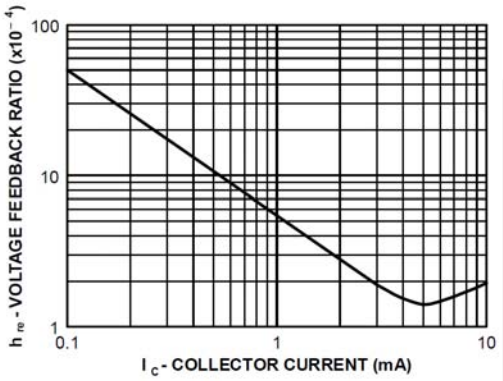


Figure 12. Voltage Feedback Ratio

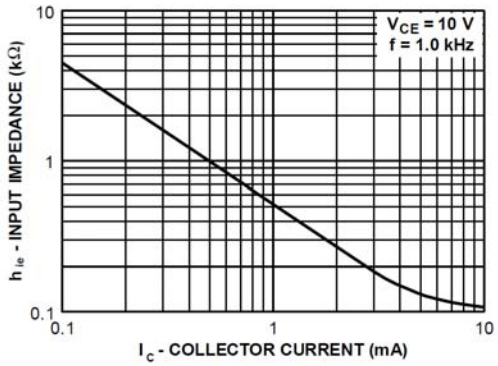


Figure 13. Input Impedance

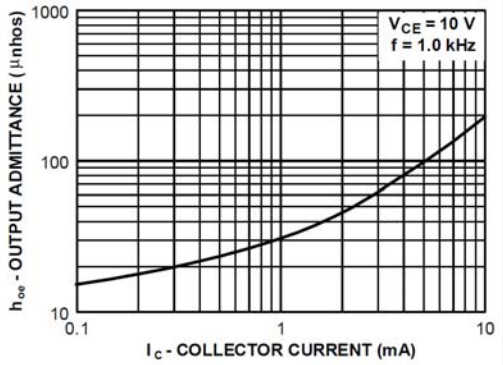


Figure 14. Output Admittance

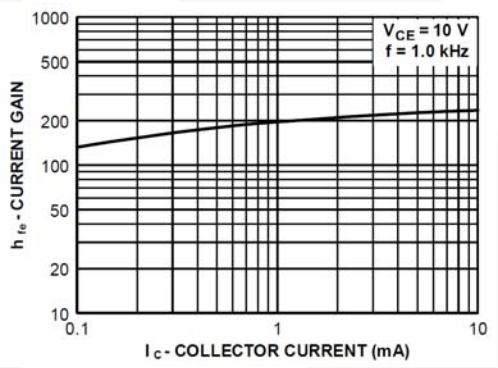
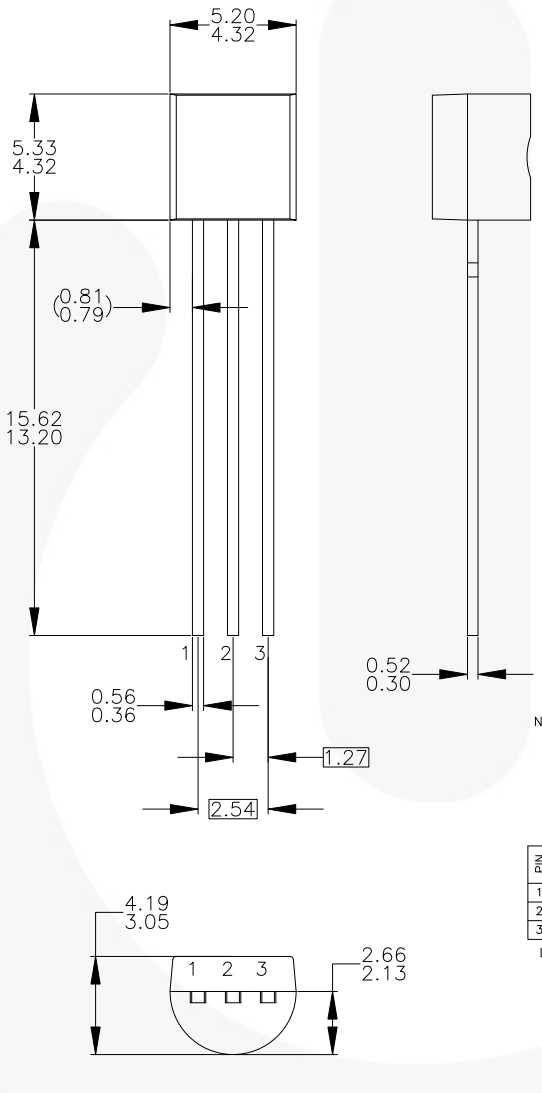


Figure 15. Current Gain

Physical Dimensions

TO-92



NOTES: UNLESS OTHERWISE SPECIFIED

A) DRAWING WITH REFERENCE TO JEDEC TO-92 RECOMMENDATIONS.
B) ALL DIMENSIONS ARE IN MILLIMETERS.
C) DRAWING CONFORMS TO ASME Y14.5M-1994.
D) TO-92 (92,94,96,97,98) PIN CONFIGURATION:

PIN	92			94			96			97			98		
	P	F	M	P	F	M	B	F	M	P	F	M	P	F	M
1	E	S	S	E	S	S	B	D	G	C	G	D	C	G	D
2	B	D	G	C	G	D	E	S	S	B	D	G	E	S	S
3	C	G	D	B	D	G	C	G	D	E	S	S	B	D	G

LEGEND:
P – BIPOLAR
F – JFET
M – DMOS
E – EMITTER
B – BASE
C – COLLECTOR
D – DRAIN
S – SOURCE
G – GATE

E) FOR PACKAGE 92, 94, 96, 97 AND 98: PIN CONFIGURATION DRAIN "D" AND SOURCE "S" ARE INTERCHANGEAGLE AT JFET "F" OPTION.
F) DRAWING FILENAME: MKT-ZA03DREV3.

Figure 16. 3-LEAD, TO92, MOLDED 0.200 IN LINE SPACING LD FORM (J61Z OPTION) (ACTIVE)

Package drawings are provided as a service to customers considering Fairchild components. Drawings may change in any manner without notice. Please note the revision and/or date on the drawing and contact a Fairchild Semiconductor representative to verify or obtain the most recent revision. Package specifications do not expand the terms of Fairchild's worldwide terms and conditions, specifically the warranty therein, which covers Fairchild products.

Always visit Fairchild Semiconductor's online packaging area for the most recent package drawings:
http://www.fairchildsemi.com/dwg/ZA/ZA03D.pdf.

For current tape and reel specifications, visit Fairchild Semiconductor's online packaging area:

http://www.fairchildsemi.com/packing_dwg/PKG-ZA03D_BK.pdf.

Physical Dimensions (continued)

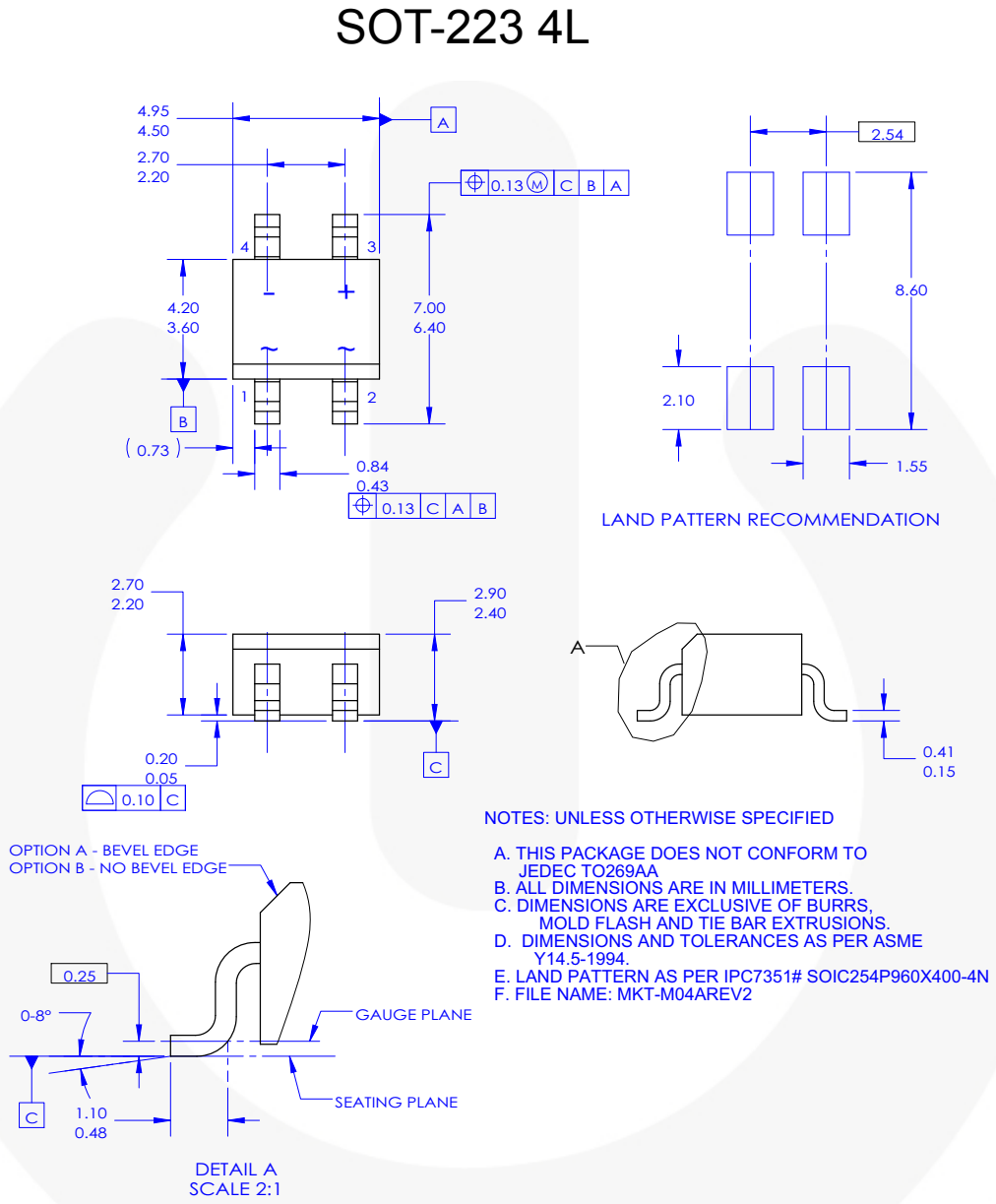


Figure 17. MOLDED PACKAGE, SOT-223, 4-LEAD (ACTIVE)

Package drawings are provided as a service to customers considering Fairchild components. Drawings may change in any manner without notice. Please note the revision and/or date on the drawing and contact a Fairchild Semiconductor representative to verify or obtain the most recent revision. Package specifications do not expand the terms of Fairchild's worldwide terms and conditions, specifically the warranty therein, which covers Fairchild products.

Always visit Fairchild Semiconductor's online packaging area for the most recent package drawings:
http://www.fairchildsemi.com/dwg/MA/MA04A.pdf.

For current tape and reel specifications, visit Fairchild Semiconductor's online packaging area:

http://www.fairchildsemi.com/packing_dwg/PKG-MA04A_BK.pdf.

Physical Dimensions (continued)

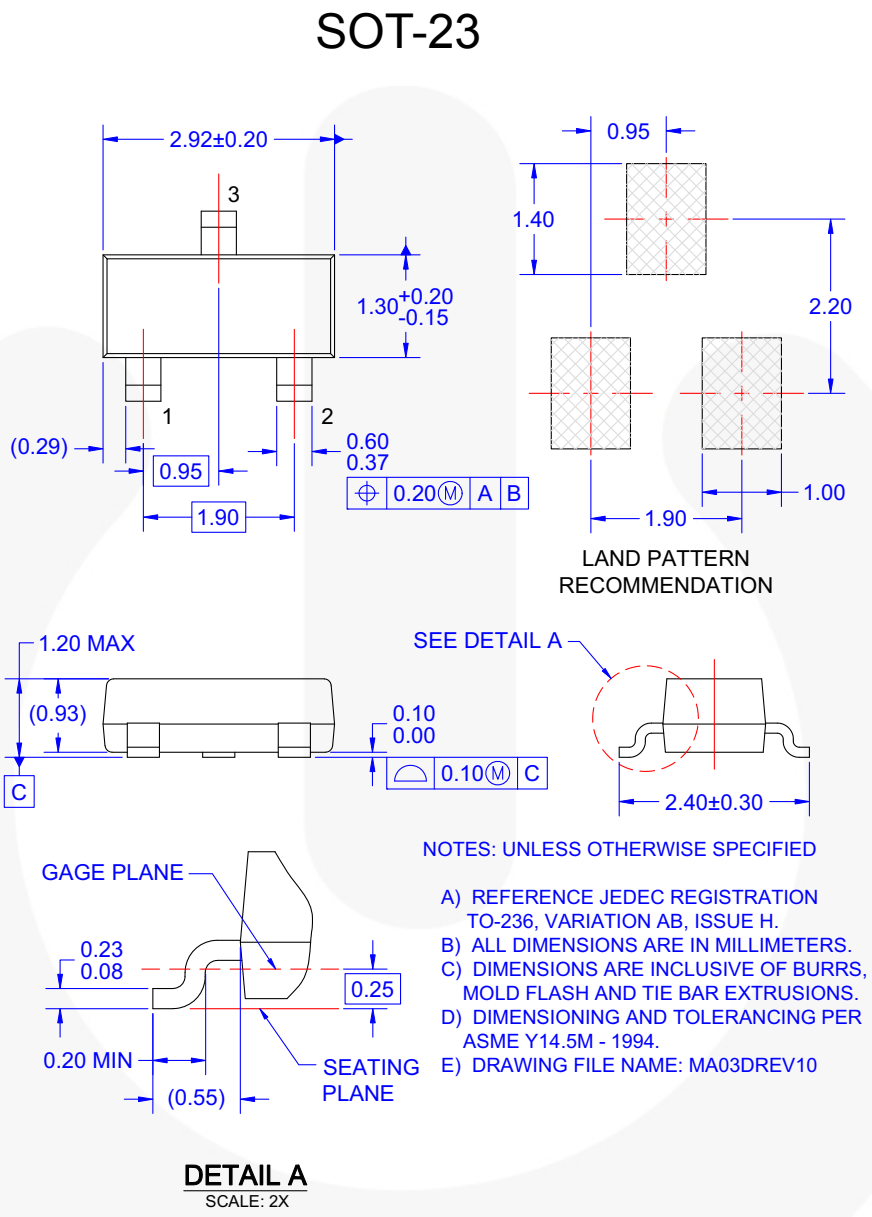


Figure 18. 3-LEAD, SOT23, JEDEC TO-236, LOW PROFILE (ACTIVE)

Package drawings are provided as a service to customers considering Fairchild components. Drawings may change in any manner without notice. Please note the revision and/or date on the drawing and contact a Fairchild Semiconductor representative to verify or obtain the most recent revision. Package specifications do not expand the terms of Fairchild's worldwide terms and conditions, specifically the warranty therein, which covers Fairchild products.

Always visit Fairchild Semiconductor's online packaging area for the most recent package drawings:
http://www.fairchildsemi.com/dwg/MA/MA03D.pdf.

For current tape and reel specifications, visit Fairchild Semiconductor's online packaging area:

http://www.fairchildsemi.com/packaging/tr/SOT23-3L_tr.pdf.

FAIRCHILD SEMICONDUCTOR®

TRADEMARKS

The following includes registered and unregistered trademarks and service marks, owned by Fairchild Semiconductor and/or its global subsidiaries, and is not intended to be an exhaustive list of all such trademarks.

2Cool™
AccuPower™
AX-CAP®*
BitSiC™
Build it Now™
CorePLUS™
CorePOWER™
CROSSVOLT™
CTL™
Current Transfer Logic™
DEUXPEED®
Dual Cool™
EcoSPARK®
EfficientMax™
ESBC™
F®
Fairchild®
Fairchild Semiconductor®
FACT Quiet Series™
FACT®
FAST®
FastvCore™
FETBench™
FPS™
F-PFS™
FRFET®
Global Power Resource SM
GreenBridge™
Green FPS™
Green FPS™ e-Series™
G*max*™
GTO™
IntelliMAX™
ISOPLANAR™
Making Small Speakers Sound Louder and Better™
MegaBuck™
MICROCOUPLER™
MicroFET™
MicroPak™
MicroPak2™
MillerDrive™
MotionMax™
mWSaver™
OptoHiT™
OPTOLOGIC®
OPTOPLANAR®
PowerTrench®
PowerXS™
Programmable Active Droop™
QFET®
QS™
Quiet Series™
RapidConfigure™
™
Saving our world, 1mW/W/kW at a time™
SignalWise™
SmartMax™
SMART START™
Solutions for Your Success™
SPM®
STEALTH™
SuperFET®
SuperSOT™-3
SuperSOT™-6
SuperSOT™-8
SupreMOS®
SyncFET™
Sync-Lock™
SYSTEM GENERAL®*
TinyBoost™
TinyBuck™
TinyCalc™
TinyLogic®
TINYOPTO™
TinyPower™
TinyPWM™
TinyWire™
TranSiC™
TriFault Detect™
TRUECURRENT®*
μSerDes™
SerDes™
UHC®
Ultra FRFET™
UniFET™
VCX™
VisualMax™
VoltagePlus™
XS™

* Trademarks of System General Corporation, used under license by Fairchild Semiconductor.

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION, OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS. THESE SPECIFICATIONS DO NOT EXPAND THE TERMS OF FAIRCHILD'S WORLDWIDE TERMS AND CONDITIONS, SPECIFICALLY THE WARRANTY THEREIN, WHICH COVERS THESE PRODUCTS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support, device, or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

ANTI-COUNTERFEITING POLICY

Fairchild Semiconductor Corporation's Anti-Counterfeiting Policy. Fairchild's Anti-Counterfeiting Policy is also stated on our external website, www.fairchildsemi.com, under Sales Support.

Counterfeiting of semiconductor parts is a growing problem in the industry. All manufacturers of semiconductor products are experiencing counterfeiting of their parts. Customers who inadvertently purchase counterfeit parts experience many problems such as loss of brand reputation, substandard performance, failed applications, and increased cost of production and manufacturing delays. Fairchild is taking strong measures to protect ourselves and our customers from the proliferation of counterfeit parts. Fairchild strongly encourages customers to purchase Fairchild parts either directly from Fairchild or from Authorized Fairchild Distributors who are listed by country on our web page cited above. Products customers buy either from Fairchild directly or from Authorized Fairchild Distributors are genuine parts, have full traceability, meet Fairchild's quality standards for handling and storage and provide access to Fairchild's full range of up-to-date technical and product information. Fairchild and our Authorized Distributors will stand behind all warranties and will appropriately address any warranty issues that may arise. Fairchild will not provide any warranty coverage or other assistance for parts bought from Unauthorized Sources. Fairchild is committed to combat this global problem and encourage our customers to do their part in stopping this practice by buying direct or from authorized distributors.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative / In Design	Datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	Datasheet contains preliminary data; supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve design.
No Identification Needed	Full Production	Datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve the design.
Obsolete	Not In Production	Datasheet contains specifications on a product that is discontinued by Fairchild Semiconductor. The datasheet is for reference information only.

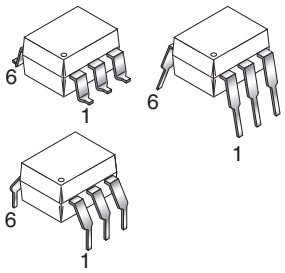
Rev. I64

FAIRCHILD
SEMICONDUCTOR®

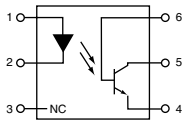
GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25 4N26 4N27 4N28 4N35 4N36
4N37 H11A1 H11A2 H11A3 H11A4 H11A5

WHITE PACKAGE (-M SUFFIX)

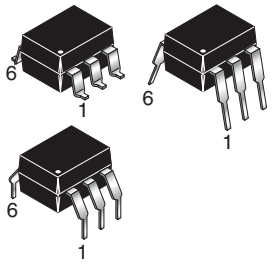


SCHEMATIC



PIN 1. ANODE
2. CATHODE
3. NO CONNECTION
4. EMITTER
5. COLLECTOR
6. BASE

BLACK PACKAGE (NO -M SUFFIX)



DESCRIPTION

The general purpose optocouplers consist of a gallium arsenide infrared emitting diode driving a silicon phototransistor in a 6-pin dual in-line package.

FEATURES

- Also available in white package by specifying -M suffix, eg. 4N25-M
- UL recognized (File # E90700)
- VDE recognized (File # 94766)
  - Add option V for white package (e.g., 4N25V-M)
  - Add option 300 for black package (e.g., 4N25.300)

APPLICATIONS

- Power supply regulators
- Digital logic inputs
- Microprocessor inputs

FAIRCHILD SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25 4N26 4N27 4N28 4N35 4N36
4N37 H11A1 H11A2 H11A3 H11A4 H11A5

ABSOLUTE MAXIMUM RATINGS (T_A = 25°C unless otherwise specified)

Parameter	Symbol	Value	Units
TOTAL DEVICE			
Storage Temperature	T_{STG}	-55 to +150	°C
Operating Temperature	T_{OPR}	-55 to +100	°C
Lead Solder Temperature	T_{SOL}	260 for 10 sec	°C
Total Device Power Dissipation @ T_A = 25°C	P_D	250	mW
Derate above 25°C		3.3 (non-M), 2.94 (-M)	
EMITTER			
DC/Average Forward Input Current	I_F	100 (non-M), 60 (-M)	mA
Reverse Input Voltage	V_R	6	V
Forward Current - Peak (300µs, 2% Duty Cycle)	I_F(pk)	3	A
LED Power Dissipation @ T_A = 25°C	P_D	150 (non-M), 120 (-M)	mW
Derate above 25°C		2.0 (non-M), 1.41 (-M)	mW/°C
DETECTOR			
Collector-Emitter Voltage	V_{CEO}	30	V
Collector-Base Voltage	V_{CBO}	70	V
Emitter-Collector Voltage	V_{ECO}	7	V
Detector Power Dissipation @ T_A = 25°C	P_D	150	mW
Derate above 25°C		2.0 (non-M), 1.76 (-M)	mW/°C

FAIRCHILD SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25 4N26 4N27 4N28 4N35 4N36
4N37 H11A1 H11A2 H11A3 H11A4 H11A5

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise specified)

INDIVIDUAL COMPONENT CHARACTERISTICS

Parameter	Test Conditions	Symbol	Min	Typ*	Max	Unit
EMITTER						
Input Forward Voltage	(I_F = 10 mA)	V_F		1.18	1.50	V
Reverse Leakage Current	(V_R = 6.0 V)	I_R		0.001	10	µA
DETECTOR						
Collector-Emitter Breakdown Voltage	(I_C = 1.0 mA, I_F = 0)	BV_{CEO}	30	100		V
Collector-Base Breakdown Voltage	(I_C = 100 µA, I_F = 0)	BV_{CBO}	70	120		V
Emitter-Collector Breakdown Voltage	(I_E = 100 µA, I_F = 0)	BV_{ECO}	7	10		V
Collector-Emitter Dark Current	(V_{CE} = 10 V, I_F = 0)	I_{CEO}		1	50	nA
Collector-Base Dark Current	(V_{CB} = 10 V)	I_{CBO}			20	nA
Capacitance	(V_{CE} = 0 V, f = 1 MHz)	C_{CE}		8		pF

ISOLATION CHARACTERISTICS

Characteristic	Test Conditions	Symbol	Min	Typ*	Max	Units
Input-Output Isolation Voltage	(Non '-M', Black Package) (f = 60 Hz, t = 1 min)	V_{ISO}	5300			Vac(rms)
	('-M', White Package) (f = 60 Hz, t = 1 sec)		7500			Vac(pk)
Isolation Resistance	(V_{I-O} = 500 VDC)	R_{ISO}	10^{11}			Ω
Isolation Capacitance	(V_{I-O} = &, f = 1 MHz)	C_{ISO}		0.5		pF
	('-M' White Package)			0.2	2	pF

Note
* Typical values at T_A = 25°C

FAIRCHILD SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25 4N26 4N27 4N28 4N35 4N36
4N37 H11A1 H11A2 H11A3 H11A4 H11A5

TRANSFER CHARACTERISTICS (T_A = 25°C Unless otherwise specified.)

DC Characteristic	Test Conditions	Symbol	Device	Min	Typ*	Max	Unit
Current Transfer Ratio, Collector to Emitter	(I_F = 10 mA, V_{CE} = 10 V)	CTR	4N35 4N36 4N37	100			%
			H11A1	50			
			H11A5	30			
			4N25 4N26 H11A2 H11A3	20			
			4N27 4N28 H11A4	10			
	(I_F = 10 mA, V_{CE} = 10 V, T_A = -55°C)		4N35 4N36 4N37	40			
	(I_F = 10 mA, V_{CE} = 10 V, T_A = +100°C)		4N35 4N36 4N37	40			
Collector-Emitter Saturation Voltage	(I_C = 2 mA, I_F = 50 mA)	$V_{CE\ (SAT)}$	4N25 4N26 4N27 4N28			0.5	V
	(I_C = 0.5 mA, I_F = 10 mA)		4N35 4N36 4N37			0.3	
			H11A1 H11A2 H11A3 H11A4 H11A5			0.4	
AC Characteristic Non-Saturated Turn-on Time	(I_F = 10 mA, V_{CC} = 10 V, R_L = 100Ω) (Fig.20)	T_{ON}	4N25 4N26 4N27 4N28 H11A1 H11A2 H11A3 H11A4 H11A5		2		µs
Non Saturated Turn-on Time	(I_C = 2 mA, V_{CC} = 10 V, R_L = 100Ω) (Fig.20)	T_{ON}	4N35 4N36 4N37		2	10	µs

FAIRCHILD SEMICONDUCTOR®

TRANSFER CHARACTERISTICS (T_A = 25°C Unless otherwise specified.) (Continued)

AC Characteristic	Test Conditions	Symbol	Device	Min	Typ*	Max	Unit
Turn-off Time	(I_F = 10 mA, V_{CC} = 10 V, R_L = 100Ω) (Fig.20)	T_{OFF}	4N25 4N26 4N27 4N28 H11A1 H11A2 H11A3 H11A4 H11A5		2		µs
	(I_C = 2 mA, V_{CC} = 10 V, R_L = 100Ω) (Fig.20)		4N35 4N36 4N37		2	10	

* Typical values at T_A = 25°C

FAIRCHILD SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25	4N26	4N27	4N28	4N35	4N36
4N37	H11A1	H11A2	H11A3	H11A4	H11A5

TYPICAL PERFORMANCE CURVES

Fig. 1 LED Forward Voltage vs. Forward Current (Black Package)

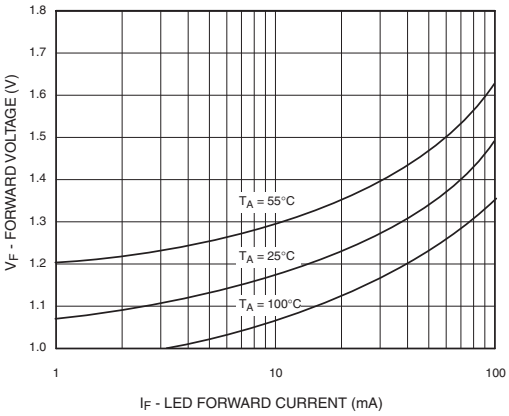


Fig. 2 LED Forward Voltage vs. Forward Current (White Package)

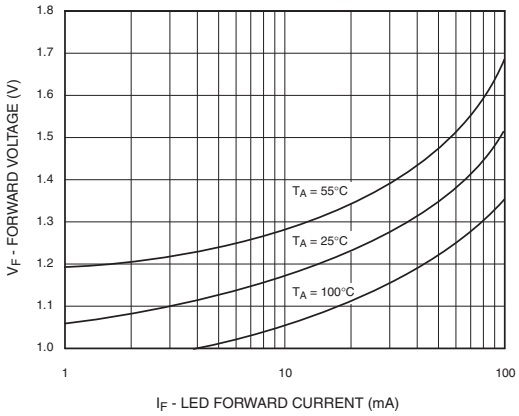


Fig.3 Normalized CTR vs. Forward Current (Black Package)

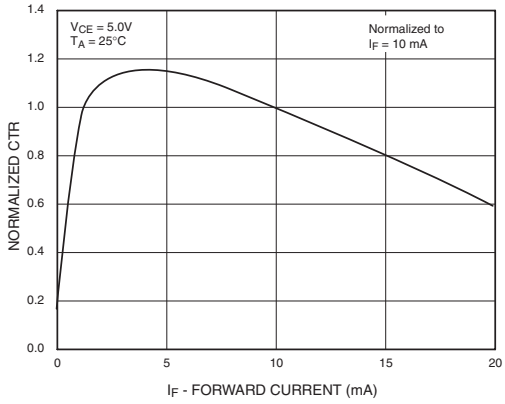


Fig.4 Normalized CTR vs. Forward Current (White Package)

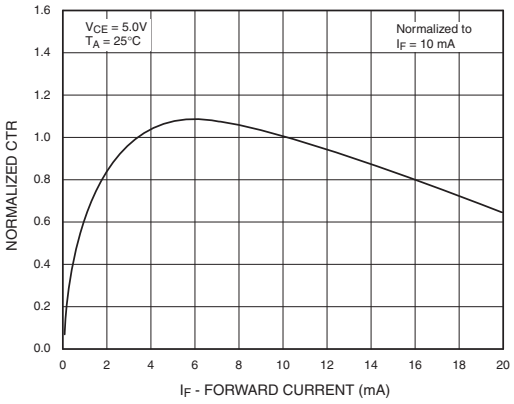


Fig. 5 Normalized CTR vs. Ambient Temperature (Black Package)

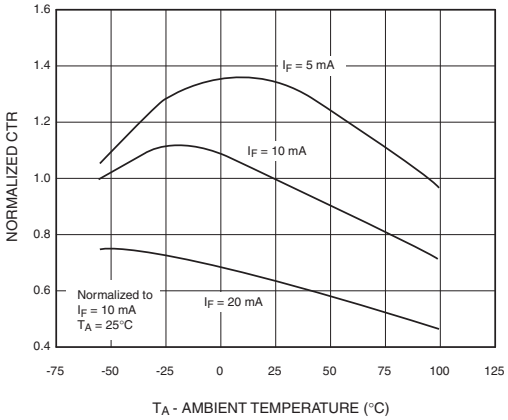
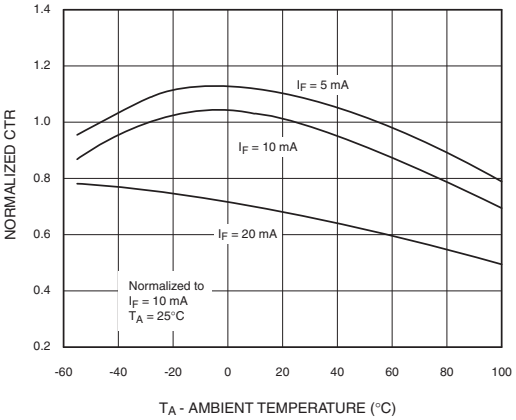


Fig. 6 Normalized CTR vs. Ambient Temperature (White Package)



FAIRCHILD SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25	4N26	4N27	4N28	4N35	4N36
4N37	H11A1	H11A2	H11A3	H11A4	H11A5

Fig. 7 CTR vs. RBE (Unsaturated) (Black Package)

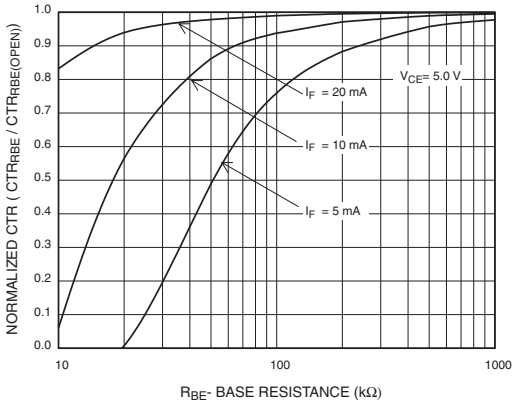


Fig. 8 CTR vs. RBE (Unsaturated) (White Package)

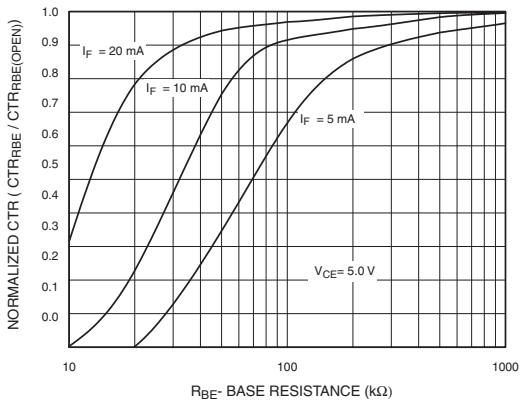


Fig. 9 CTR vs. RBE (Saturated) (Black Package)

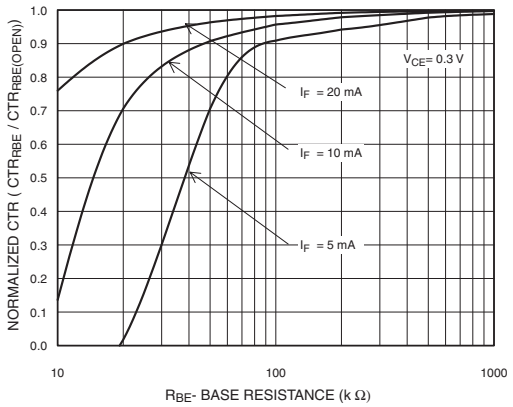


Fig. 10 CTR vs. RBE (Saturated) (White Package)

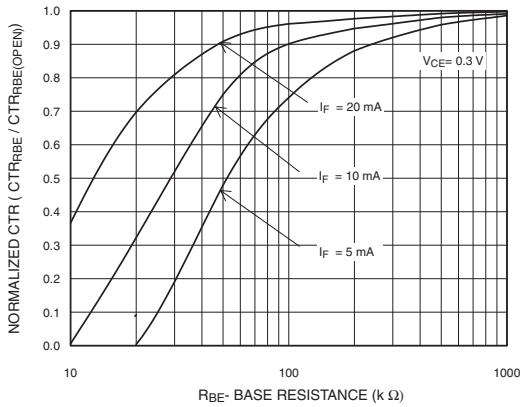


Fig. 11 Collector-Emitter Saturation Voltage vs Collector Current (Black Package)

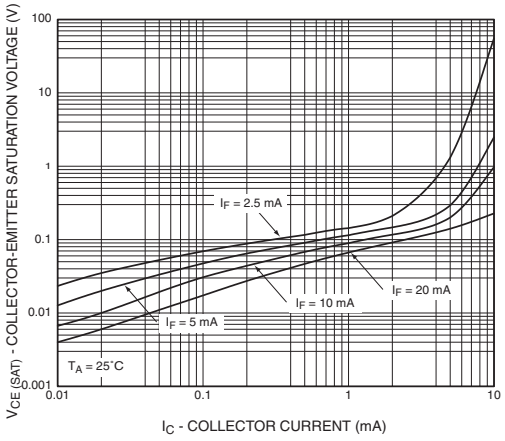


Fig. 12 Collector-Emitter Saturation Voltage vs Collector Current (White Package)

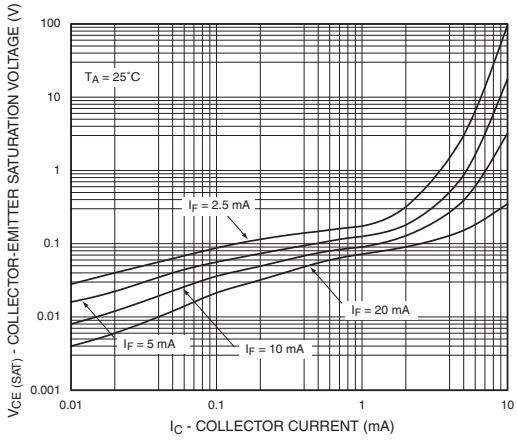


Fig. 13 Switching Speed vs. Load Resistor (Black Package)

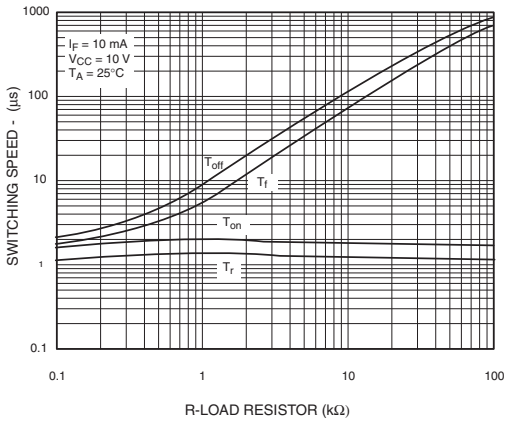


Fig. 14 Switching Speed vs. Load Resistor (White Package)

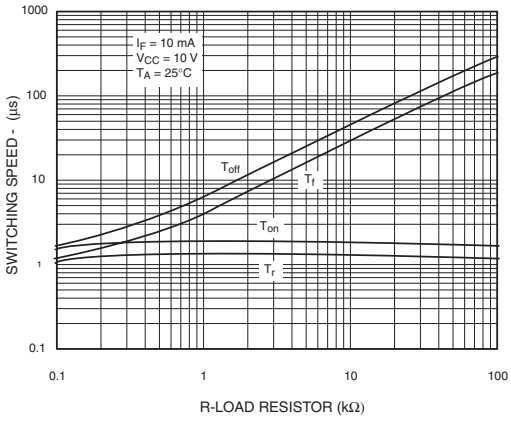


Fig. 15 Normalized t_{on} vs. R_{BE} (Black Package)

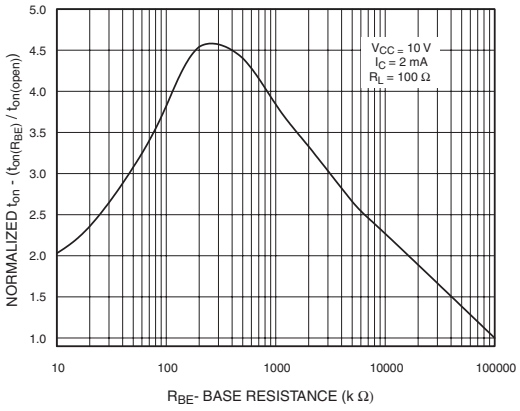


Fig. 16 Normalized t_{on} vs. R_{BE} (White Package)

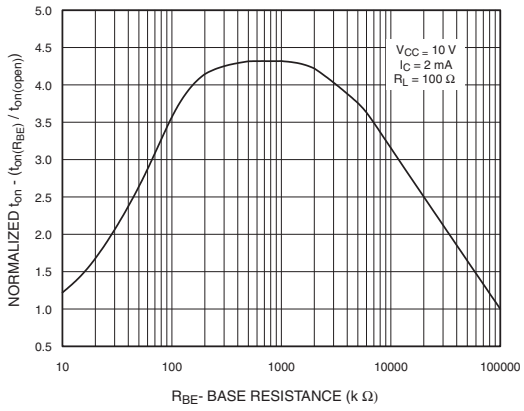


Fig. 17 Normalized t_{off} vs. R_{BE} (Black Package)

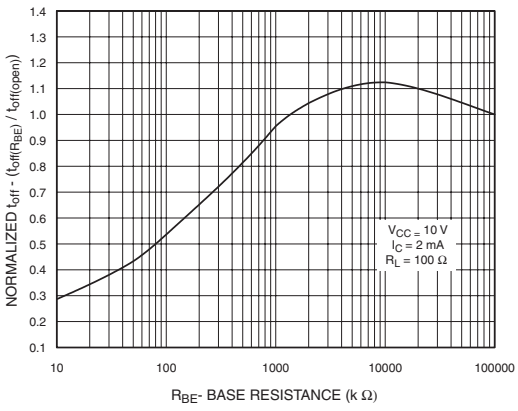


Fig. 18 Normalized t_{off} vs. R_{BE} (White Package)

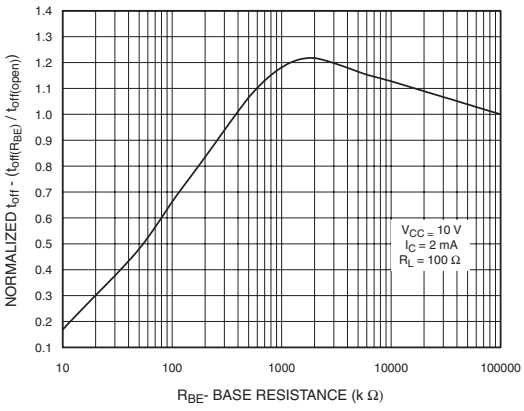
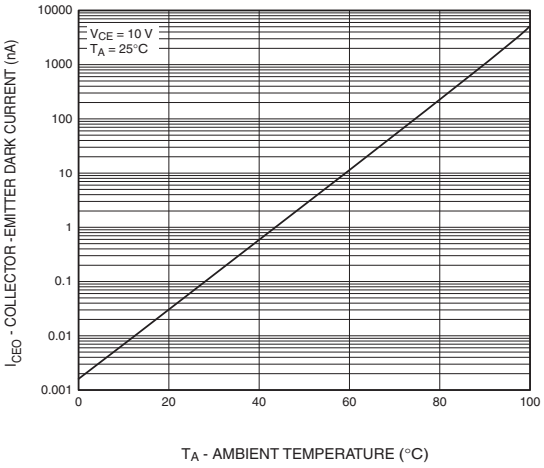


Fig. 19 Dark Current vs. Ambient Temperature



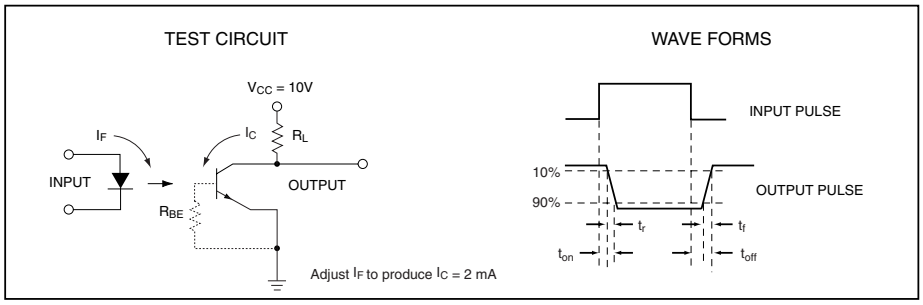


Figure 20. Switching Time Test Circuit and Waveforms

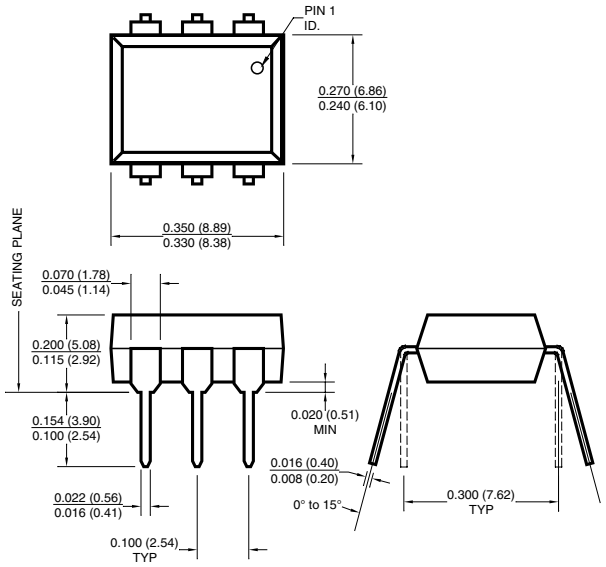
FAIRCHILD SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

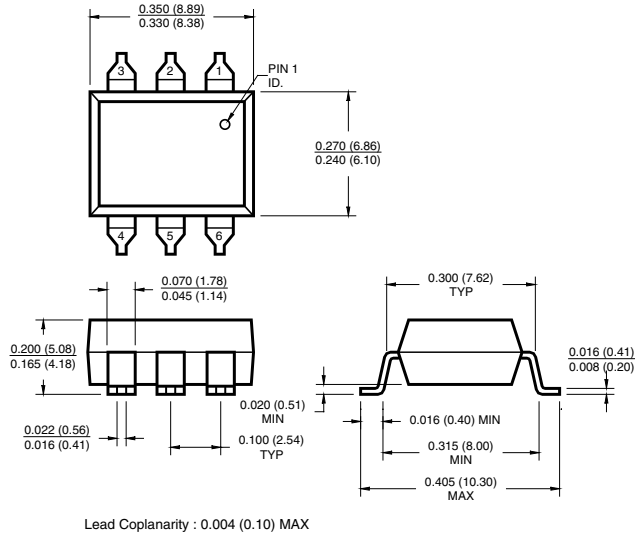
4N25	4N26	4N27	4N28	4N35	4N36
4N37	H11A1	H11A2	H11A3	H11A4	H11A5

Black Package (No -M Suffix)

Package Dimensions (Through Hole)

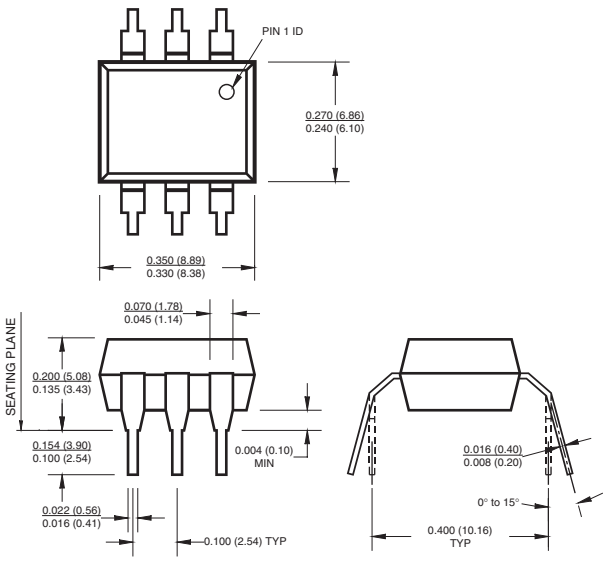


Package Dimensions (Surface Mount)

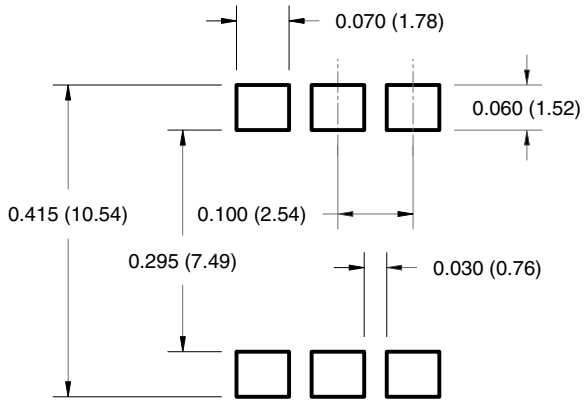


Lead Coplanarity : 0.004 (0.10) MAX

Package Dimensions (0.4" Lead Spacing)



Recommended Pad Layout for Surface Mount Leadform



NOTE
All dimensions are in inches (millimeters)

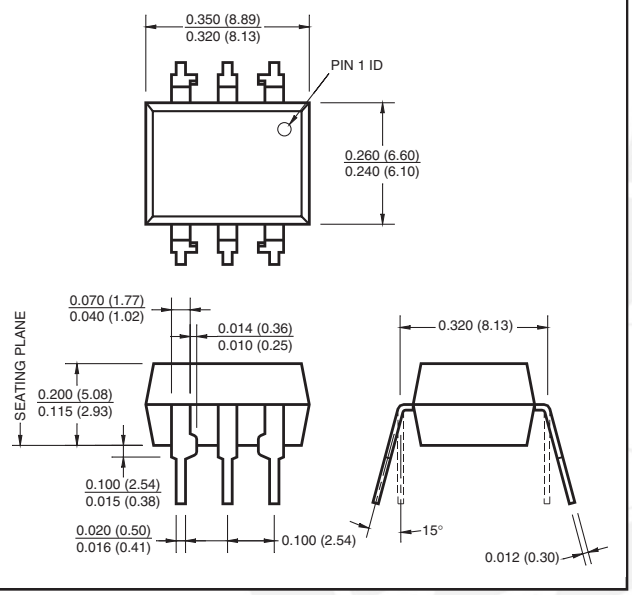
FAIRCHILD
SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

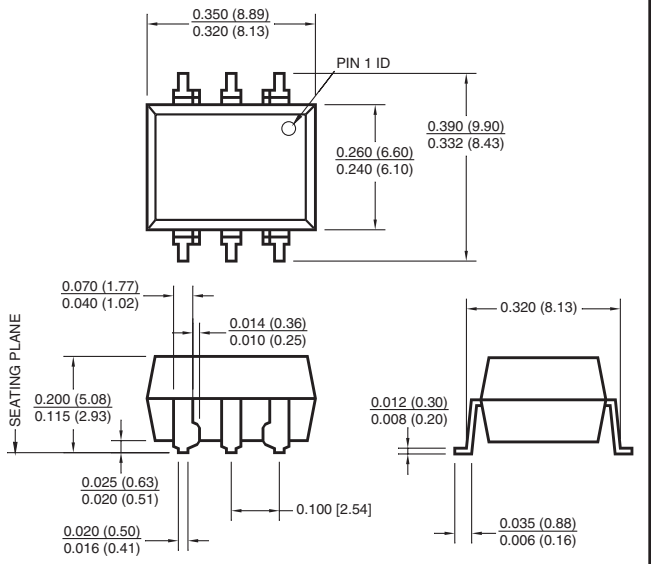
4N25	4N26	4N27	4N28	4N35	4N36
4N37	H11A1	H11A2	H11A3	H11A4	H11A5

White Package (-M Suffix)

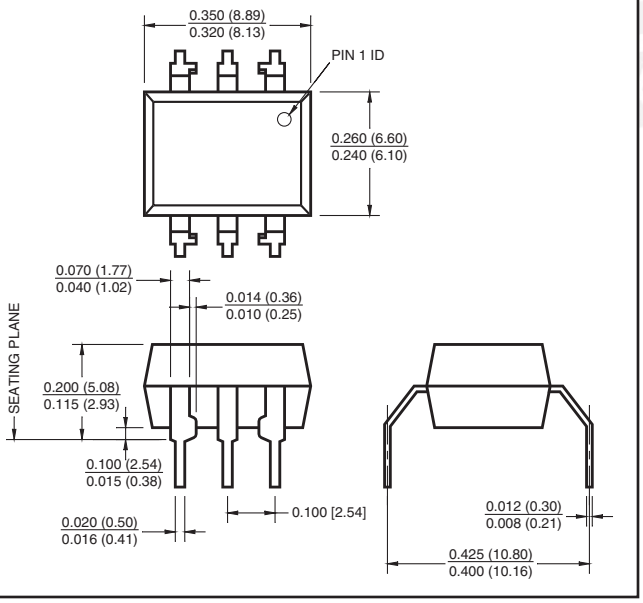
Package Dimensions (Through Hole)



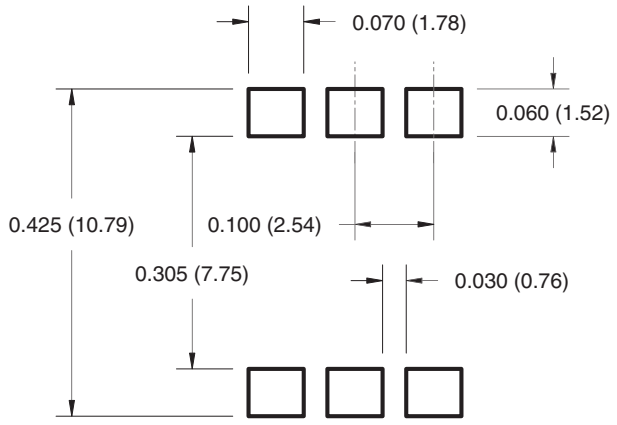
Package Dimensions (Surface Mount)



Package Dimensions (0.4" Lead Spacing)



Recommended Pad Layout for Surface Mount Leadform



NOTE
All dimensions are in inches (millimeters)

FAIRCHILD SEMICONDUCTOR®

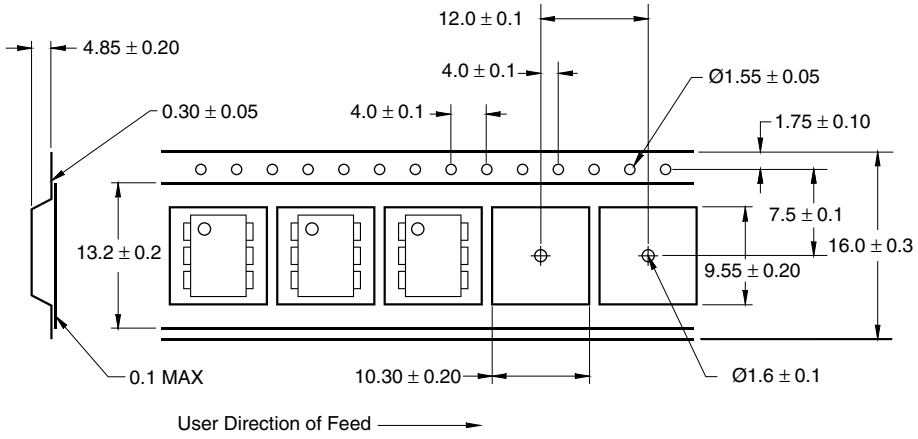
GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25 4N26 4N27 4N28 4N35 4N36
4N37 H11A1 H11A2 H11A3 H11A4 H11A5

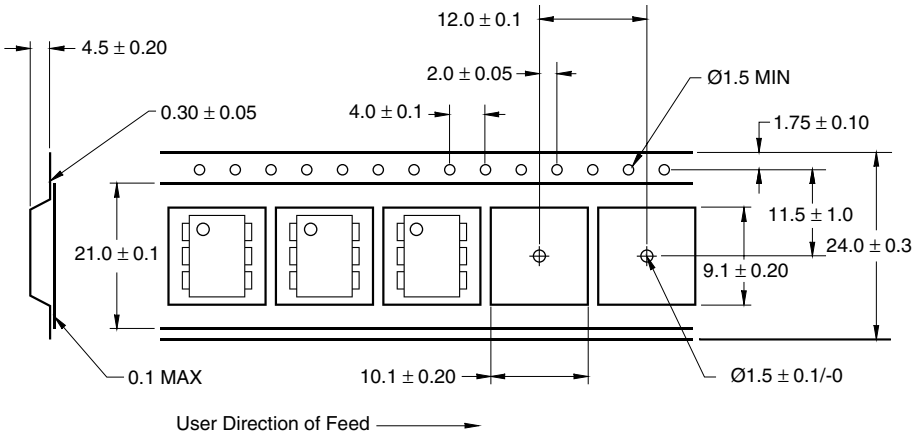
ORDERING INFORMATION

Order Entry Identifier		
Black Package (No Suffix)	**White Package (-M Suffix)**	**Option**
.S	S	Surface Mount Lead Bend
.SD	SR2	Surface Mount; Tape and reel
.W	T	0.4" Lead Spacing
.300	V	VDE 0884
.300W	TV	VDE 0884, 0.4" Lead Spacing
.3S	SV	VDE 0884, Surface Mount
.3SD	SR2V	VDE 0884, Surface Mount, Tape & Reel

QT Carrier Tape Specifications (Black Package, No Suffix)



QT Carrier Tape Specifications (White Package, -M Suffix)



FAIRCHILD SEMICONDUCTOR®

GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25	4N26	4N27	4N28	4N35	4N36
4N37	H11A1	H11A2	H11A3	H11A4	H11A5

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com
Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

4x4 Matrix Membrane Keypad (#27899)

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.

Features

- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

Key Specifications

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions:
  Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
  Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

Application Ideas

- Security systems
- Menu selection
- Data entry for embedded systems

How it Works

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in Figure 1.

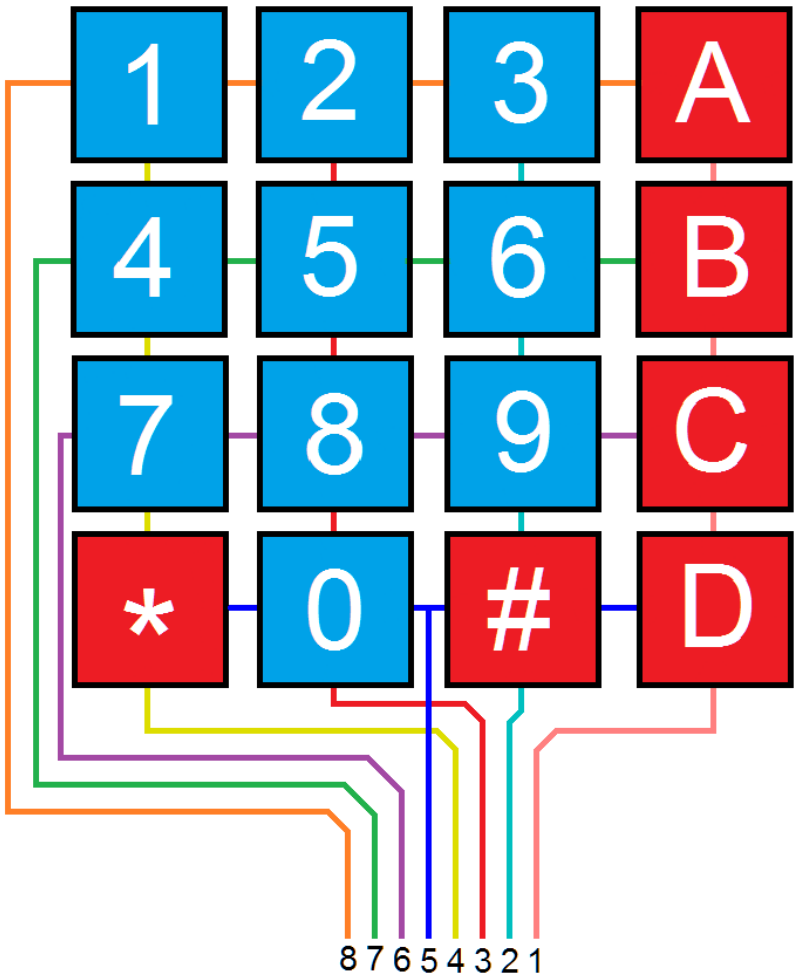


Figure 1: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed.

For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1 high. This means that a contact has been made between column 4 and row 1, so button 'A' has been pressed.

Connection Diagrams

Figure 2

For use with the BASIC Stamp example program listed below.

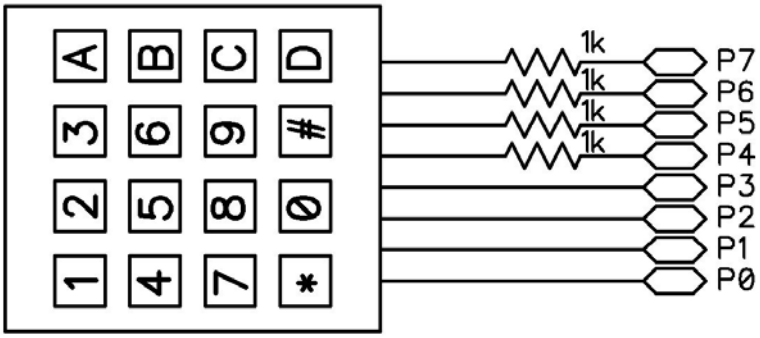
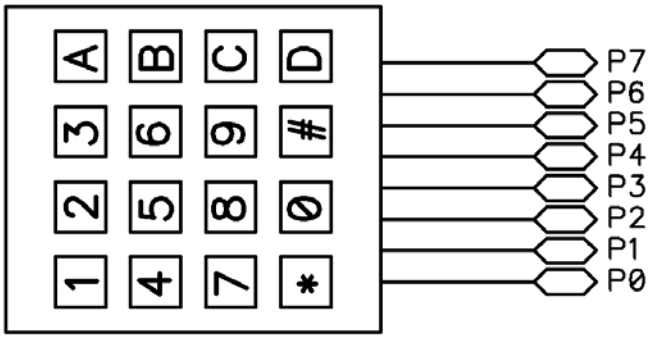


Figure 3

For use with the Propeller P8X32A example program listed below.



BASIC Stamp® Example Code

The example code below displays the button states of the 4x4 Matrix Membrane Keypad. It uses the Debug Terminal, which is built into the BASIC Stamp Editor software. The software is a free download from www.parallax.com/basicstampsoftware.

```
' 4x4MatrixKeypad_Demo.bs2
' Display buttons pressed on the 4x4 Matrix Membrane Keypad
' Author: Parallax HK Engineering

' {$STAMP BS2}
' {$PBASIC 2.5}

row         VAR  Nib                    ' Variable space for row counting
column      VAR  Nib                    ' Variable space for column counting
keypad      VAR  Word                   ' Variable space to store keypad output
keypadOld   VAR  Word                   ' Variable space to store old keypad output
temp        VAR  Nib                    ' Variable space for polling column states

DEBUG CLS                               ' Clear Debug Terminal
GOSUB Update                            ' Display keypad graphic

DO
  GOSUB ReadKeypad                      ' Read keypad button states
  DEBUG HOME, BIN16 keypad, CR, CR,     ' Display 16-bit keypad value
              BIN4 keypad >> 12,CR,     ' Display 1st row 4-bit keypad value
              BIN4 keypad >> 8, CR,     ' Display 2nd row 4-bit keypad value
              BIN4 keypad >> 4, CR,     ' Display 3rd row 4-bit keypad value
              BIN4 keypad               ' Display 4th row 4-bit keypad value
```

```
  IF keypad <> keypadOld THEN               ' If different button is pressed,
    GOSUB Update                            ' update the keypad graphic to clear
  ENDIF                                     ' old display

  IF keypad THEN                            ' Display button pressed in graphic
    GOSUB display
  ENDIF

  keypadOld = keypad                        ' Store keypad value in variable keypadOld
LOOP

' -----[ Subroutine - ReadKeypad ]-------------------------------------------------
' Read keypad button states
ReadKeypad:
  keypad = 0
  OUTL   = %00000000                        ' Initialize IO
  DIRL   = %00000000

  FOR row = 0 TO 3
    DIRB = %1111                            ' Set columns (P7-P4) as outputs
    OUTB = %0000                            ' Pull columns low (act as pull down)
    OUTA = 1 << row                         ' Set rows high one by one
    DIRA = 1 << row

    temp = 0                                ' Reset temp variable to 0
    FOR column = 0 TO 3
      INPUT (column + 4)                    ' Set columns as inputs
      temp = temp | (INB & (1 << column))   ' Poll column state and store in temp
    NEXT

    keypad = keypad << 4 | (Temp REV 4)     ' Store keypad value
  NEXT
RETURN

' -----[ Subroutine - Update ]-----------------------------------------------------
' Graphical depiction of keypad
Update:
  DEBUG CRSRXY,0,7,
    "+---+---+---+---+",CR,
    "|   |   |   |   |",CR,
    "+---+---+---+---+",CR,
    "|   |   |   |   |",CR,
    "+---+---+---+---+",CR,
    "|   |   |   |   |",CR,
    "+---+---+---+---+",CR,
    "|   |   |   |   |",CR,
    "+---+---+---+---+"
RETURN

' -----[ Subroutine - Display ]---------------------------------------------------
' Display button pressed in keypad graphic
Display:
  IF KeyPad.BIT15 THEN  DEBUG CRSRXY, 02,08,"1"
  IF Keypad.BIT14 THEN  DEBUG CRSRXY, 06,08,"2"
  IF KeyPad.BIT13 THEN  DEBUG CRSRXY, 10,08,"3"
  IF Keypad.BIT12 THEN  DEBUG CRSRXY, 14,08,"A"
  IF KeyPad.BIT11 THEN  DEBUG CRSRXY, 02,10,"4"
  IF Keypad.BIT10 THEN  DEBUG CRSRXY, 06,10,"5"
  IF KeyPad.BIT9  THEN  DEBUG CRSRXY, 10,10,"6"
  IF Keypad.BIT8  THEN  DEBUG CRSRXY, 14,10,"B"
  IF KeyPad.BIT7  THEN  DEBUG CRSRXY, 02,12,"7"
  IF Keypad.BIT6  THEN  DEBUG CRSRXY, 06,12,"8"
  IF KeyPad.BIT5  THEN  DEBUG CRSRXY, 10,12,"9"
```

```
  IF Keypad.BIT4  THEN  DEBUG CRSRXY, 14,12,"C"
  IF KeyPad.BIT3  THEN  DEBUG CRSRXY, 02,14,"*"
  IF Keypad.BIT2  THEN  DEBUG CRSRXY, 06,14,"0"
  IF KeyPad.BIT1  THEN  DEBUG CRSRXY, 10,14,"#"
  IF Keypad.BIT0  THEN  DEBUG CRSRXY, 14,14,"D"
RETURN
```

Propeller™ P8X32A Example Code

The example code below displays the button states of the 4x4 Matrix Membrane Keypad, and is a modified version of the 4x4 Keypad Reader DEMO object by Beau Schwabe.

Note: This application uses the 4x4 Keypad Reader.spin object. It also uses the Parallax Serial Terminal to display the device output. Both objects and the Parallax Serial Terminal itself are included with the with the Propeller Tool v1.2.7 or higher, which is available from the Downloads link at www.parallax.com/Propeller.

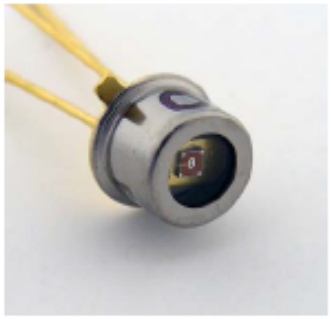
```
{{ 4x4 Keypad Reader PST.spin
Returns the entire 4x4 keypad matrix into a single WORD variable indicating which buttons are
pressed. }}

CON

  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

OBJ
  text : "Parallax Serial Terminal"
  KP   : "4x4 Keypad Reader"

VAR
  word  keypad

PUB start
  'start term
  text.start(115200)
  text.str(string(13,"4x4 Keypad Demo..."))
  text.position(1, 7)
  text.str(string(13,"RAW keypad value 'word'"))

  text.position(1, 13)
  text.str(string(13,"Note: Try pressing multiple keys"))

  repeat
    keypad := KP.ReadKeyPad     '<-- One line command to read the 4x4 keypad
    text.position(5, 2)
    text.bin(keypad>>0, 4)      'Display 1st ROW
    text.position(5,3)
    text.bin(keypad>>4, 4)      'Display 2nd ROW
    text.position(5, 4)
    text.bin(keypad>>8, 4)      'Display 3rd ROW
    text.position(5, 5)
    text.bin(keypad>>12, 4)     'Display 4th ROW
    text.position(5, 9)
    text.bin(keypad, 16)        'Display RAW keypad value
```

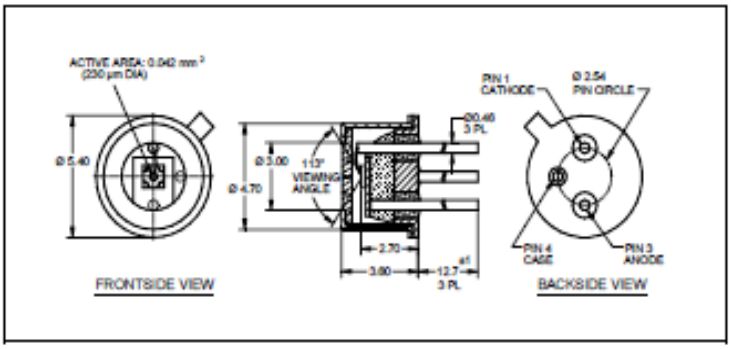
Revision History

v1.0: original document
v1.1: Updated Figure 1 on page 2
v1.2: Updated Figure 1 on page 2 (again); updated BS2 comments

PACIFIC SILICON SENSOR Incorporated

Pacific Silicon Sensor Series 8 Data Sheet
Part Description AD230-8-TO52-S1
Order # 06-015





FEATURES

- ∅ 230 µm active area
- High gain at low bias voltage
- Fast rise time
- Low capacitance

DESCRIPTION

0.042 mm² High Speed, High Gain Avalanche Photodiode with N on P construction. Hermetically packaged in a TO-52-S1 with a clear borosilicate glass window cap.

APPLICATIONS

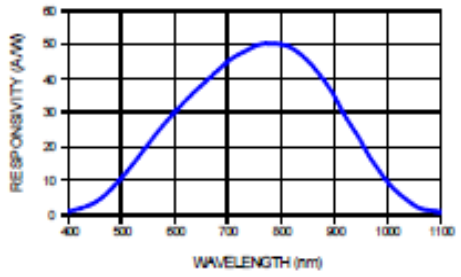
- High speed optical communications
- Laser range finder
- Medical equipment
- High speed photometry

RoHS COMPLIANT

ABSOLUTE MAXIMUM RATING

SYMBOL	PARAMETER	MIN	MAX	UNITS
T_{STG}	Storage Temp	-55	+125	°C
T_{OP}	Operating Temp	-40	+100	°C
$T_{SOLDERING}$	Soldering Temp 10 seconds		+260	°C
	Electrical Power Dissipation @ 22°C	-	100	mW
	Optical Peak Value, once for 1 second	-	200	mW
I_{PH} (DC)	Continuous Optical Operation	-	250	µA
I_{PH} (AC)	Pulsed Signal Input 50 µs "on" / 1 ms "off"	-	1	mA

SPECTRAL RESPONSE at M = 100



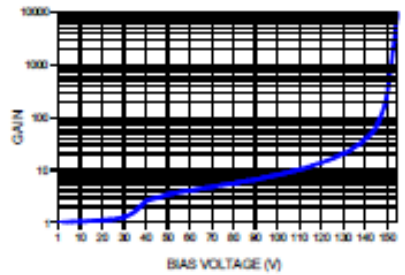
ELECTRO-OPTICAL CHARACTERISTICS @ 22 °C

SYMBOL	CHARACTERISTIC	TEST CONDITIONS	MIN	TYP	MAX	UNITS
I_D	Dark Current	M = 100*	—	0.3	1.5	nA
C	Capacitance	M = 100*	—	1.2	—	pF
V_{BR}	Breakdown Voltage	I_D = 2 µA	80	200	—	V
	Temperature Coefficient of V_{BR}		0.35	0.45	0.55	V/K
	Responsivity	M = 100; = 0 V; λ = 800 nm	45	50	—	A/W
Δf_{3dB}	Bandwidth	-3dB	—	2	—	GHz
t_r	Rise Time		—	180	—	ps
	Optimum Gain		50	60	—	
	"Excess Noise" factor	M = 100	—	2.2	—	
	"Excess Noise" Index	M = 100	—	0.2	—	
	Noise Current	M = 100	—	0.5	—	$pA/Hz^{1/2}$
	Max Gain		200	—	—	
NEP	Noise Equivalent Power	M = 100; λ = 800 nm	—	1.0×10^{-14}	—	$W/Hz^{1/2}$

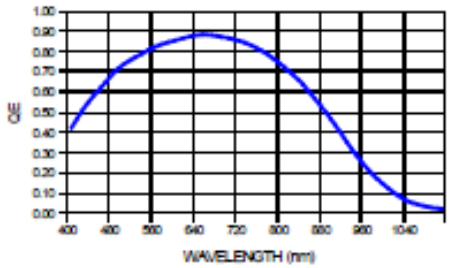
* Measurement conditions: Setup of photo current 1.0 nA at M = 1 and irradiated by a 680 nm, 60 nm bandwidth LED. Increase the photo current up to 1 µA, (M = 100) by internal multiplication due to an increasing bias voltage.

Disclaimer: Due to our policy of continued development, specifications are subject to change without notice.

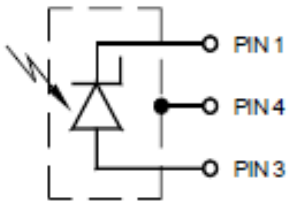
TYPICAL GAIN vs BIAS VOLTAGE



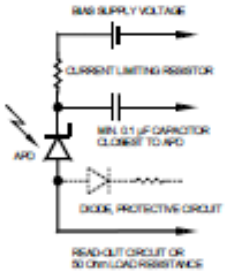
QUANTUM EFFICIENCY for M = 1



DEVICE SCHEMATIC



SUGGESTED CIRCUIT SCHEMATIC



APPLICATION NOTES

- Current should be limited by a protecting resistor or current limiting IC inside the power supply.
- Use of low noise read-out IC.
- For high gain applications (M>50) bias voltage should be temperature compensated.
- For low light level applications, blocking of ambient light should be used.

HANDLING PRECAUTIONS:

- Soldering temperature - 260°C for 10 seconds max. The device must be protected against solder flux vapor.
- Minimum pin length - 2 mm
- ESD protection - Standard precautionary measures are sufficient.
- Storage - Store devices in conductive foam.
- Avoid skin contact with window.
- Clean window with Ethyl alcohol if necessary.
- Do not scratch or abrade window.

USA:
Pacific Silicon Sensor, Inc.
5700 Corsa Avenue, #105
Westlake Village, CA 91362 USA
Phone (818) 706-3400
Fax (818) 889-7053
Email: sales@pacific-sensor.com
www.pacific-sensor.com

International sales:
Silicon Sensor International AG
Peter-Behrens-Str. 15
D-12459 Berlin, Germany
Phone +49 (0)30-63 99 23 10
Fax +49 (0)30-63 99 23 33
Email: sales@silicon-sensor.de
www.silicon-sensor.de

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 16 Kbytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1 Kbyte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels in TQFP Package Only
    2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7V - 5.5V for ATmega16L
  - 4.5V - 5.5V for ATmega16
- Speed Grades
  - 0 - 8 MHz for ATmega16L
  - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA

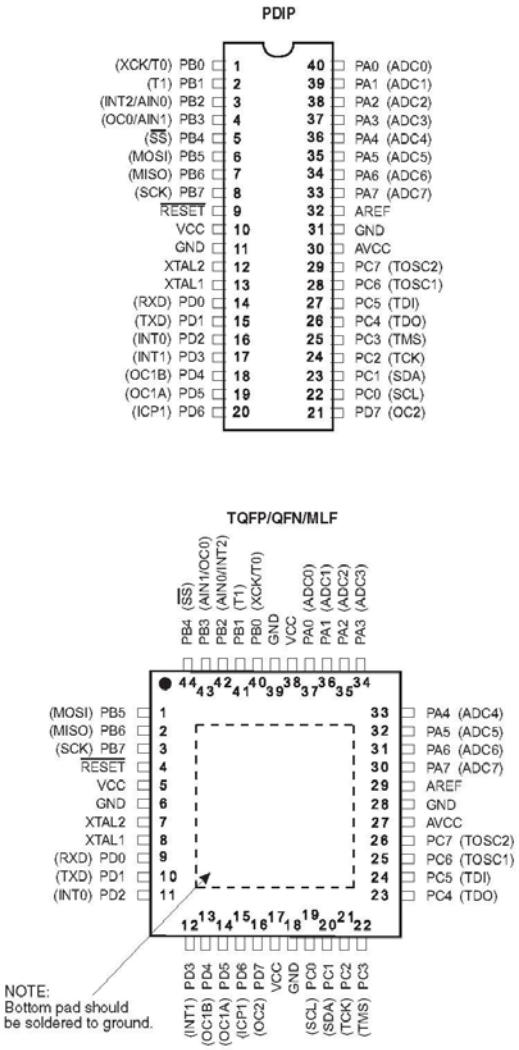
8-bit AVR® Microcontroller with 16K Bytes In-System Programmable Flash

ATmega16
ATmega16L

Rev. 2466T–AVR–07/10

Pin Configurations

Figure 1. Pinout ATmega16



Disclaimer

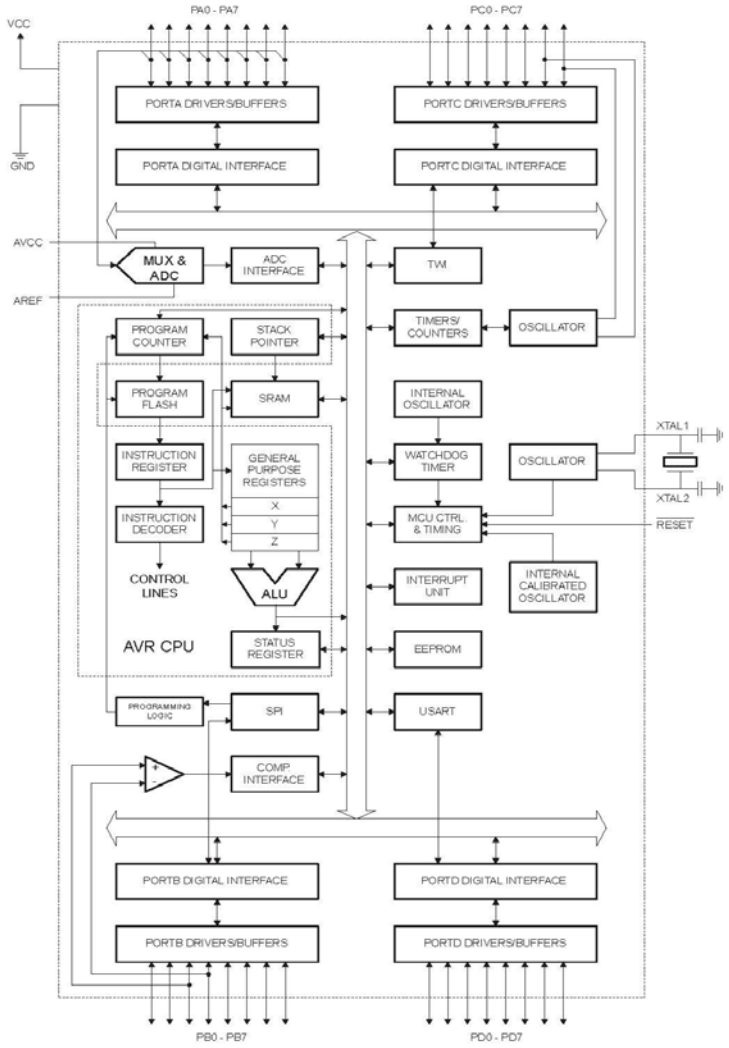
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16 provides the following features: 16 Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega16 as listed on page 58.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 61.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega16 as listed on page 63.

$\overline{\text{RESET}}$

Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38. Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

AREF

AREF is the analog reference pin for the A/D Converter.

Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

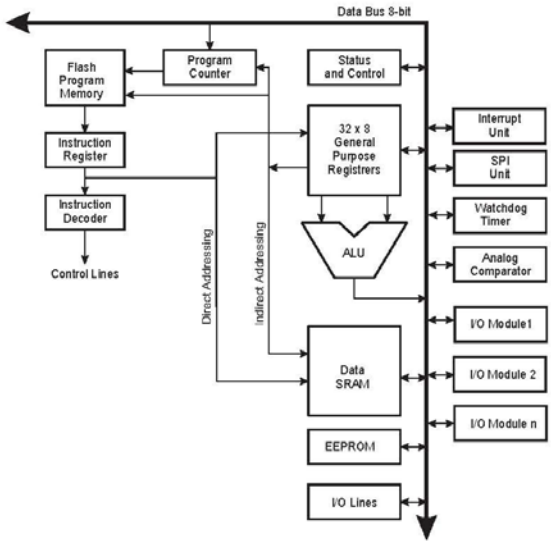
AVR CPU Core

Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Architectural Overview

Figure 3. Block Diagram of the AVR MCU Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 × 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-register, Y-register, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16-bit or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, $20 - $5F.

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.

Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

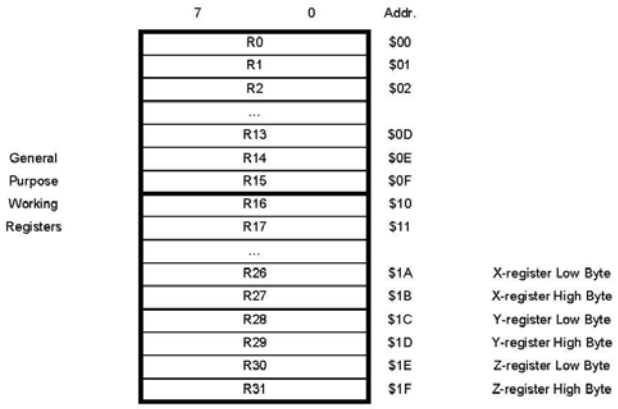
General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 4 shows the structure of the 32 general purpose working registers in the CPU.

Figure 4. AVR CPU General Purpose Working Registers



Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 4, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer Registers can be set to index any register in the file.

The X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as described in Figure 5.

Figure 5. The X-register, Y-register, and Z-register

In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the Instruction Set Reference for details).

Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer. If software reads the Program Counter from the Stack after a call or an interrupt, unused bits (15:13) should be masked out.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU}, directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 6 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 6. The Parallel Instruction Fetches and Instruction Executions

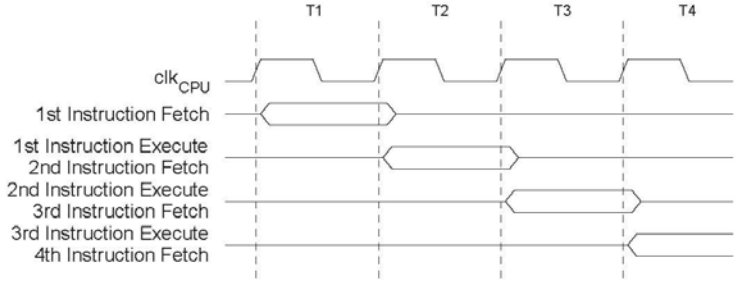
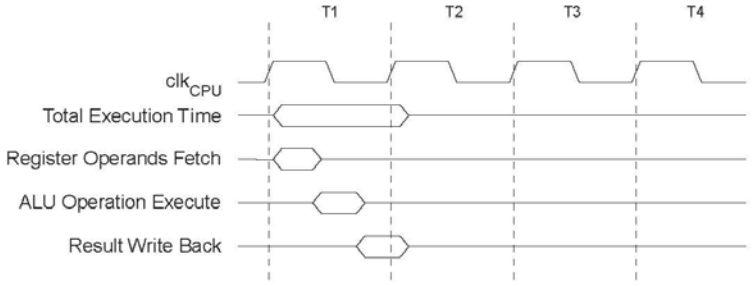


Figure 7 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 7. Single Cycle ALU Operation



Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 259 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 45. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request

0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the General Interrupt Control Register (GICR). Refer to "Interrupts" on page 45 for more information. The Reset Vector can also be moved to the start of the boot Flash section by programming the BOOTRST Fuse, see "Boot Loader Support – Read-While-Write Self-Programming" on page 246.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the global interrupt enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example

```
in  r16, SREG      ; store SREG value
cli     ; disable interrupts during timed sequence
sbi EECR, EEMWE    ; start EEPROM write
sbi EECR, EEWE
out SREG, r16      ; restore SREG value (I-bit)
```

C Code Example

```
char cSREG;
cSREG = SREG;  /* store SREG value */
/* disable interrupts during timed sequence */
_CLI();
EECR |= (1<<EEMWE); /* start EEPROM write */
EECR |= (1<<EEWE);
SREG = cSREG; /* restore SREG value (I-bit) */
```

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in this example.

Assembly Code Example

```
sei   ; set global interrupt enable
sleep ; enter sleep, waiting for interrupt
; note: will enter sleep before any pending
; interrupt(s)
```

C Code Example

```
_SEI(); /* set global interrupt enable */
_SLEEP(); /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
```

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I-bit in SREG is set.

AVR ATmega16 Memories

This section describes the different memories in the ATmega16. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega16 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

In-System Reprogrammable Flash Program Memory

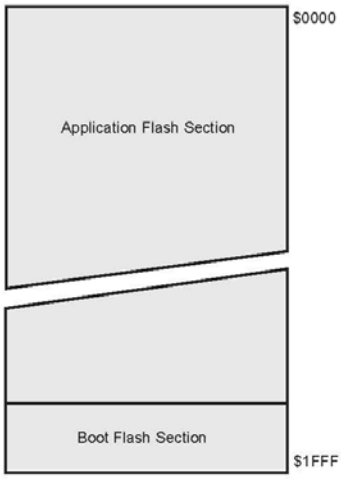
The ATmega16 contains 16 Kbytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 8K × 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega16 Program Counter (PC) is 13 bits wide, thus addressing the 8K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in "Boot Loader Support – Read-While-Write Self-Programming" on page 246. "Memory Programming" on page 259 contains a detailed description on Flash data serial downloading using the SPI pins or the JTAG interface.

Constant tables can be allocated within the entire program memory address space (see the LPM – Load Program Memory Instruction Description).

Timing diagrams for instruction fetch and execution are presented in "Instruction Execution Timing" on page 13.

Figure 8. Program Memory Map



SRAM Data Memory

Figure 9 shows how the ATmega16 SRAM Memory is organized.

The lower 1120 Data Memory locations address the Register File, the I/O Memory, and the internal data SRAM. The first 96 locations address the Register File and I/O Memory, and the next 1024 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

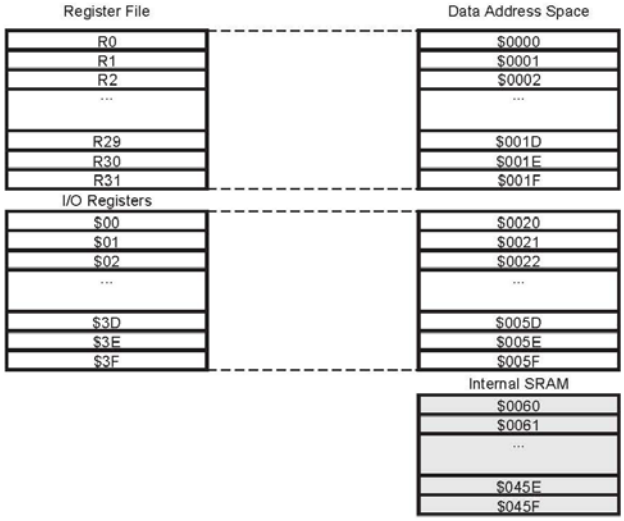
The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y-register or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, and the 1024 bytes of internal data SRAM in the ATmega16 are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 11.

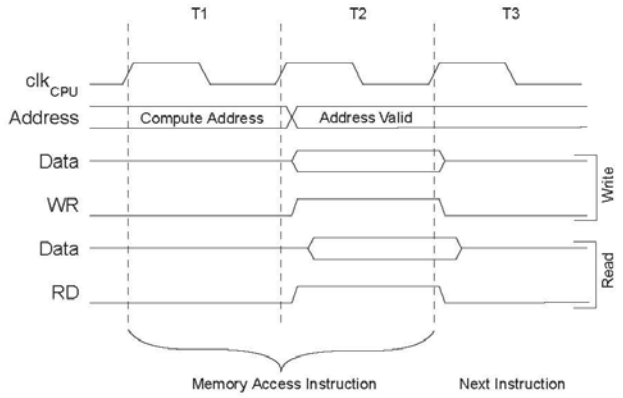
Figure 9. Data Memory Map



Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clk_{CPU} cycles as described in Figure 10.

Figure 10. On-chip Data SRAM Access Cycles



EEPROM Data Memory

The ATmega16 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

For a detailed description of SPI, JTAG, and Parallel data downloading to the EEPROM, see page 273, page 278, and page 262, respectively.

EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space.

The write access time for the EEPROM is given in Table 1. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies, V_{CC} is likely to rise or fall slowly on Power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See "Preventing EEPROM Corruption" on page 22 for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

The EEPROM Address Register – EEARH and EEARL

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	–	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

- **Bits 15..9 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16 and will always read as zero.

- **Bits 8..0 – EEAR8..0: EEPROM Address**

The EEPROM Address Registers – EEARH and EEARL – specify the EEPROM address in the 512 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 511. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

The EEPROM Data Register – EEDR

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..0 – EEDR7.0: EEPROM Data**

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

The EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

- **Bits 7..4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16 and will always read as zero.

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEWE is cleared.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set, setting EEWE within four clock cycles will write data to the EEPROM at the selected address If EEMWE is zero, setting EEWE will have no effect. When EEMWE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for an EEPROM write procedure.

- **Bit 1 – EEWE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be written to one to write the value into the EEPROM. The EEMWE bit must be written to one before a logical one is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEWE becomes zero.
2. Wait until SPMEN in SPMCR becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMWE bit while writing a zero to EEWE in EECR.
6. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted. See "Boot Loader Support – Read-While-Write Self-Programming" on page 246 for details about boot programming.

Caution: An interrupt between step 5 and step 6 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM Access, the EEAR or EEDR reGister will be modified, causing the interrupted EEPROM Access to fail. It is recommended to have the Global Interrupt Flag cleared during all the steps to avoid these problems.

When the write access time has elapsed, the EEWE bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE has been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal – EERE – is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEWE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register.

The calibrated Oscillator is used to time the EEPROM accesses. Table 1 lists the typical programming time for EEPROM access from the CPU.

Table 1. EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles[1]	Typ Programming Time
EEPROM write (from CPU)	8448	8.5 ms

Note: 1. Uses 1 MHz clock, independent of CKSEL Fuse setting.

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (for example by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples

also assume that no Flash Boot Loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.

Assembly Code Example

```
EEPROM_write:
  ; Wait for completion of previous write
  sbic EECR,EEWE
  rjmp EEPROM_write
  ; Set up address (r18:r17) in address register
  out  EEARH, r18
  out  EEARL, r17
  ; Write data (r16) to data register
  out  EEDR,r16
  ; Write logical one to EEMWE
  sbi  EECR,EEMWE
  ; Start eeprom write by setting EEWE
  sbi  EECR,EEWE
  ret
```

C Code Example

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
  /* Wait for completion of previous write */
  while(EECR & (1<<EEWE))
    ;
  /* Set up address and data registers */
  EEAR = uiAddress;
  EEDR = ucData;
  /* Write logical one to EEMWE */
  EECR |= (1<<EEMWE);
  /* Start eeprom write by setting EEWE */
  EECR |= (1<<EEWE);
}
```

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
EEPROM_read:
  ; Wait for completion of previous write
  sbic EECR,EEWE
  rjmp EEPROM_read
  ; Set up address (r18:r17) in address register
  out  EEARH, r18
  out  EEARL, r17
  ; Start eeprom read by writing EERE
  sbi  EECR,EERE
  ; Read data from data register
  in   r16,EEDR
  ret
```

C Code Example

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
  /* Wait for completion of previous write */
  while(EECR & (1<<EEWE))
    ;
  /* Set up address register */
  EEAR = uiAddress;
  /* Start eeprom read by writing EERE */
  EECR |= (1<<EERE);
  /* Return data from data register */
  return EEDR;
}
```

EEPROM Write During Power-down Sleep Mode

When entering Power-down Sleep mode while an EEPROM write operation is active, the EEPROM write operation will continue, and will complete before the Write Access time has passed. However, when the write operation is completed, the Oscillator continues running, and as a consequence, the device does not enter Power-down entirely. It is therefore recommended to verify that the EEPROM write operation is completed before entering Power-down.

Preventing EEPROM Corruption

During periods of low V_{CC}, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD). If the detection level of the internal BOD does not match the needed detection level, an external low V_{CC} Reset Protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.

I/O Memory

The I/O space definition of the ATmega16 is shown in "Register Summary" on page 331.

All ATmega16 I/Os and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the Instruction Set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses $00 - $3F must be used. When addressing I/O Registers as data space using LD and ST instructions, $20 must be added to these addresses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

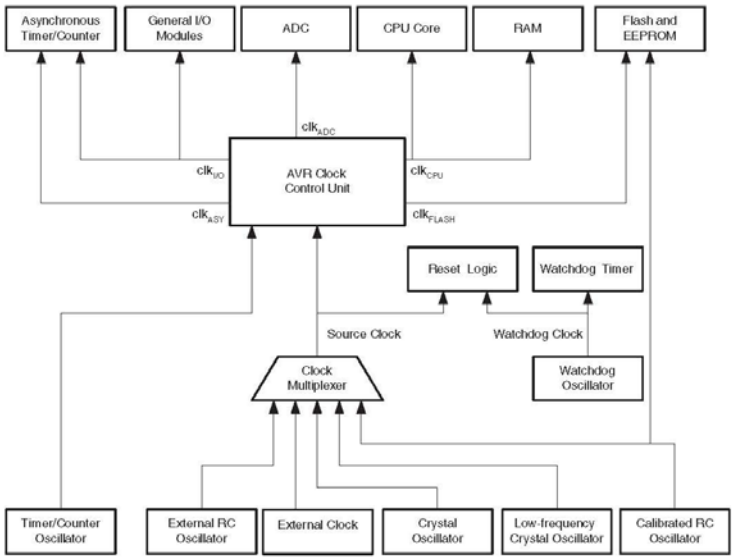
The I/O and Peripherals Control Registers are explained in later sections.

System Clock and Clock Options

Clock Systems and their Distribution

Figure 11 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in "Power Management and Sleep Modes" on page 32. The clock systems are detailed Figure 11.

Figure 11. Clock Distribution



CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that address recognition in the TWI module is carried out asynchronously when $clk_{I/O}$ is halted, enabling TWI address reception in all sleep modes.

Flash Clock – clk_{FLASH}

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

Asynchronous Timer Clock – clk_{ASY}

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external 32 kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.

ADC Clock – clk_{ADC}

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

Clock Sources

The device has the following clock source options, selectable by Flash Fuse bits as shown below. The clock from the selected source is input to the AVR clock generator, and routed to the appropriate modules.

Table 2. Device Clocking Options Select[1]

Device Clocking Option	CKSEL3..0
External Crystal/Ceramic Resonator	1111 - 1010
External Low-frequency Crystal	1001
External RC Oscillator	1000 - 0101
Calibrated Internal RC Oscillator	0100 - 0001
External Clock	0000

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

The various choices for each clocking option is given in the following sections. When the CPU wakes up from Power-down or Power-save, the selected clock source is used to time the start-up, ensuring stable Oscillator operation before instruction execution starts. When the CPU starts from Reset, there is as an additional delay allowing the power to reach a stable level before commencing normal operation. The Watchdog Oscillator is used for timing this real-time part of the start-up time. The number of WDT Oscillator cycles used for each time-out is shown in Table 3. The frequency of the Watchdog Oscillator is voltage dependent as shown in "ATmega16 Typical Characteristics" on page 299.

Table 3. Number of Watchdog Oscillator Cycles

Typ Time-out (V_{CC} = 5.0V)	Typ Time-out (V_{CC} = 3.0V)	Number of Cycles
4.1 ms	4.3 ms	4K (4,096)
65 ms	69 ms	64K (65,536)

Default Clock Source

The device is shipped with CKSEL = "0001" and SUT = "10". The default clock source setting is therefore the 1 MHz Internal RC Oscillator with longest startup time. This default setting ensures that all users can make their desired clock source setting using an In-System or Parallel Programmer.

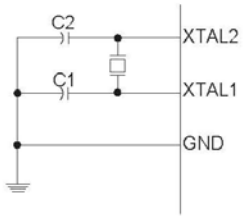
Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in Figure 12. Either a quartz crystal or a ceramic resonator may be used. The CKOPT Fuse selects between two different Oscillator amplifier modes. When CKOPT is programmed, the Oscillator output will oscillate will a full rail-to-rail swing on the output. This mode is suitable when operating in a very noisy environment or when the output from XTAL2 drives a second clock buffer. This mode has a wide frequency range. When CKOPT is unprogrammed, the Oscillator has a smaller output swing. This reduces power consumption considerably. This mode has a limited frequency range and it can not be used to drive other clock buffers.

For resonators, the maximum frequency is 8 MHz with CKOPT unprogrammed and 16 MHz with CKOPT programmed. C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for

choosing capacitors for use with crystals are given in Table 4. For ceramic resonators, the capacitor values given by the manufacturer should be used.

Figure 12. Crystal Oscillator Connections



The Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..1 as shown in Table 4.

Table 4. Crystal Oscillator Operating Modes

CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101(1)	0.4 - 0.9	–
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

Note: 1. This option should not be used with crystals, only with ceramic resonators.

The CKSEL0 Fuse together with the SUT1..0 Fuses select the start-up times as shown in Table 5.

Table 5. Start-up Times for the Crystal Oscillator Clock Selection

CKSEL0	SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
0	00	258 CK[1]	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK[1]	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK[2]	–	Ceramic resonator, BOD enabled
0	11	1K CK[2]	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK[2]	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	–	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

Notes:
1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

Low-frequency Crystal Oscillator

To use a 32.768 kHz watch crystal as the clock source for the device, the Low-frequency Crystal Oscillator must be selected by setting the CKSEL Fuses to "1001". The crystal should be connected as shown in Figure 12. By programming the CKOPT Fuse, the user can enable internal capacitors on XTAL1 and XTAL2, thereby removing the need for external capacitors. The internal capacitors have a nominal value of 36 pF.

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 6.

Table 6. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

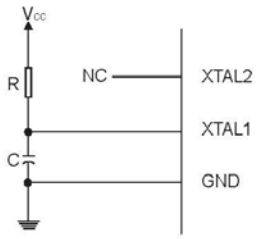
SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	1K CK(1)	4.1 ms	Fast rising power or BOD enabled
01	1K CK(1)	65 ms	Slowly rising power
10	32K CK	65 ms	Stable frequency at start-up
11	Reserved		

Note: 1. These options should only be used if frequency stability at start-up is not important for the application.

External RC Oscillator

For timing insensitive applications, the external RC configuration shown in Figure 13 can be used. The frequency is roughly estimated by the equation f = 1/(3RC). C should be at least 22 pF. By programming the CKOPT Fuse, the user can enable an internal 36 pF capacitor between XTAL1 and GND, thereby removing the need for an external capacitor. For more information on Oscillator operation and details on how to choose R and C, refer to the External RC Oscillator application note.

Figure 13. External RC Configuration



The Oscillator can operate in four different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..0 as shown in Table 7.

Table 7. External RC Oscillator Operating Modes

CKSEL3..0	Frequency Range (MHz)
0101	0.1 ≤ 0.9
0110	0.9 - 3.0
0111	3.0 - 8.0
1000	8.0 - 12.0

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 8.

Table 8. Start-up Times for the External RC Oscillator Clock Selection

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	18 CK	–	BOD enabled
01	18 CK	4.1 ms	Fast rising power
10	18 CK	65 ms	Slowly rising power
11	6 CK(1)	4.1 ms	Fast rising power or BOD enabled

Note: 1. This option should not be used when operating close to the maximum frequency of the device.

Calibrated Internal RC Oscillator

The Calibrated Internal RC Oscillator provides a fixed 1.0 MHz, 2.0 MHz, 4.0 MHz, or 8.0 MHz clock. All frequencies are nominal values at 5V and 25°C. This clock may be selected as the system-clock by programming the CKSEL Fuses as shown in Table 9. If selected, it will operate with no external components. The CKOPT Fuse should always be unpro-grammed when using this clock option. During Reset, hardware loads the calibration byte into the OSCCAL Register and thereby automatically calibrates the RC Oscillator. At 5V, 25°C and 1.0 MHz, 2.0 MHz, 4.0 MHz or 8.0 MHz Oscillator frequency selected, this calibration gives a frequency within ±3% of the nominal frequency. Using calibration methods as described in application notes available at www.atmel.com/avr it is possible to achieve ±1% accuracy at any given V_{CC} and Temperature. When this Oscillator is used as the Chip Clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the reset time-out. For more information on the pre-programmed calibration value, see the section "Calibration Byte" on page 261.

Table 9. Internal Calibrated RC Oscillator Operating Modes

CKSEL3..0	Nominal Frequency (MHz)
0001(1)	1.0
0010	2.0
0011	4.0
0100	8.0

Note: 1. The device is shipped with this option selected.

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 10. XTAL1 and XTAL2 should be left unconnected (NC).

Table 10. Start-up Times for the Internal Calibrated RC Oscillator Clock Selection

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	6 CK	–	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10(1)	6 CK	65 ms	Slowly rising power
11	Reserved		

Note: 1. The device is shipped with this option selected.

Oscillator Calibration Register – OSCCAL

Bit	7	6	5	4	3	2	1	0	
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

- **Bits 7..0 – CAL7..0: Oscillator Calibration Value**

Writing the calibration byte to this address will trim the Internal Oscillator to remove process variations from the Oscillator frequency. This is done automatically during Chip Reset. When OSCCAL is zero, the lowest available frequency is chosen. Writing non-zero values to this register will increase the frequency of the Internal Oscillator. Writing $FF to the register gives the highest available frequency. The calibrated Oscillator is used to time EEPROM and Flash access. If EEPROM or Flash is written, do not calibrate to more than 10% above the nominal frequency. Otherwise, the EEPROM or Flash write may fail. Note that the Oscillator is intended for calibration to 1.0 MHz, 2.0 MHz, 4.0 MHz, or 8.0 MHz. Tuning to other values is not guaranteed, as indicated in Table 11.

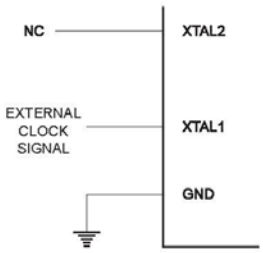
Table 11. Internal RC Oscillator Frequency Range.

OSCCAL Value	Min Frequency in Percentage of Nominal Frequency (%)	Max Frequency in Percentage of Nominal Frequency (%)
$00	50	100
$7F	75	150
$FF	100	200

External Clock

To drive the device from an external clock source, XTAL1 should be driven as shown in Figure 14. To run the device on an external clock, the CKSEL Fuses must be programmed to "0000". By programming the CKOPT Fuse, the user can enable an internal 36 pF capacitor between XTAL1 and GND.

Figure 14. External Clock Drive Configuration



When this clock source is selected, start-up times are determined by the SUT Fuses as shown in Table 12.

Table 12. Start-up Times for the External Clock Selection

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	6 CK	–	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10	6 CK	65 ms	Slowly rising power
11	Reserved		

When applying an external clock, it is required to avoid sudden changes in the applied clock frequency to ensure stable operation of the MCU. A variation in frequency of more than 2% from one clock cycle to the next can lead to unpredictable behavior. It is required to ensure that the MCU is kept in reset during such changes in the clock frequency.

Timer/Counter Oscillator

For AVR microcontrollers with Timer/Counter Oscillator pins (TOSC1 and TOSC2), the crystal is connected directly between the pins. No external capacitors are needed. The Oscillator is optimized for use with a 32.768 kHz watch crystal. Applying an external clock source to TOSC1 is not recommended.

Note: The Timer/Counter Oscillator uses the same type of crystal oscillator as Low-Frequency Oscillator and the internal capacitors have the same nominal value of 36 pF.

Power Management and Sleep Modes

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

To enter any of the six sleep modes, the SE bit in MCUCR must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the MCUCR Register select which sleep mode (Idle, ADC Noise Reduction, Power-down, Power-save, Standby, or Extended Standby) will be activated by the SLEEP instruction. See Table 13 for a summary. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, it executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File and SRAM are unaltered when the device wakes up from sleep. If a Reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Figure 11 on page 24 presents the different clock systems in the ATmega16, and their distribution. The figure is helpful in selecting an appropriate sleep mode.

MCU Control Register – MCUCR

The MCU Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7, 5, 4 – SM2..0: Sleep Mode Select Bits 2, 1, and 0**

These bits select between the six available sleep modes as shown in Table 13.

Table 13. Sleep Mode Select

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby(1)
1	1	1	Extended Standby(1)

Note: 1. Standby mode and Extended Standby mode are only available with external crystals or resonators.

- **Bit 6 – SE: Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

Idle Mode

When the SM2..0 bits are written to 000, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing SPI, USART, Analog Comparator, ADC, Two-wire Serial Interface, Timer/Counters, Watchdog, and the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH}, while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow and USART Transmit Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register – ACSR. This will reduce power consumption in Idle mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

ADC Noise Reduction Mode

When the SM2..0 bits are written to 001, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the ADC, the External Interrupts, the Two-wire Serial Interface address watch, Timer/Counter2 and the Watchdog to continue operating (if enabled). This sleep mode basically halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart form the ADC Conversion Complete interrupt, only an External Reset, a Watchdog Reset, a Brown-out Reset, a Two-wire Serial Interface Address Match Interrupt, a Timer/Counter2 interrupt, an SPM/EEPROM ready interrupt, an External level interrupt on INT0 or INT1, or an external interrupt on INT2 can wake up the MCU from ADC Noise Reduction mode.

Power-down Mode

When the SM2..0 bits are written to 010, the SLEEP instruction makes the MCU enter Power-down mode. In this mode, the External Oscillator is stopped, while the External interrupts, the Two-wire Serial Interface address watch, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog Reset, a Brown-out Reset, a Two-wire Serial Interface address match interrupt, an External level interrupt on INT0 or INT1, or an External interrupt on INT2 can wake up the MCU. This sleep mode basically halts all generated clocks, allowing operation of asynchronous modules only.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. Refer to "External Interrupts" on page 68 for details.

When waking up from Power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL Fuses that define the reset time-out period, as described in "Clock Sources" on page 25.

Power-save Mode

When the SM2..0 bits are written to 011, the SLEEP instruction makes the MCU enter Power-save mode. This mode is identical to Power-down, with one exception:

If Timer/Counter2 is clocked asynchronously, that is, the AS2 bit in ASSR is set, Timer/Counter2 will run during sleep. The device can wake up from either Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK, and the Global Interrupt Enable bit in SREG is set.

If the Asynchronous Timer is NOT clocked asynchronously, Power-down mode is recommended instead of Power-save mode because the contents of the registers in the Asynchronous Timer should be considered undefined after wake-up in Power-save mode if AS2 is 0.

This sleep mode basically halts all clocks except clk_{ASY}, allowing operation only of asynchronous modules, including Timer/Counter2 if clocked asynchronously.

Standby Mode

When the SM2..0 bits are 110 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in six clock cycles.

Extended Standby Mode

When the SM2..0 bits are 111 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Extended Standby mode. This mode is identical to Power-save mode with the exception that the Oscillator is kept running. From Extended Standby mode, the device wakes up in six clock cycles..

Table 14. Active Clock Domains and Wake Up Sources in the Different Sleep Modes

	Active Clock domains					Oscillators		Wake-up Sources					
Sleep Mode	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Osc. Enabled	INT2 INT1 INT0	TWI Address Match	Timer 2	SPM / EEPROM Ready	ADC	Other I/O
Idle			X	X	X	X	X[2]	X	X	X	X	X	X
ADC Noise Reduction				X	X	X	X[2]	X[3]	X	X	X	X	
Power Down								X[3]	X				
Power Save					X[2]		X[2]	X[3]	X	X[2]			
Standby[1]						X		X[3]	X				
Extended Standby[1]					X[2]	X	X[2]	X[3]	X	X[2]			

Notes: 1. External Crystal or resonator selected as clock source.
2. If AS2 bit in ASSR is set.
3. Only INT2 or level interrupt INT1 and INT0.

Minimizing Power Consumption

There are several issues to consider when trying to minimize the power consumption in an AVR controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device's functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

Analog to Digital Converter

If enabled, the ADC will be enabled in all sleep modes. To save power, the ADC should be disabled before entering any sleep mode. When the ADC is turned off and on again, the next conversion will be an extended conversion. Refer to "Analog to Digital Converter" on page 204 for details on ADC operation.

Analog Comparator

When entering Idle mode, the Analog Comparator should be disabled if not used. When entering ADC Noise Reduction mode, the Analog Comparator should be disabled. In the other sleep modes, the Analog Comparator is automatically disabled. However, if the Analog Comparator is set up to use the Internal Voltage Reference as input, the Analog Comparator should be disabled in all sleep modes. Otherwise, the Internal Voltage Reference will be enabled, independent of sleep mode. Refer to "Analog Comparator" on page 201 for details on how to configure the Analog Comparator.

Brown-out Detector

If the Brown-out Detector is not needed in the application, this module should be turned off. If the Brown-out Detector is enabled by the BODEN Fuse, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to "Brown-out Detection" on page 40 for details on how to configure the Brown-out Detector.

Internal Voltage Reference

The Internal Voltage Reference will be enabled when needed by the Brown-out Detector, the Analog Comparator or the ADC. If these modules are disabled as described in the sections above, the internal voltage reference will be disabled and it will not be consuming power. When turned on again, the user must allow the reference to start up before the output is used. If the reference is kept on in sleep mode, the output can be used immediately. Refer to "Internal Voltage Reference" on page 42 for details on the start-up time.

Watchdog Timer

If the Watchdog Timer is not needed in the application, this module should be turned off. If the Watchdog Timer is enabled, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to "Watchdog Timer" on page 42 for details on how to configure the Watchdog Timer.

Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important thing is then to ensure that no pins drive resistive loads. In sleep modes where the both the I/O clock ($clk_{I/O}$) and the ADC clock (clk_{ADC}) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. Refer to the section "Digital Input Enable and Sleep Modes" on page 54 for details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to $V_{CC}/2$, the input buffer will use excessive power.

JTAG Interface and On-chip Debug System

If the On-chip debug system is enabled by the OCDEN Fuse and the chip enter Power down or Power save sleep mode, the main clock source remains enabled. In these sleep modes, this will contribute significantly to the total current consumption. There are three alternative ways to avoid this:

- Disable OCDEN Fuse.
- Disable JTAGEN Fuse.
- Write one to the JTD bit in MCUCSR.

The TDO pin is left floating when the JTAG interface is enabled while the JTAG TAP controller is not shifting data. If the hardware connected to the TDO pin does not pull up the logic level, power consumption will increase. Note that the TDI pin for the next device in the scan chain contains a pull-up that avoids this problem. Writing the JTD bit in the MCUCSR register to one or leaving the JTAG fuse unprogrammed disables the JTAG interface.

System Control and Reset

Resetting the AVR

During Reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. The instruction placed at the Reset Vector must be a JMP – absolute jump – instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa. The circuit diagram in Figure 15 shows the reset logic. Table 15 defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

After all reset sources have gone inactive, a delay counter is invoked, stretching the Internal Reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the CKSEL Fuses. The different selections for the delay period are presented in "Clock Sources" on page 25.

Reset Sources

The ATmega16 has five sources of reset:

- **Power-on Reset.**
  The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- **External Reset.**
  The MCU is reset when a low level is present on the $\overline{\text{RESET}}$ pin for longer than the minimum pulse length.
- **Watchdog Reset.**
  The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.
- **Brown-out Reset.**
  The MCU is reset when the supply voltage V_{CC} is below the Brown-out Reset threshold (V_{BOT}) and the Brown-out Detector is enabled.
- **JTAG AVR Reset.**
  The MCU is reset as long as there is a logic one in the Reset Register, one of the scan chains of the JTAG system. Refer to the section "IEEE 1149.1 (JTAG) Boundary-scan" on page 228 for details.

Figure 15. Reset Logic

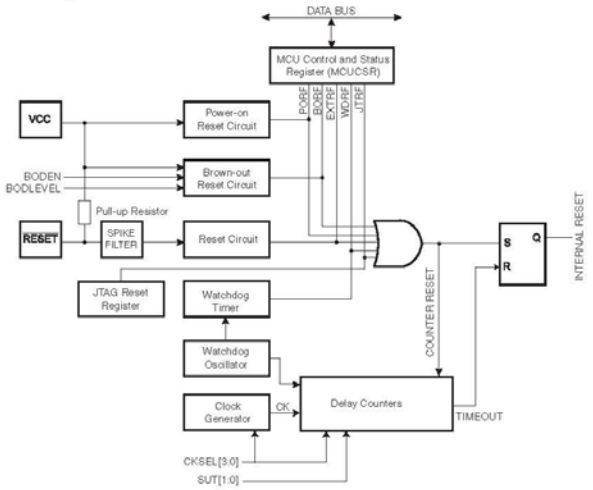


Table 15. Reset Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling)[1]			1.3	2.3	
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		$0.1V_{CC}$		$0.9V_{CC}$	
t_{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin				1.5	µs
V_{BOT}	Brown-out Reset Threshold Voltage[2]	BODLEVEL = 1	2.5	2.7	3.2	V
		BODLEVEL = 0	3.6	4.0	4.5	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		µs
		BODLEVEL = 0		2		
V_{HYST}	Brown-out Detector hysteresis			50		mV

Notes: 1. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).
2. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to $V_{CC} = V_{BOT}$ during the production test. This guarantees that a Brown-out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 1 for ATmega16L and BODLEVEL = 0 for ATmega16. BODLEVEL = 1 is not applicable for ATmega16.

Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 15. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after V_{CC} rise. The RESET signal is activated again, without any delay, when V_{CC} decreases below the detection level.

Figure 16. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}.

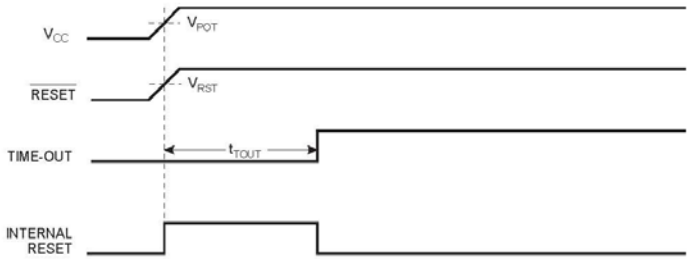
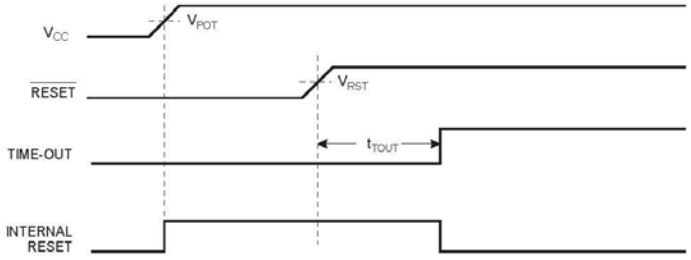


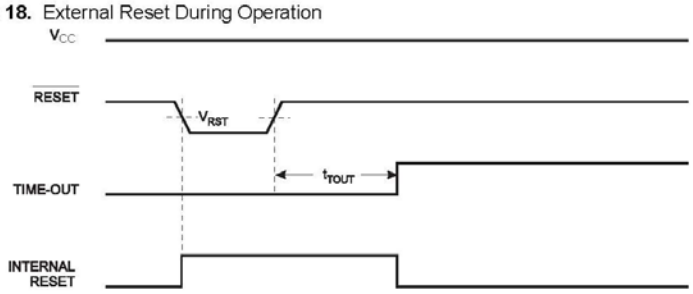
Figure 17. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than the minimum pulse width (see Table 15) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} – on its positive edge, the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

Figure 18. External Reset During Operation



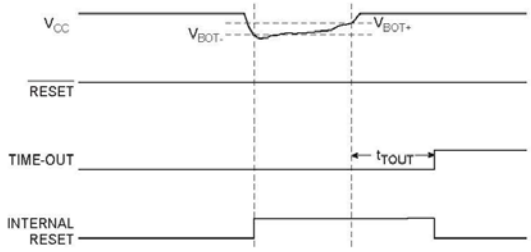
Brown-out Detection

ATmega16 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} - V_{HYST}/2$.

The BOD circuit can be enabled/disabled by the fuse BODEN. When the BOD is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 19), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 19), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in Table 15.

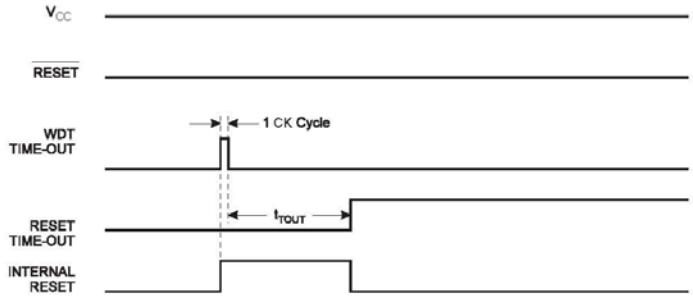
Figure 19. Brown-out Reset During Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to page 42 for details on operation of the Watchdog Timer.

Figure 20. Watchdog Reset During Operation



MCU Control and Status Register – MCUCSR

The MCU Control and Status Register provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	See Bit Description					

- **Bit 4 – JTRF: JTAG Reset Flag**

This bit is set if a reset is being caused by a logic one in the JTAG Reset Register selected by the JTAG instruction AVR_RESET. This bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then reset the MCUCSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

Internal Voltage Reference

ATmega16 features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC. The 2.56V reference to the ADC is generated from the internal bandgap reference.

Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. The start-up time is given in Table 16. To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODEN Fuse).
2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR).
3. When the ADC is enabled.

Thus, when the BOD is not enabled, after setting the ACBG bit or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used. To reduce power consumption in Power-down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-down mode.

Table 16. Internal Voltage Reference Characteristics

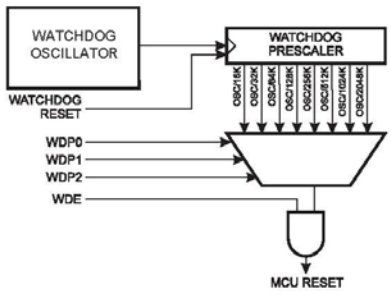
Symbol	Parameter	Min	Typ	Max	Units
V_{BG}	Bandgap reference voltage	1.15	1.23	1.4	V
t_{BG}	Bandgap reference start-up time		40	70	µs
I_{BG}	Bandgap reference current consumption		10		µA

Watchdog Timer

The Watchdog Timer is clocked from a separate On-chip Oscillator which runs at 1 MHz. This is the typical value at V_{CC} = 5V. See characterization data for typical values at other V_{CC} levels. By controlling the Watchdog Timer prescaler, the Watchdog Reset interval can be adjusted as shown in Table 17 on page 43. The WDR – Watchdog Reset – instruction resets the Watchdog Timer. The Watchdog Timer is also reset when it is disabled and when a Chip Reset occurs. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog Reset, the ATmega16 resets and executes from the Reset Vector. For timing details on the Watchdog Reset, refer to page 41.

To prevent unintentional disabling of the Watchdog, a special turn-off sequence must be followed when the Watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details.

Figure 21. Watchdog Timer



Watchdog Timer Control Register – WDTCR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..5 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16 and will always read as zero.

- **Bit 4 – WDTOE: Watchdog Turn-off Enable**

This bit must be set when the WDE bit is written to logic zero. Otherwise, the Watchdog will not be disabled. Once written to one, hardware will clear this bit after four clock cycles. Refer to the description of the WDE bit for a Watchdog disable procedure.

- **Bit 3 – WDE: Watchdog Enable**

When the WDE is written to logic one, the Watchdog Timer is enabled, and if the WDE is written to logic zero, the Watchdog Timer function is disabled. WDE can only be cleared if the WDTOE bit has logic level one. To disable an enabled Watchdog Timer, the following procedure must be followed:

1. In the same operation, write a logic one to WDTOE and WDE. A logic one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logic 0 to WDE. This disables the Watchdog.

- **Bits 2..0 – WDP2, WDP1, WDP0: Watchdog Timer Prescaler 2, 1, and 0**

The WDP2, WDP1, and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 17.

Table 17. Watchdog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 3.0V	Typical Time-out at V_{CC} = 5.0V
0	0	0	16K (16,384)	17.1 ms	16.3 ms
0	0	1	32K (32,768)	34.3 ms	32.5 ms
0	1	0	64K (65,536)	68.5 ms	65 ms
0	1	1	128K (131,072)	0.14 s	0.13 s
1	0	0	256K (262,144)	0.27 s	0.26 s
1	0	1	512K (524,288)	0.55 s	0.52 s
1	1	0	1,024K (1,048,576)	1.1 s	1.0 s
1	1	1	2,048K (2,097,152)	2.2 s	2.1 s

The following code example shows one assembly and one C function for turning off the WDT. The example assumes that interrupts are controlled (for example by disabling interrupts globally) so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
WDT_off:
  ; Reset WDT
  WDR
  ; Write logical one to WDTOE and WDE
  in   r16, WDTCR
  ori r16, (1<<WDTOE)|(1<<WDE)
  out  WDTCR, r16
  ; Turn off WDT
  ldi  r16, (0<<WDE)
  out  WDTCR, r16
  ret
```

C Code Example

```
void WDT_off(void)
{
  /* Reset WDT*/
  _WDR();
  /* Write logical one to WDTOE and WDE */
  WDTCR |= (1<<WDTOE) | (1<<WDE);
  /* Turn off WDT */
  WDTCR = 0x00;
}
```

Interrupts

This section describes the specifics of the interrupt handling as performed in ATmega16. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 13.

Interrupt Vectors in ATmega16

Table 18. Reset and Interrupt Vectors

Vector No.	Program Address[2]	Source	Interrupt Definition
1	$000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$002	INT0	External Interrupt Request 0
3	$004	INT1	External Interrupt Request 1
4	$006	TIMER2 COMP	Timer/Counter2 Compare Match
5	$008	TIMER2 OVF	Timer/Counter2 Overflow
6	$00A	TIMER1 CAPT	Timer/Counter1 Capture Event
7	$00C	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	$00E	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	$010	TIMER1 OVF	Timer/Counter1 Overflow
10	$012	TIMER0 OVF	Timer/Counter0 Overflow
11	$014	SPI, STC	Serial Transfer Complete
12	$016	USART, RXC	USART, Rx Complete
13	$018	USART, UDRE	USART Data Register Empty
14	$01A	USART, TXC	USART, Tx Complete
15	$01C	ADC	ADC Conversion Complete
16	$01E	EE_RDY	EEPROM Ready
17	$020	ANA_COMP	Analog Comparator
18	$022	TWI	Two-wire Serial Interface
19	$024	INT2	External Interrupt Request 2
20	$026	TIMER0 COMP	Timer/Counter0 Compare Match
21	$028	SPM_RDY	Store Program Memory Ready

Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 246.
2. When the IVSEL bit in GICR is set, interrupt vectors will be moved to the start of the Boot Flash section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash section.

Table 19 shows Reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 19. Reset and Interrupt Vectors Placement[1]

BOOTRST	IVSEL	Reset address	Interrupt Vectors Start Address
1	0	$0000	$0002
1	1	$0000	Boot Reset Address + $0002
0	0	Boot Reset Address	$0002
0	1	Boot Reset Address	Boot Reset Address + $0002

Note: 1. The Boot Reset Address is shown in Table 100 on page 257. For the BOOTRST Fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega16 is:

```
Address   Labels   Code                 Comments
$000              jmp   RESET          ; Reset Handler
$002              jmp   EXT_INT0       ; IRQ0 Handler
$004              jmp   EXT_INT1       ; IRQ1 Handler
$006              jmp   TIM2_COMP      ; Timer2 Compare Handler
$008              jmp   TIM2_OVF       ; Timer2 Overflow Handler
$00A              jmp   TIM1_CAPT      ; Timer1 Capture Handler
$00C              jmp   TIM1_COMPA     ; Timer1 CompareA Handler
$00E              jmp   TIM1_COMPB     ; Timer1 CompareB Handler
$010              jmp   TIM1_OVF       ; Timer1 Overflow Handler
$012              jmp   TIM0_OVF       ; Timer0 Overflow Handler
$014              jmp   SPI_STC        ; SPI Transfer Complete Handler
$016              jmp   USART_RXC      ; USART RX Complete Handler
$018              jmp   USART_UDRE     ; UDR Empty Handler
$01A              jmp   USART_TXC      ; USART TX Complete Handler
$01C              jmp   ADC            ; ADC Conversion Complete Handler
$01E              jmp   EE_RDY         ; EEPROM Ready Handler
$020              jmp   ANA_COMP       ; Analog Comparator Handler
$022              jmp   TWSI           ; Two-wire Serial Interface Handler
$024              jmp   EXT_INT2       ; IRQ2 Handler
$026              jmp   TIM0_COMP      ; Timer0 Compare Handler
$028              jmp   SPM_RDY        ; Store Program Memory Ready Handler
;
$02A     RESET:   ldi   r16,high(RAMEND) ; Main program start
$02B              out   SPH,r16        ; Set Stack Pointer to top of RAM
$02C              ldi   r16,low(RAMEND)
$02D              out   SPL,r16
$02E              sei                  ; Enable interrupts
$02F              <instr>  xxx
...       ...     ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2 Kbytes and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address   Labels   Code                  Comments
$000      RESET:   ldi   r16,high(RAMEND) ; Main program start
$001               out   SPH,r16          ; Set Stack Pointer to top of RAM
$002             ldi   r16,low(RAMEND)
$003               out   SPL,r16
$004               sei                   ; Enable interrupts
$005               <instr>  xxx
;
.org $1C02
$1C02              jmp   EXT_INT0        ; IRQ0 Handler
$1C04              jmp   EXT_INT1        ; IRQ1 Handler
...       ....     ..                    ;
$1C28              jmp   SPM_RDY         ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2 Kbytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address   Labels          Code                    Comments
.org $002
$002               jmp   EXT_INT0        ; IRQ0 Handler
$004               jmp   EXT_INT1        ; IRQ1 Handler
...       ....     ..                    ;
$028               jmp   SPM_RDY         ; Store Program Memory Ready Handler
;
.org $1C00
$1C00     RESET:   ldi   r16,high(RAMEND) ; Main program start
$1C01              out   SPH,r16          ; Set Stack Pointer to top of RAM
$1C02              ldi   r16,low(RAMEND)
$1C03              out   SPL,r16
$1C04              sei                   ; Enable interrupts
$1C05              <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2 Kbytes and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address   Labels   Code                    Comments
.org $1C00
$1C00              jmp   RESET         ; Reset handler
$1C02              jmp   EXT_INT0      ; IRQ0 Handler
$1C04              jmp   EXT_INT1      ; IRQ1 Handler
...       ....     ..                  ;
$1C28              jmp   SPM_RDY       ; Store Program Memory Ready Handler
;
$1C2A     RESET:   ldi   r16,high(RAMEND) ; Main program start
$1C2B              out   SPH,r16          ; Set Stack Pointer to top of RAM
$1C2C              ldi   r16,low(RAMEND)
$1C2D              out   SPL,r16
$1C2E              sei                   ; Enable interrupts
$1C2F              <instr>  xxx
```

Moving Interrupts Between Application and Boot Space

The General Interrupt Control Register controls the placement of the Interrupt Vector table.

General Interrupt Control Register – GICR

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – IVSEL: Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the Interrupt Vectors are placed at the start of the Flash memory. When this bit is set (one), the interrupt vectors are moved to the beginning of the Boot Loader section of the Flash. The actual address of the start of the Boot Flash section is determined by the BOOTSZ Fuses. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 246 for details. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit:

1. Write the Interrupt Vector Change Enable (IVCE) bit to one.
2. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the Status Register is unaffected by the automatic disabling.

Note: If Interrupt Vectors are placed in the Boot Loader section and Boot Lock bit BLB02 is programmed, interrupts are disabled while executing from the Application section. If Interrupt Vectors are placed in the Application section and Boot Lock bit BLB12 is programed, interrupts are disabled while executing from the Boot Loader section. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 246 for details on Boot Lock bits.

- **Bit 0 – IVCE: Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See Code Example below

Assembly Code Example

```
Move_interrupts:
  ; Enable change of interrupt vectors
  ldi  r16, (1<<IVCE)
  out  GICR, r16
  ; Move interrupts to boot Flash section
  ldi  r16, (1<<IVSEL)
  out  GICR, r16
  ret
```

C Code Example

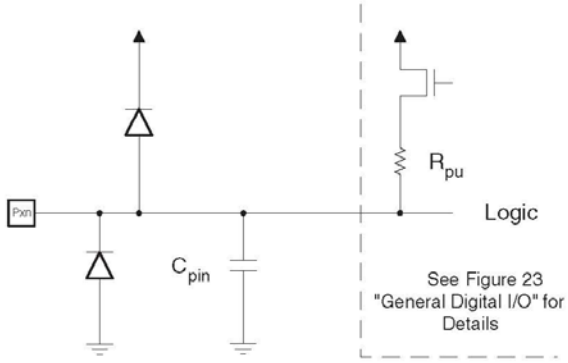
```
void Move_interrupts(void)
{
  /* Enable change of interrupt vectors */
  GICR = (1<<IVCE);
  /* Move interrupts to boot Flash section */
  GICR = (1<<IVSEL);
}
```

I/O Ports

Introduction

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 22. Refer to "Electrical Characteristics" on page 291 for a complete list of parameters.

Figure 22. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case "x" represents the numbering letter for the port, and a lower case "n" represents the bit number. However, when using the register or bit defines in a program, the precise form must be used, that is, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in "Register Description for I/O Ports" on page 66.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. In addition, the Pull-up Disable – PUD bit in SFIOR disables the pull-up function for all pins in all ports when set.

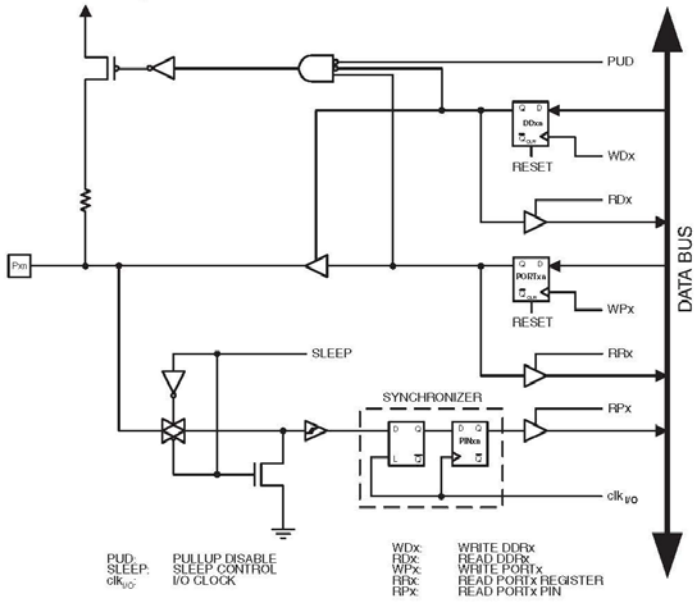
Using the I/O port as General Digital I/O is described in "Ports as General Digital I/O" on page 50. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in "Alternate Port Functions" on page 55. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 23 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 23. General Digital I/O[1]



Note: 1. WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports.

Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in "Register Description for I/O Ports" on page 66, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled ({DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the SFIOR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.

Table 20 summarizes the control signals for the pin value.

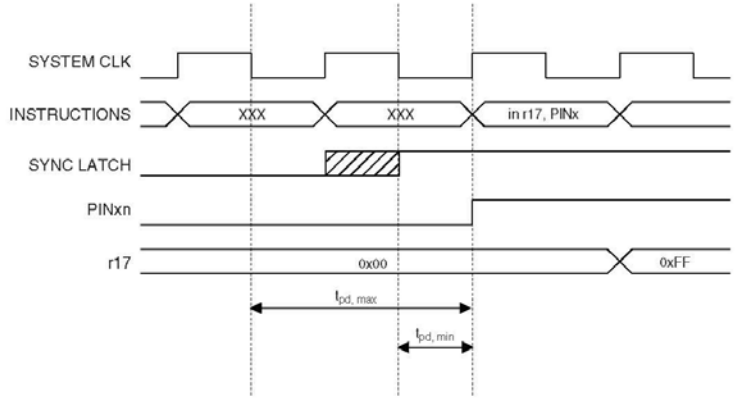
Table 20. Port Pin Configurations

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 23, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 24 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

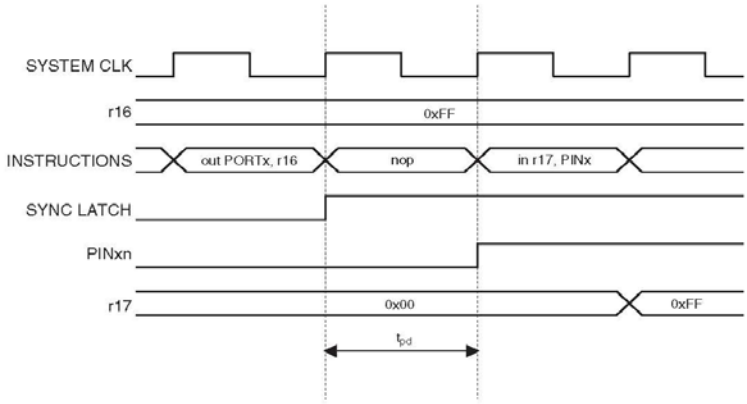
Figure 24. Synchronization when Reading an Externally Applied Pin Value



Consider the clock period starting shortly *after* the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the "SYNC LATCH" signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows $t_{pd,max}$ and $t_{pd,min}$, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a *nop* instruction must be inserted as indicated in Figure 25. The *out* instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay t_{pd} through the synchronizer is one system clock period.

Figure 25. Synchronization when Reading a Software Assigned Pin Value



The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a *nop* instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example[1]

```
...
; Define pull-ups and set outputs high
; Define directions for port pins
ldi   r16,(1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0)
ldi   r17,(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
out   PORTB,r16
out   DDRB,r17
; Insert nop for synchronization
nop
; Read port pins
in    r16,PINB
...
```

C Code Example[1]

```
unsigned char i;
...
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Insert nop for synchronization*/
_NOP();
/* Read port pins */
i = PINB;
...
```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1, 6, and 7, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

Digital Input Enable and Sleep Modes

As shown in Figure 23, the digital input signal can be clamped to ground at the input of the schmitt-trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in Power-down mode, Power-save mode, Standby mode, and Extended Standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as External Interrupt pins. If the External Interrupt Request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in "Alternate Port Functions" on page 55.

If a logic high level ("one") is present on an Asynchronous External Interrupt pin configured as "Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin" while the External Interrupt is *not* enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned sleep modes, as the clamping in these sleep modes produces the requested logic change.

Unconnected pins

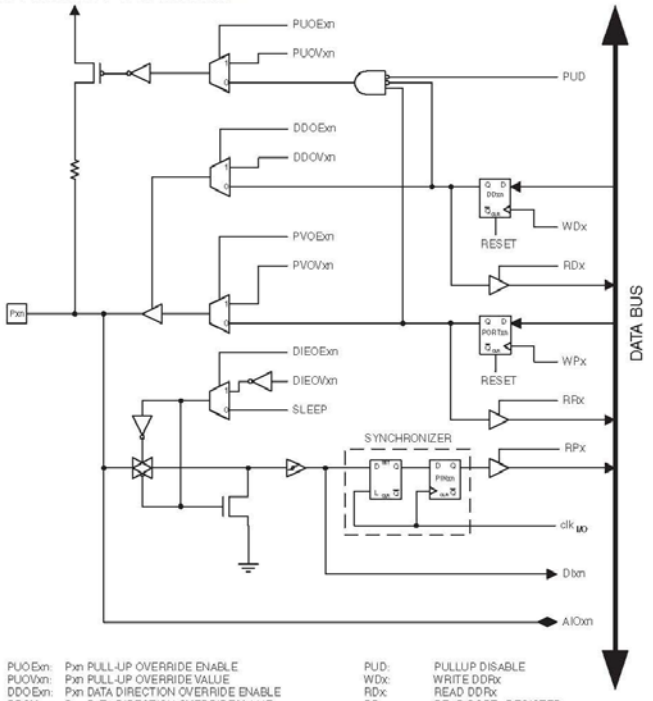
If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

Alternate Port Functions

Most port pins have alternate functions in addition to being General Digital I/Os. Figure 26 shows how the port pin control signals from the simplified Figure 23 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Figure 26. Alternate Port Functions(1)



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WPx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE	$clk_{I/O}$:	I/O CLOCK
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE	Dixn:	DIGITAL INPUT PIN n ON PORTx
SLEEP:	SLEEP CONTROL	AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Note: 1. WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

Table 21 summarizes the function of the overriding signals. The pin and port indexes from Figure 26 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 21. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE is set, the Output Driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVOE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn Register bit.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU-state (Normal Mode, sleep modes).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (Normal Mode, sleep modes).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the schmitt trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/ output	This is the Analog Input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

Special Function I/O Register – SFIOR

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – PUD: Pull-up disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See "Configuring the Pin" on page 51 for more details about this feature.

Alternate Functions of Port A

Port A has an alternate function as analog input for the ADC as shown in Table 22. If some Port A pins are configured as outputs, it is essential that these do not switch when a conversion is in progress. This might corrupt the result of the conversion.

Table 22. Port A Pins Alternate Functions

Port Pin	Alternate Function
PA7	ADC7 (ADC input channel 7)
PA6	ADC6 (ADC input channel 6)
PA5	ADC5 (ADC input channel 5)
PA4	ADC4 (ADC input channel 4)
PA3	ADC3 (ADC input channel 3)
PA2	ADC2 (ADC input channel 2)
PA1	ADC1 (ADC input channel 1)
PA0	ADC0 (ADC input channel 0)

Table 23 and Table 24 relate the alternate functions of Port A to the overriding signals shown in Figure 26 on page 55.

Table 23. Overriding Signals for Alternate Functions in PA7..PA4

Signal Name	PA7/ADC7	PA6/ADC6	PA5/ADC5	PA4/ADC4
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	ADC7 INPUT	ADC6 INPUT	ADC5 INPUT	ADC4 INPUT

Table 24. Overriding Signals for Alternate Functions in PA3..PA0

Signal Name	PA3/ADC3	PA2/ADC2	PA1/ADC1	PA0/ADC0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 25.

Table 25. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	$\overline{SS}$ (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)

The alternate pin configuration is as follows:

- **SCK – Port B, Bit 7**

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB7. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB7. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB7 bit.

- **MISO – Port B, Bit 6**

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input regardless of the setting of DDB6. When the SPI is enabled as a Slave, the data direction of this pin is controlled by DDB6. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB6 bit.

- **MOSI – Port B, Bit 5**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB5. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB5 bit.

- **$\overline{SS}$ – Port B, Bit 4**

$\overline{SS}$: Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB4. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB4. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB4 bit.

- **AIN1/OC0 – Port B, Bit 3**

AIN1, Analog Comparator Negative Input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the analog comparator.

OC0, Output Compare Match output: The PB3 pin can serve as an external output for the Timer/Counter0 Compare Match. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC0 pin is also the output pin for the PWM mode timer function.

- **AIN0/INT2 – Port B, Bit 2**

AIN0, Analog Comparator Positive input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.

INT2, External Interrupt Source 2: The PB2 pin can serve as an external interrupt source to the MCU.

- **T1 – Port B, Bit 1**

T1, Timer/Counter1 Counter Source.

- **T0/XCK – Port B, Bit 0**

T0, Timer/Counter0 Counter Source.

XCK, USART External Clock. The Data Direction Register (DDB0) controls whether the clock is output (DDB0 set) or input (DDB0 cleared). The XCK pin is active only when the USART operates in Synchronous mode.

Table 26 and Table 27 relate the alternate functions of Port B to the overriding signals shown in Figure 26 on page 55. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 26. Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/SCK	PB6/MISO	PB5/MOSI	PB4/$\overline{SS}$
PUOE	SPE • $\overline{MSTR}$	SPE • MSTR	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$
PUOV	PORTB7 • $\overline{PUD}$	PORTB6 • $\overline{PUD}$	PORTB5 • $\overline{PUD}$	PORTB4 • $\overline{PUD}$
DDOE	SPE • $\overline{MSTR}$	SPE • MSTR	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$
DDOV	0	0	0	0
PVOE	SPE • MSTR	SPE • $\overline{MSTR}$	SPE • MSTR	0
PVOV	SCK OUTPUT	SPI SLAVE OUTPUT	SPI MSTR OUTPUT	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	SCK INPUT	SPI MSTR INPUT	SPI SLAVE INPUT	SPI $\overline{SS}$
AIO	–	–	–	–

Table 27. Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/OC0/AIN1	PB2/INT2/AIN0	PB1/T1	PB0/T0/XCK
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC0 ENABLE	0	0	UMSEL
PVOV	OC0	0	0	XCK OUTPUT
DIEOE	0	INT2 ENABLE	0	0
DIEOV	0	1	0	0
DI	–	INT2 INPUT	T1 INPUT	XCK INPUT/T0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	–	–

Alternate Functions of Port C

The Port C pins with alternate functions are shown in Table 28. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

Table 28. Port C Pins Alternate Functions

Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin 2)
PC6	TOSC1 (Timer Oscillator Pin 1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

The alternate pin configuration is as follows:

- **TOSC2 – Port C, Bit 7**

TOSC2, Timer Oscillator pin 2: When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PC7 is disconnected from the port, and becomes the inverting output of the Oscillator amplifier. In this mode, a Crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

- **TOSC1 – Port C, Bit 6**

TOSC1, Timer Oscillator pin 1: When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PC6 is disconnected from the port, and becomes the input of the inverting Oscillator amplifier. In this mode, a Crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

- **TDI – Port C, Bit 5**

TDI, JTAG Test Data In: Serial input data to be shifted in to the Instruction Register or Data Register (scan chains). When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **TDO – Port C, Bit 4**

TDO, JTAG Test Data Out: Serial output data from Instruction Register or Data Register. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

The TD0 pin is tri-stated unless TAP states that shifts out data are entered.

- **TMS – Port C, Bit 3**

TMS, JTAG Test Mode Select: This pin is used for navigating through the TAP-controller state machine. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **TCK – Port C, Bit 2**

TCK, JTAG Test Clock: JTAG operation is synchronous to TCK. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **SDA – Port C, Bit 1**

SDA, Two-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC1 is disconnected from the port and becomes the Serial Data I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation. When this pin is used by the Two-wire Serial Interface, the pull-up can still be controlled by the PORTC1 bit.

- **SCL – Port C, Bit 0**

SCL, Two-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC0 is disconnected from the port and becomes the Serial Clock I/O pin for the Two-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation. When this pin is used by the Two-wire Serial Interface, the pull-up can still be controlled by the PORTC0 bit.

Table 29 and Table 30 relate the alternate functions of Port C to the overriding signals shown in Figure 26 on page 55.

Table 29. Overriding Signals for Alternate Functions in PC7..PC4

Signal Name	PC7/TOSC2	PC6/TOSC1	PC5/TDI	PC4/TDO
PUOE	AS2	AS2	JTAGEN	JTAGEN
PUOV	0	0	1	0
DDOE	AS2	AS2	JTAGEN	JTAGEN
DDOV	0	0	0	SHIFT_IR + SHIFT_DR
PVOE	0	0	0	JTAGEN
PVOV	0	0	0	TDO
DIEOE	AS2	AS2	JTAGEN	JTAGEN
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	T/C2 OSC OUTPUT	T/C2 OSC INPUT	TDI	–

Table 30. Overriding Signals for Alternate Functions in PC3..PC0[1]

Signal Name	PC3/TMS	PC2/TCK	PC1/SDA	PC0/SCL
PUOE	JTAGEN	JTAGEN	TWEN	TWEN
PUOV	1	1	PORTC1 • $\overline{PUD}$	PORTC0 • $\overline{PUD}$
DDOE	JTAGEN	JTAGEN	TWEN	TWEN
DDOV	0	0	SDA_OUT	SCL_OUT
PVOE	0	0	TWEN	TWEN
PVOV	0	0	0	0
DIEOE	JTAGEN	JTAGEN	0	0
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	TMS	TCK	SDA INPUT	SCL INPUT

Note: 1. When enabled, the Two-wire Serial Interface enables slew-rate controls on the output pins PC0 and PC1. This is not shown in the figure. In addition, spike filters are connected between the AIO outputs shown in the port figure and the digital logic of the TWI module.

Alternate Functions of Port D

The Port D pins with alternate functions are shown in Table 31.

Table 31. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

The alternate pin configuration is as follows:

- **OC2 – Port D, Bit 7**

OC2, Timer/Counter2 Output Compare Match output: The PD7 pin can serve as an external output for the Timer/Counter2 Output Compare. The pin has to be configured as an output (DDD7 set (one)) to serve this function. The OC2 pin is also the output pin for the PWM mode timer function.

- **ICP1 – Port D, Bit 6**

ICP1 – Input Capture Pin: The PD6 pin can act as an Input Capture pin for Timer/Counter1.

- **OC1A – Port D, Bit 5**

OC1A, Output Compare Match A output: The PD5 pin can serve as an external output for the Timer/Counter1 Output Compare A. The pin has to be configured as an output (DDD5 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

- **OC1B – Port D, Bit 4**

OC1B, Output Compare Match B output: The PD4 pin can serve as an external output for the Timer/Counter1 Output Compare B. The pin has to be configured as an output (DDD4 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

- **INT1 – Port D, Bit 3**

INT1, External Interrupt Source 1: The PD3 pin can serve as an external interrupt source.

- **INT0 – Port D, Bit 2**

INT0, External Interrupt Source 0: The PD2 pin can serve as an external interrupt source.

- **TXD – Port D, Bit 1**

TXD, Transmit Data (Data output pin for the USART). When the USART Transmitter is enabled, this pin is configured as an output regardless of the value of DDD1.

- **RXD – Port D, Bit 0**

RXD, Receive Data (Data input pin for the USART). When the USART Receiver is enabled this pin is configured as an input regardless of the value of DDD0. When the USART forces this pin to be an input, the pull-up can still be controlled by the PORTD0 bit.

Table 32 and Table 33 relate the alternate functions of Port D to the overriding signals shown in Figure 26 on page 55.

Table 32. Overriding Signals for Alternate Functions PD7..PD4

Signal Name	PD7/OC2	PD6/ICP1	PD5/OC1A	PD4/OC1B
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC2 ENABLE	0	OC1A ENABLE	OC1B ENABLE
PVOV	OC2	0	OC1A	OC1B
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	–	ICP1 INPUT	–	–
AIO	–	–	–	–

Table 33. Overriding Signals for Alternate Functions in PD3..PD0

Signal Name	PD3/INT1	PD2/INT0	PD1/TXD	PD0/RXD
PUOE	0	0	TXEN	RXEN
PUOV	0	0	0	PORTD0 • $\overline{PUD}$
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	0	0	TXEN	0
PVOV	0	0	TXD	0
DIEOE	INT1 ENABLE	INT0 ENABLE	0	0
DIEOV	1	1	0	0
DI	INT1 INPUT	INT0 INPUT	–	RXD
AIO	–	–	–	–

Register Description for I/O Ports

Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Input Pins Address – PINA

Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

External Interrupts

The External Interrupts are triggered by the INT0, INT1, and INT2 pins. Observe that, if enabled, the interrupts will trigger even if the INT0..2 pins are configured as outputs. This feature provides a way of generating a software interrupt. The external interrupts can be triggered by a falling or rising edge or a low level (INT2 is only an edge triggered interrupt). This is set up as indicated in the specification for the MCU Control Register – MCUCR – and MCU Control and Status Register – MCUCSR. When the external interrupt is enabled and is configured as level triggered (only INT0/INT1), the interrupt will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 and INT1 requires the presence of an I/O clock, described in "Clock Systems and their Distribution" on page 24. Low level interrupts on INT0/INT1 and the edge interrupt on INT2 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except Idle mode.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the Watchdog Oscillator clock. The period of the Watchdog Oscillator is 1 µs (nominal) at 5.0V and 25°C. The frequency of the Watchdog Oscillator is voltage dependent as shown in "Electrical Characteristics" on page 291. The MCU will wake up if the input has the required level during this sampling or if it is held until the end of the start-up time. The start-up time is defined by the SUT Fuses as described in "System Clock and Clock Options" on page 24. If the level is sampled twice by the Watchdog Oscillator clock but disappears before the end of the start-up time, the MCU will still wake up, but no interrupt will be generated. The required level must be held long enough for the MCU to complete the wake up to trigger the level interrupt.

MCU Control Register – MCUCR

The MCU Control Register contains control bits for interrupt sense control and general MCU functions.

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-bit and the corresponding interrupt mask in the GICR are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 34. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 34. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- **Bit 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 35. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 35. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

MCU Control and Status Register – MCUCSR

Bit	7	6	5	4	3	2	1	0	
	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	See Bit Description					

- **Bit 6 – ISC2: Interrupt Sense Control 2**

The Asynchronous External Interrupt 2 is activated by the external pin INT2 if the SREG I-bit and the corresponding interrupt mask in GICR are set. If ISC2 is written to zero, a falling edge on INT2 activates the interrupt. If ISC2 is written to one, a rising edge on INT2 activates the interrupt. Edges on INT2 are registered asynchronously. Pulses on INT2 wider than the minimum pulse width given in Table 36 will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. When changing the ISC2 bit, an interrupt can occur. Therefore, it is recommended to first disable INT2 by clearing its Interrupt Enable bit in the GICR Register. Then, the ISC2 bit can be changed. Finally, the INT2 Interrupt Flag should be cleared by writing a logical one to its Interrupt Flag bit (INTF2) in the GIFR Register before the interrupt is re-enabled.

Table 36. Asynchronous External Interrupt Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
t_{INT}	Minimum pulse width for asynchronous external interrupt			50		ns

General Interrupt Control Register – GICR

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU General Control Register (MCUCR) define whether the External Interrupt is activated on rising

and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from the INT1 interrupt Vector.

- **Bit 6 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU General Control Register (MCUCR) define whether the External Interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from the INT0 interrupt vector.

- **Bit 5 – INT2: External Interrupt Request 2 Enable**

When the INT2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control2 bit (ISC2) in the MCU Control and Status Register (MCUCSR) defines whether the External Interrupt is activated on rising or falling edge of the INT2 pin. Activity on the pin will cause an interrupt request even if INT2 is configured as an output. The corresponding interrupt of External Interrupt Request 2 is executed from the INT2 Interrupt Vector.

General Interrupt Flag Register – GIFR

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	INTF2	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 6 – INTF0: External Interrupt Flag 0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GICR are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

- **Bit 5 – INTF2: External Interrupt Flag 2**

When an event on the INT2 pin triggers an interrupt request, INTF2 becomes set (one). If the I-bit in SREG and the INT2 bit in GICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. Note that when entering some sleep modes with the INT2 interrupt disabled, the input buffer on this pin will be disabled. This may cause a logic change in internal signals which will set the INTF2 Flag. See "Digital Input Enable and Sleep Modes" on page 54 for more information.

8-bit Timer/Counter0 with PWM

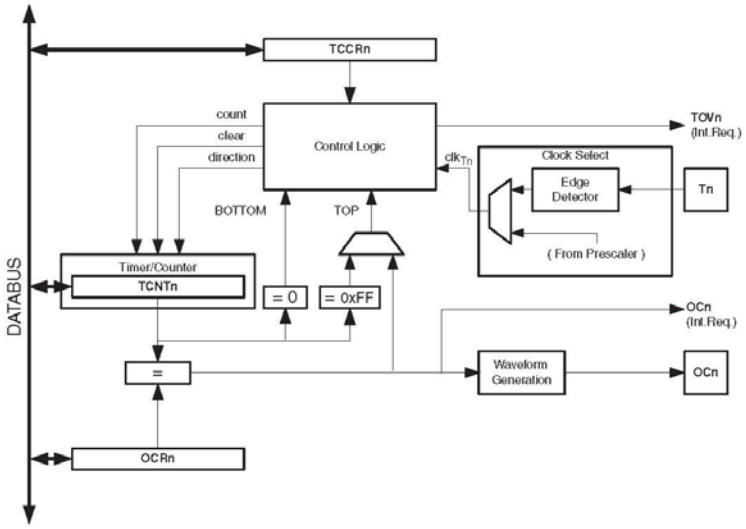
Timer/Counter0 is a general purpose, single compare unit, 8-bit Timer/Counter module. The main features are:

- **Single Compare Unit Counter**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch-free, Phase Correct Pulse Width Modulator (PWM)**
- **Frequency Generator**
- **External Event Counter**
- **10-bit Clock Prescaler**
- **Overflow and Compare Match Interrupt Sources (TOV0 and OCF0)**

Overview

A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 27. For the actual placement of I/O pins, refer to "Pinout ATmega16" on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "8-bit Timer/Counter Register Description" on page 83.

Figure 27. 8-bit Timer/Counter Block Diagram



Registers

The Timer/Counter (TCNT0) and Output Compare Register (OCR0) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in the figure) signals are all visible in the Timer Interrupt Flag Register (TIFR). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK). TIFR and TIMSK are not shown in the figure since these registers are shared by other timer units.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T0 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T0}).

The double buffered Output Compare Register (OCR0) is compared with the Timer/Counter value at all times. The result of the compare can be used by the waveform generator to generate a PWM or variable frequency output on the Output Compare Pin (OC0). See "Output Compare

Unit" on page 73. for details. The compare match event will also set the Compare Flag (OCF0) which can be used to generate an output compare interrupt request.

Definitions

Many register and bit references in this document are written in general form. A lower case "n" replaces the Timer/Counter number, in this case 0. However, when using the register or bit defines in a program, the precise form must be used, that is, TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions in Table 37 are also used extensively throughout the document.

Table 37. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0 Register. The assignment is dependent on the mode of operation.

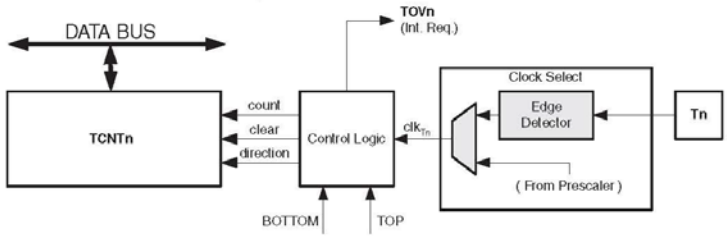
Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the clock select logic which is controlled by the clock select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0). For details on clock sources and prescaler, see "Timer/Counter0 and Timer/Counter1 Prescalers" on page 87.

Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 28 shows a block diagram of the counter and its surroundings.

Figure 28. Counter Unit Block Diagram



Signal description (internal signals):

count	Increment or decrement TCNT0 by 1.
direction	Select between increment and decrement.
clear	Clear TCNT0 (set all bits to zero).
clk_{Tn}	Timer/Counter clock, referred to as clk_{T0} in the following.
TOP	Signalize that TCNT0 has reached maximum value.
BOTTOM	Signalize that TCNT0 has reached minimum value (zero).

Depending of the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T0}). clk_{T0} can be generated from an external or internal clock source, selected by the Clock Select bits (CS02:0). When no clock source is selected (CS02:0 = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of

whether clk_{T0} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare output OC0. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 76.

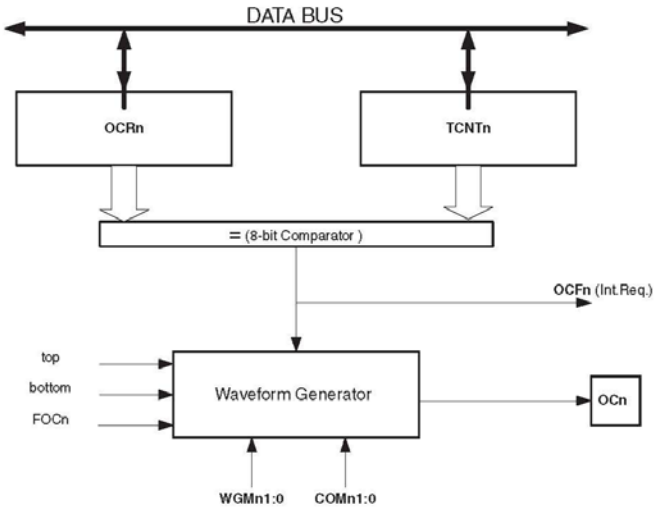
The Timer/Counter Overflow (TOV0) Flag is set according to the mode of operation selected by the WGM01:0 bits. TOV0 can be used for generating a CPU interrupt.

Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Register (OCR0). Whenever TCNT0 equals OCR0, the comparator signals a match. A match will set the Output Compare Flag (OCF0) at the next timer clock cycle. If enabled (OCIE0 = 1 and Global Interrupt Flag in SREG is set), the Output Compare Flag generates an output compare interrupt. The OCF0 Flag is automatically cleared when the interrupt is executed. Alternatively, the OCF0 Flag can be cleared by software by writing a logical one to its I/O bit location. The waveform generator uses the match signal to generate an output according to operating mode set by the WGM01:0 bits and Compare Output mode (COM01:0) bits. The max and bottom signals are used by the waveform generator for handling the special cases of the extreme values in some modes of operation (See "Modes of Operation" on page 76.).

Figure 29 shows a block diagram of the output compare unit.

Figure 29. Output Compare Unit, Block Diagram



The OCR0 Register is double buffered when using any of the Pulse Width Modulation (PWM) modes. For the normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR0 Compare Register to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR0 Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR0 Buffer Register, and if double buffering is disabled the CPU will access the OCR0 directly.

Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC0) bit. Forcing compare match will not set the OCF0 Flag or reload/clear the timer, but the OC0 pin will be updated as if a real compare match had occurred (the COM01:0 bits settings define whether the OC0 pin is set, cleared or toggled).

Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 Register will block any compare match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0 to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT0 when using the output compare unit, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0 value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is downcounting.

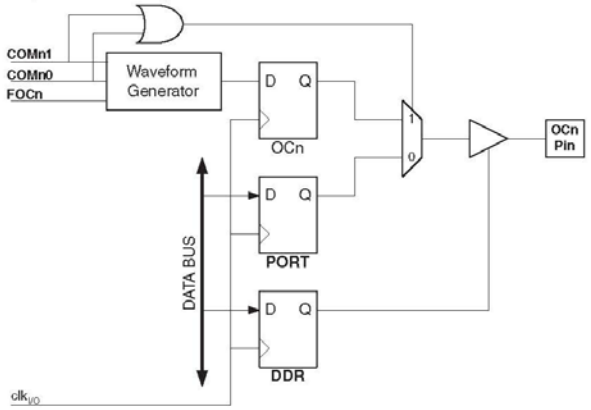
The setup of the OC0 should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC0 value is to use the Force Output Compare (FOC0) strobe bits in Normal mode. The OC0 Register keeps its value even when changing between waveform generation modes.

Be aware that the COM01:0 bits are not double buffered together with the compare value. Changing the COM01:0 bits will take effect immediately.

Compare Match Output Unit

The Compare Output mode (COM01:0) bits have two functions. The Waveform Generator uses the COM01:0 bits for defining the Output Compare (OC0) state at the next compare match. Also, the COM01:0 bits control the OC0 pin output source. Figure 30 shows a simplified schematic of the logic affected by the COM01:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port Control Registers (DDR and PORT) that are affected by the COM01:0 bits are shown. When referring to the OC0 state, the reference is for the internal OC0 Register, not the OC0 pin. If a System Reset occur, the OC0 Register is reset to "0".

Figure 30. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC0) from the Waveform Generator if either of the COM01:0 bits are set. However, the OC0 pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC0 pin (DDR_OC0) must be set as output before the OC0 value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the output compare pin logic allows initialization of the OC0 state before the output is enabled. Note that some COM01:0 bit settings are reserved for certain modes of operation. See "8-bit Timer/Counter Register Description" on page 83.

Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM01:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM01:0 = 0 tells the waveform generator that no action on the OC0 Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 39 on page 84. For fast PWM mode, refer to Table 40 on page 84, and for phase correct PWM refer to Table 41 on page 84.

A change of the COM01:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC0 strobe bits.

Modes of Operation

The mode of operation, that is, the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM01:0) and Compare Output mode (COM01:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM01:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM01:0 bits control whether the output should be set, cleared, or toggled at a compare match (See "Compare Match Output Unit" on page 74.).

For detailed timing information refer to Figure 34, Figure 35, Figure 36 and Figure 37 in "Timer/Counter Timing Diagrams" on page 81.

Normal Mode

The simplest mode of operation is the normal mode (WGM01:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (TOP = 0xFF) and then restarts from the bottom (0x00). In normal operation the Timer/Counter Overflow Flag (TOV0) will be set in the same timer clock cycle as the TCNT0 becomes zero. The TOV0 Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV0 Flag, the timer resolution can be increased by software. There are no special cases to consider in the normal mode, a new counter value can be written anytime.

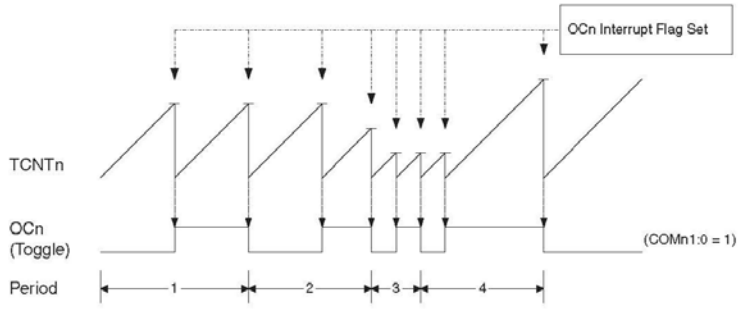
The output compare unit can be used to generate interrupts at some given time. Using the output compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM01:0 = 2), the OCR0 Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT0) matches the OCR0. The OCR0 defines the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in Figure 31. The counter value (TCNT0) increases until a compare match occurs between TCNT0 and OCR0, and then counter (TCNT0) is cleared.

Figure 31. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0 Flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have

the double buffering feature. If the new value written to OCR0 is lower than the current value of TCNT0, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the compare match can occur.

For generating a waveform output in CTC mode, the OC0 output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM01:0 = 1). The OC0 value will not be visible on the port pin unless the data direction for the pin is set to output. The waveform generated will have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0 is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRn)}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

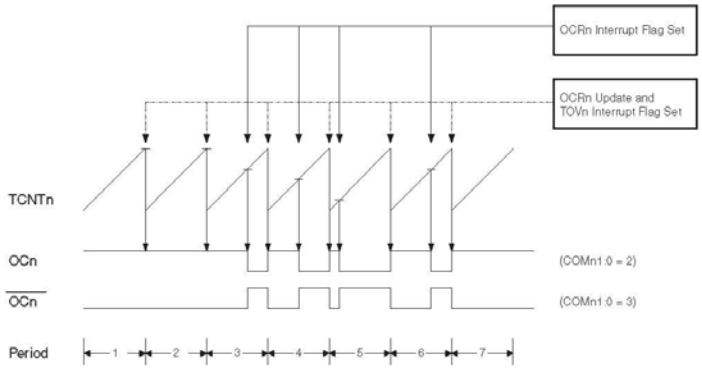
As for the Normal mode of operation, the TOV0 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM01:0 = 3) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to MAX then restarts from BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC0) is cleared on the compare match between TCNT0 and OCR0, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the MAX value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 32. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0 and TCNT0.

Figure 32. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches MAX. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC0 pin. Setting the COM01:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM01:0 to 3 (See Table 40 on page 84). The actual OC0 value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC0 Register at the compare match between OCR0 and TCNT0, and clearing (or setting) the OC0 Register at the timer clock cycle the counter is cleared (changes from MAX to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

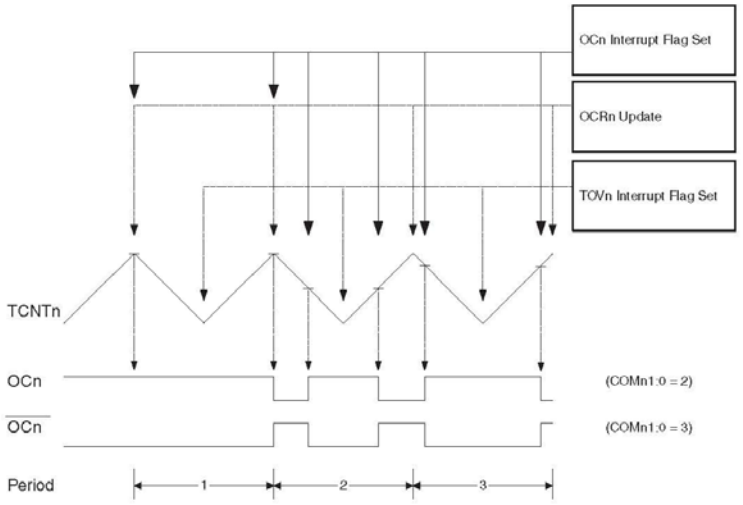
The extreme values for the OCR0 Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0 is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR0 equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM01:0 bits.)

Phase Correct PWM Mode

The phase correct PWM mode (WGM01:0 = 1) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to MAX and then from MAX to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC0) is cleared on the compare match between TCNT0 and OCR0 while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The PWM resolution for the phase correct PWM mode is fixed to eight bits. In phase correct PWM mode the counter is incremented until the counter value matches MAX. When the counter reaches MAX, it changes the count direction. The TCNT0 value will be equal to MAX for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 33. The TCNT0 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent compare matches between OCR0 and TCNT0.

Figure 33. Phase Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches BOTTOM. The Interrupt Flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

In phase correct PWM mode, the compare unit allows generation of PWM waveforms on the OC0 pin. Setting the COM01:0 bits to 2 will produce a non-inverted PWM. An inverted PWM output can be generated by setting the COM01:0 to 3 (see Table 41 on page 84). The actual OC0 value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by clearing (or setting) the OC0 Register at the compare match between OCR0 and TCNT0 when the counter increments, and setting (or clearing) the OC0 Register at compare match between OCR0 and TCNT0 when the counter decrements. The

PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0 Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR0 is set equal to BOTTOM, the output will be continuously low and if set equal to MAX the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

At the very start of Period 2 in Figure 33 OCn has a transition from high to low even though there is no Compare Match. The point of this transition is to guarantee symmetry around BOTTOM. There are two cases that give a transition without Compare Match:

- OCR0A changes its value from MAX, like in Figure 33. When the OCR0A value is MAX the OCn pin value is the same as the result of a down-counting Compare Match. To ensure symmetry around BOTTOM the OCn value at MAX must be correspond to the result of an up-counting Compare Match.
- The Timer starts counting from a value higher than the one in OCR0A, and for that reason misses the Compare Match and hence the OCn change that would have happened on the way up.

Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T0}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt Flags are set. Figure 34 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 34. Timer/Counter Timing Diagram, no Prescaling

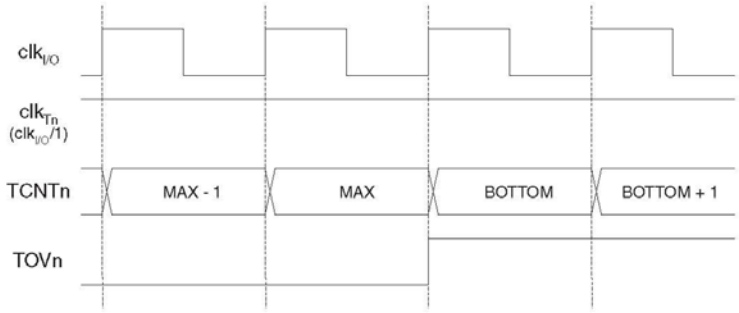


Figure 35 shows the same timing data, but with the prescaler enabled.

Figure 35. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

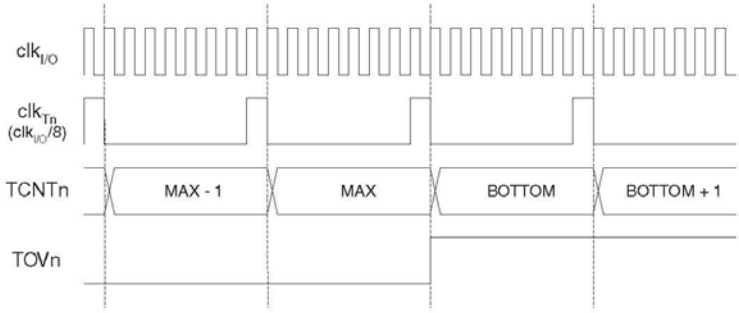


Figure 36 shows the setting of OCF0 in all modes except CTC mode.

Figure 36. Timer/Counter Timing Diagram, Setting of OCF0, with Prescaler ($f_{clk\_I/O}/8$)

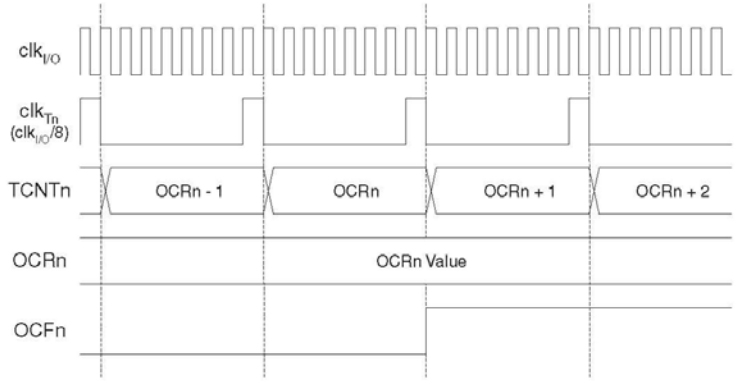
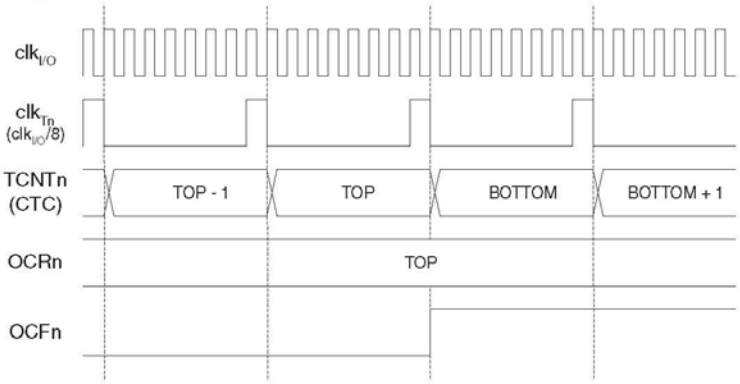


Figure 37 shows the setting of OCF0 and the clearing of TCNT0 in CTC mode.

Figure 37. Timer/Counter Timing Diagram, Clear Timer on Compare Match Mode, with Prescaler ($f_{clk\_I/O}/8$)



8-bit Timer/Counter Register Description

Timer/Counter Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0: Force Output Compare**

The FOC0 bit is only active when the WGM00 bit specifies a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0 is written when operating in PWM mode. When writing a logical one to the FOC0 bit, an immediate compare match is forced on the Waveform Generation unit. The OC0 output is changed according to its COM01:0 bits setting. Note that the FOC0 bit is implemented as a strobe. Therefore it is the value present in the COM01:0 bits that determines the effect of the forced compare.

A FOC0 strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0 as TOP.

The FOC0 bit is always read as zero.

- **Bit 3, 6 – WGM01:0: Waveform Generation Mode**

These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of Waveform Generation to be used. Modes of operation supported by the Timer/Counter unit are: Normal mode, Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes. See Table 38 and "Modes of Operation" on page 76.

Table 38. Waveform Generation Mode Bit Description[1]

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Note: 1. The CTC0 and PWM0 bit definition names are now obsolete. Use the WGM01:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

- **Bit 5:4 – COM01:0: Compare Match Output Mode**

These bits control the Output Compare pin (OC0) behavior. If one or both of the COM01:0 bits are set, the OC0 output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0 pin must be set in order to enable the output driver.

When OC0 is connected to the pin, the function of the COM01:0 bits depends on the WGM01:0 bit setting. Table 39 shows the COM01:0 bit functionality when the WGM01:0 bits are set to a normal or CTC mode (non-PWM).

Table 39. Compare Output Mode, non-PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Toggle OC0 on compare match
1	0	Clear OC0 on compare match
1	1	Set OC0 on compare match

Table 40 shows the COM01:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 40. Compare Output Mode, Fast PWM Mode(1)

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)

Note: 1. A special case occurs when OCR0 equals TOP and COM01 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See "Fast PWM Mode" on page 77 for more details.

Table 41 shows the COM01:0 bit functionality when the WGM01:0 bits are set to phase correct PWM mode.

Table 41. Compare Output Mode, Phase Correct PWM Mode(1)

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match when up-counting. Set OC0 on compare match when downcounting.
1	1	Set OC0 on compare match when up-counting. Clear OC0 on compare match when downcounting.

Note: 1. A special case occurs when OCR0 equals TOP and COM01 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 79 for more details.

- **Bit 2:0 – CS02:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter.

Table 42. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

Timer/Counter Register – TCNT0

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 Register blocks (removes) the compare match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a compare match between TCNT0 and the OCR0 Register.

Output Compare Register – OCR0

Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0 pin.

Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – OCIE0: Timer/Counter0 Output Compare Match Interrupt Enable**

When the OCIE0 bit is written to one, and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter0 occurs, that is, when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, that is, when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – OCF0: Output Compare Flag 0**

The OCF0 bit is set (one) when a compare match occurs between the Timer/Counter0 and the data in OCR0 – Output Compare Register0. OCF0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0 is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0 (Timer/Counter0 Compare Match Interrupt Enable), and OCF0 are set (one), the Timer/Counter0 Compare Match Interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed. In phase correct PWM mode, this bit is set when Timer/Counter0 changes counting direction at $00.

Timer/Counter0 and Timer/Counter1 Prescalers

Timer/Counter1 and Timer/Counter0 share the same prescaler module, but the Timer/Counters can have different prescaler settings. The description below applies to both Timer/Counter1 and Timer/Counter0.

Internal Clock Source

The Timer/Counter can be clocked directly by the system clock (by setting the CSn2:0 = 1). This provides the fastest operation, with a maximum Timer/Counter clock frequency equal to system clock frequency ($f_{CLK\_I/O}$). Alternatively, one of four taps from the prescaler can be used as a clock source. The prescaled clock has a frequency of either $f_{CLK\_I/O}/8$, $f_{CLK\_I/O}/64$, $f_{CLK\_I/O}/256$, or $f_{CLK\_I/O}/1024$.

Prescaler Reset

The prescaler is free running, that is, operates independently of the clock select logic of the Timer/Counter, and it is shared by Timer/Counter1 and Timer/Counter0. Since the prescaler is not affected by the Timer/Counter's clock select, the state of the prescaler will have implications for situations where a prescaled clock is used. One example of prescaling artifacts occurs when the timer is enabled and clocked by the prescaler (6 > CSn2:0 > 1). The number of system clock cycles from when the timer is enabled to the first count occurs can be from 1 to N+1 system clock cycles, where N equals the prescaler divisor (8, 64, 256, or 1024).

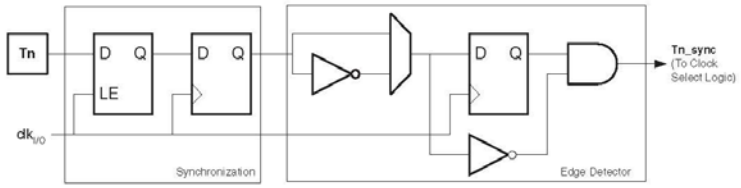
It is possible to use the Prescaler Reset for synchronizing the Timer/Counter to program execution. However, care must be taken if the other Timer/Counter that shares the same prescaler also uses prescaling. A prescaler reset will affect the prescaler period for all Timer/Counters it is connected to.

External Clock Source

An external clock source applied to the T1/T0 pin can be used as Timer/Counter clock (clk_{T1}/clk_{T0}). The T1/T0 pin is sampled once every system clock cycle by the pin synchronization logic. The synchronized (sampled) signal is then passed through the edge detector. Figure 38 shows a functional equivalent block diagram of the T1/T0 synchronization and edge detector logic. The registers are clocked at the positive edge of the internal system clock ($clk_{I/O}$). The latch is transparent in the high period of the internal system clock.

The edge detector generates one clk_{T1}/clk_{T0} pulse for each positive (CSn2:0 = 7) or negative (CSn2:0 = 6) edge it detects.

Figure 38. T1/T0 Pin Sampling



The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1/T0 pin to the counter is updated.

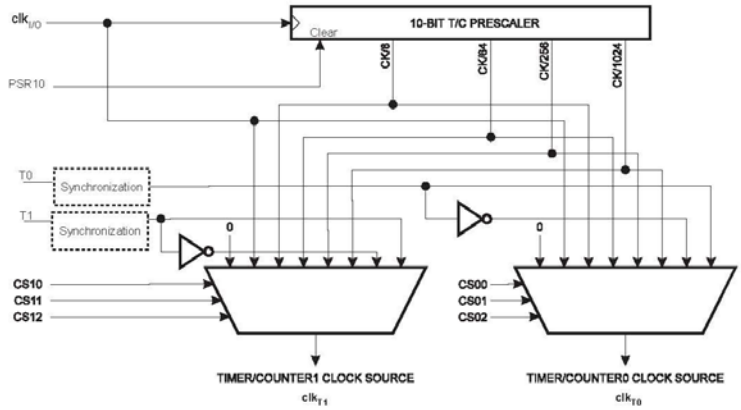
Enabling and disabling of the clock input must be done when T1/T0 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

Each half period of the external clock applied must be longer than one system clock cycle to ensure correct sampling. The external clock must be guaranteed to have less than half the system clock frequency ($f_{ExtClk} < f_{clk\_I/O}/2$) given a 50/50% duty cycle. Since the edge detector uses

sampling, the maximum frequency of an external clock it can detect is half the sampling frequency (Nyquist sampling theorem). However, due to variation of the system clock frequency and duty cycle caused by Oscillator source (crystal, resonator, and capacitors) tolerances, it is recommended that maximum frequency of an external clock source is less than $f_{clk\_I/O}/2.5$.

An external clock source can not be prescaled.

Figure 39. Prescaler for Timer/Counter0 and Timer/Counter1[1]



Note: 1. The synchronization logic on the input pins (T1/T0) is shown in Figure 38.

Special Function IO Register – SFIOR

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 0 – PSR10: Prescaler Reset Timer/Counter1 and Timer/Counter0**

When this bit is written to one, the Timer/Counter1 and Timer/Counter0 prescaler will be reset. The bit will be cleared by hardware after the operation is performed. Writing a zero to this bit will have no effect. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a reset of this prescaler will affect both timers. This bit will always be read as zero.

16-bit Timer/Counter1

The 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement. The main features are:

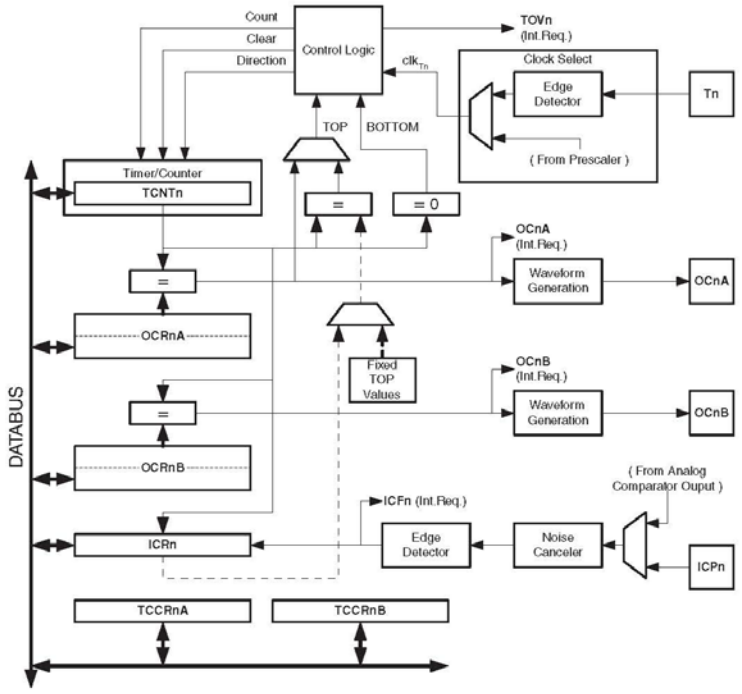
- **True 16-bit Design (that is, allows 16-bit PWM)**
- **Two Independent Output Compare Units**
- **Double Buffered Output Compare Registers**
- **One Input Capture Unit**
- **Input Capture Noise Canceler**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch-free, Phase Correct Pulse Width Modulator (PWM)**
- **Variable PWM Period**
- **Frequency Generator**
- **External Event Counter**
- **Four Independent Interrupt Sources (TOV1, OCF1A, OCF1B, and ICF1)**

Overview

Most register and bit references in this section are written in general form. A lower case "n" replaces the Timer/Counter number, and a lower case "x" replaces the output compare unit. However, when using the register or bit defines in a program, the precise form must be used (that is, TCNT1 for accessing Timer/Counter1 counter value and so on).

A simplified block diagram of the 16-bit Timer/Counter is shown in Figure 40. For the actual placement of I/O pins, refer to Figure 1 on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device specific I/O Register and bit locations are listed in the "16-bit Timer/Counter Register Description" on page 110.

Figure 40. 16-bit Timer/Counter Block Diagram[1]



Note: 1. Refer to Figure 1 on page 2, Table 25 on page 58, and Table 31 on page 63 for Timer/Counter1 pin placement and description.

Registers

The *Timer/Counter* (TCNT1), *Output Compare Registers* (OCR1A/B), and *Input Capture Register* (ICR1) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in the section "Accessing 16-bit Registers" on page 92. The *Timer/Counter Control Registers* (TCCR1A/B) are 8-bit registers and have no CPU access restrictions. Interrupt requests (abbreviated to Int.Req. in the figure) signals are all visible in the *Timer Interrupt Flag Register* (TIFR). All interrupts are individually masked with the *Timer Interrupt Mask Register* (TIMSK). TIFR and TIMSK are not shown in the figure since these registers are shared by other timer units.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T1 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the clock select logic is referred to as the timer clock (clk_{T1}).

The double buffered Output Compare Registers (OCR1A/B) are compared with the Timer/Counter value at all time. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pin (OC1A/B). See "Output Compare Units" on page 98. The compare match event will also set the Compare Match Flag (OCF1A/B) which can be used to generate an output compare interrupt request.

The Input Capture Register can capture the Timer/Counter value at a given external (edge triggered) event on either the Input Capture Pin (ICP1) or on the Analog Comparator pins (See "Analog Comparator" on page 201.) The Input Capture unit includes a digital filtering unit (Noise Canceler) for reducing the chance of capturing noise spikes.

The TOP value, or maximum Timer/Counter value, can in some modes of operation be defined by either the OCR1A Register, the ICR1 Register, or by a set of fixed values. When using OCR1A as TOP value in a PWM mode, the OCR1A Register can not be used for generating a PWM output. However, the TOP value will in this case be double buffered allowing the TOP value to be changed in run time. If a fixed TOP value is required, the ICR1 Register can be used as an alternative, freeing the OCR1A to be used as PWM output.

Definitions

The following definitions are used extensively throughout the document:

Table 43. Definitions

BOTTOM	The counter reaches the *BOTTOM* when it becomes 0x0000.
MAX	The counter reaches its *MAX*imum when it becomes 0xFFFF (decimal 65535).
TOP	The counter reaches the *TOP* when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be one of the fixed values: 0x00FF, 0x01FF, or 0x03FF, or to the value stored in the OCR1A or ICR1 Register. The assignment is dependent of the mode of operation.

Compatibility

The 16-bit Timer/Counter has been updated and improved from previous versions of the 16-bit AVR Timer/Counter. This 16-bit Timer/Counter is fully compatible with the earlier version regarding:

- All 16-bit Timer/Counter related I/O Register address locations, including Timer Interrupt Registers.
- Bit locations inside all 16-bit Timer/Counter Registers, including Timer Interrupt Registers.
- Interrupt Vectors.

The following control bits have changed name, but have same functionality and register location:

- PWM10 is changed to WGM10.
- PWM11 is changed to WGM11.
- CTC1 is changed to WGM12.

The following bits are added to the 16-bit Timer/Counter Control Registers:

- FOC1A and FOC1B are added to TCCR1A.
- WGM13 is added to TCCR1B.

The 16-bit Timer/Counter has improvements that will affect the compatibility in some special cases.

Accessing 16-bit Registers

The TCNT1, OCR1A/B, and ICR1 are 16-bit registers that can be accessed by the AVR CPU via the 8-bit data bus. The 16-bit register must be byte accessed using two read or write operations. Each 16-bit timer has a single 8-bit register for temporary storing of the High byte of the 16-bit access. The same temporary register is shared between all 16-bit registers within each 16-bit timer. Accessing the Low byte triggers the 16-bit read or write operation. When the Low byte of a 16-bit register is written by the CPU, the High byte stored in the temporary register, and the Low byte written are both copied into the 16-bit register in the same clock cycle. When the Low byte of a 16-bit register is read by the CPU, the High byte of the 16-bit register is copied into the temporary register in the same clock cycle as the Low byte is read.

Not all 16-bit accesses uses the temporary register for the High byte. Reading the OCR1A/B 16-bit registers does not involve using the temporary register.

To do a 16-bit write, *the High byte must be written before the Low byte*. For a 16-bit read, *the Low byte must be read before the High byte*.

The following code examples show how to access the 16-bit Timer Registers assuming that no interrupts updates the temporary register. The same principle can be used directly for accessing the OCR1A/B and ICR1 Registers. Note that when using "C", the compiler handles the 16-bit access.

Assembly Code Example[1]

```
...
; Set TCNT1 to 0x01FF
ldi r17,0x01
ldi r16,0xFF
out TCNT1H,r17
out TCNT1L,r16
; Read TCNT1 into r17:r16
in  r16,TCNT1L
in  r17,TCNT1H
...
```

C Code Example[1]

```
unsigned int i;
...
/* Set TCNT1 to 0x01FF */
TCNT1 = 0x1FF;
/* Read TCNT1 into i */
i = TCNT1;
...
```

Note: 1. See "About Code Examples" on page 7.

The assembly code example returns the TCNT1 value in the r17:r16 register pair.

It is important to notice that accessing 16-bit registers are atomic operations. If an interrupt occurs between the two instructions accessing the 16-bit register, and the interrupt code updates the temporary register by accessing the same or any other of the 16-bit Timer Registers, then the result of the access outside the interrupt will be corrupted. Therefore, when both the main code and the interrupt code update the temporary register, the main code must disable the interrupts during the 16-bit access.

The following code examples show how to do an atomic read of the TCNT1 Register contents. Reading any of the OCR1A/B or ICR1 Registers can be done by using the same principle.

Assembly Code Example[1]

```
TIM16_ReadTCNT1:
  ; Save global interrupt flag
  in  r18,SREG
  ; Disable interrupts
  cli
  ; Read TCNT1 into r17:r16
  in  r16,TCNT1L
  in  r17,TCNT1H
  ; Restore global interrupt flag
  out SREG,r18
  ret
```

C Code Example[1]

```
unsigned int TIM16_ReadTCNT1( void )
{
  unsigned char sreg;
  unsigned int i;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  _CLI();
  /* Read TCNT1 into i */
  i = TCNT1;
  /* Restore global interrupt flag */
  SREG = sreg;
  return i;
}
```

Note: 1. See "About Code Examples" on page 7.

The assembly code example returns the TCNT1 value in the r17:r16 register pair.

The following code examples show how to do an atomic write of the TCNT1 Register contents. Writing any of the OCR1A/B or ICR1 Registers can be done by using the same principle.

Assembly Code Example[1]

```
TIM16_WriteTCNT1:
  ; Save global interrupt flag
  in  r18,SREG
  ; Disable interrupts
  cli
  ; Set TCNT1 to r17:r16
  out TCNT1H,r17
  out TCNT1L,r16
  ; Restore global interrupt flag
  out SREG,r18
  ret
```

C Code Example[1]

```
void TIM16_WriteTCNT1 ( unsigned int i )
{
  unsigned char sreg;
  unsigned int i;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  _CLI();
  /* Set TCNT1 to i */
  TCNT1 = i;
  /* Restore global interrupt flag */
  SREG = sreg;
}
```

Note: 1. See "About Code Examples" on page 7.

The assembly code example requires that the r17:r16 register pair contains the value to be written to TCNT1.

Reusing the Temporary High Byte Register

If writing to more than one 16-bit register where the High byte is the same for all registers written, then the High byte only needs to be written once. However, note that the same rule of atomic operation described previously also applies in this case.

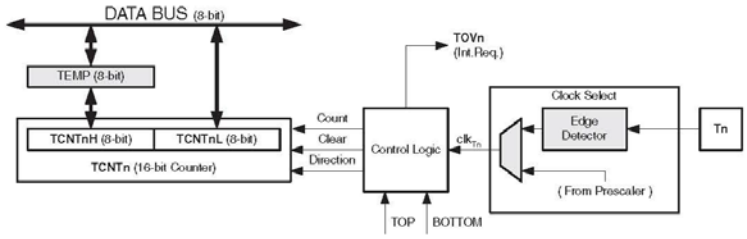
Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the *Clock Select* (CS12:0) bits located in the *Timer/Counter Control Register B* (TCCR1B). For details on clock sources and prescaler, see "Timer/Counter0 and Timer/Counter1 Prescalers" on page 87.

Counter Unit

The main part of the 16-bit Timer/Counter is the programmable 16-bit bi-directional counter unit. Figure 41 shows a block diagram of the counter and its surroundings.

Figure 41. Counter Unit Block Diagram



Signal description (internal signals):

Count	Increment or decrement TCNT1 by 1.
Direction	Select between increment and decrement.
Clear	Clear TCNT1 (set all bits to zero).
clk_{T1}	Timer/Counter clock.
TOP	Signalize that TCNT1 has reached maximum value.
BOTTOM	Signalize that TCNT1 has reached minimum value (zero).

The 16-bit counter is mapped into two 8-bit I/O memory locations: *Counter High* (TCNT1H) containing the upper eight bits of the counter, and *Counter Low* (TCNT1L) containing the lower 8 bits. The TCNT1H Register can only be indirectly accessed by the CPU. When the CPU does an access to the TCNT1H I/O location, the CPU accesses the High byte temporary register (TEMP). The temporary register is updated with the TCNT1H value when the TCNT1L is read, and TCNT1H is updated with the temporary register value when TCNT1L is written. This allows the CPU to read or write the entire 16-bit counter value within one clock cycle via the 8-bit data bus. It is important to notice that there are special cases of writing to the TCNT1 Register when the counter is counting that will give unpredictable results. The special cases are described in the sections where they are of importance.

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each *timer clock* (clk_{T1}). The clk_{T1} can be generated from an external or internal clock source, selected by the *Clock Select* bits (CS12:0). When no clock source is selected (CS12:0 = 0) the timer is stopped. However, the TCNT1 value can be accessed by the CPU, independent of whether clk_{T1} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the *Waveform Generation Mode* bits (WGM13:0) located in the *Timer/Counter Control Registers* A and B (TCCR1A and TCCR1B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC1x. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 101.

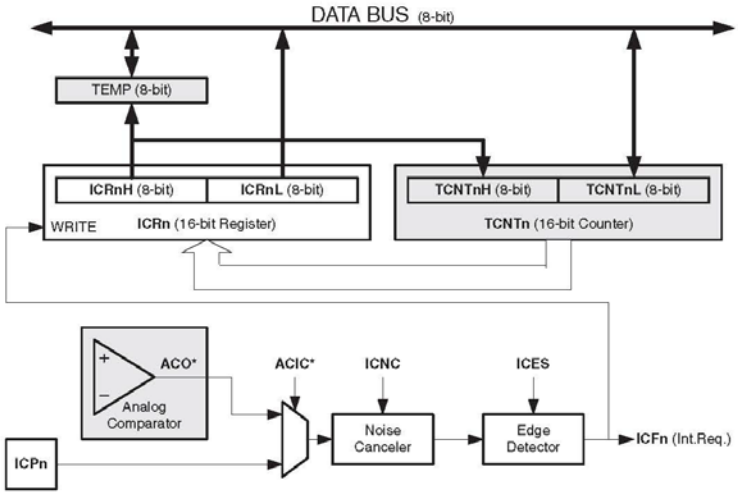
The *Timer/Counter Overflow* (TOV1) Flag is set according to the mode of operation selected by the WGM13:0 bits. TOV1 can be used for generating a CPU interrupt.

Input Capture Unit

The Timer/Counter incorporates an Input Capture unit that can capture external events and give them a time-stamp indicating time of occurrence. The external signal indicating an event, or multiple events, can be applied via the ICP1 pin or alternatively, via the Analog Comparator unit. The time-stamps can then be used to calculate frequency, duty-cycle, and other features of the signal applied. Alternatively the time-stamps can be used for creating a log of the events.

The Input Capture unit is illustrated by the block diagram shown in Figure 42. The elements of the block diagram that are not directly a part of the Input Capture unit are gray shaded. The small "n" in register and bit names indicates the Timer/Counter number.

Figure 42. Input Capture Unit Block Diagram



When a change of the logic level (an event) occurs on the *Input Capture pin* (ICP1), alternatively on the *Analog Comparator output* (ACO), and this change confirms to the setting of the edge detector, a capture will be triggered. When a capture is triggered, the 16-bit value of the counter (TCNT1) is written to the *Input Capture Register* (ICR1). The *Input Capture Flag* (ICF1) is set at the same system clock as the TCNT1 value is copied into ICR1 Register. If enabled (TICIE1 = 1), the Input Capture Flag generates an Input Capture Interrupt. The ICF1 Flag is automatically cleared when the interrupt is executed. Alternatively the ICF1 Flag can be cleared by software by writing a logical one to its I/O bit location.

Reading the 16-bit value in the *Input Capture Register* (ICR1) is done by first reading the Low byte (ICR1L) and then the High byte (ICR1H). When the Low byte is read the High byte is copied into the High byte temporary register (TEMP). When the CPU reads the ICR1H I/O location it will access the TEMP Register.

The ICR1 Register can only be written when using a Waveform Generation mode that utilizes the ICR1 Register for defining the counter's TOP value. In these cases the *Waveform Generation mode* (WGM13:0) bits must be set before the TOP value can be written to the ICR1 Register. When writing the ICR1 Register the High byte must be written to the ICR1H I/O location before the Low byte is written to ICR1L.

For more information on how to access the 16-bit registers refer to "Accessing 16-bit Registers" on page 92.

Input Capture Pin Source

The main trigger source for the Input Capture unit is the *Input Capture pin* (ICP1). Timer/Counter1 can alternatively use the Analog Comparator output as trigger source for the Input Capture unit. The Analog Comparator is selected as trigger source by setting the *Analog Comparator Input Capture* (ACIC) bit in the *Analog Comparator Control and Status Register* (ACSR). Be aware that changing trigger source can trigger a capture. The Input Capture Flag must therefore be cleared after the change.

Both the *Input Capture pin* (ICP1) and the *Analog Comparator output* (ACO) inputs are sampled using the same technique as for the T1 pin (Figure 38 on page 87). The edge detector is also identical. However, when the noise canceler is enabled, additional logic is inserted before the edge detector, which increases the delay by four system clock cycles. Note that the input of the noise canceler and edge detector is always enabled unless the Timer/Counter is set in a waveform generation mode that uses ICR1 to define TOP.

An Input Capture can be triggered by software by controlling the port of the ICP1 pin.

Noise Canceler

The noise canceler improves noise immunity by using a simple digital filtering scheme. The noise canceler input is monitored over four samples, and all four must be equal for changing the output that in turn is used by the edge detector.

The noise canceler is enabled by setting the *Input Capture Noise Canceler* (ICNC1) bit in *Timer/Counter Control Register B* (TCCR1B). When enabled the noise canceler introduces additional four system clock cycles of delay from a change applied to the input, to the update of the ICR1 Register. The noise canceler uses the system clock and is therefore not affected by the prescaler.

Using the Input Capture Unit

The main challenge when using the Input Capture unit is to assign enough processor capacity for handling the incoming events. The time between two events is critical. If the processor has not read the captured value in the ICR1 Register before the next event occurs, the ICR1 will be overwritten with a new value. In this case the result of the capture will be incorrect.

When using the Input Capture Interrupt, the ICR1 Register should be read as early in the interrupt handler routine as possible. Even though the Input Capture Interrupt has relatively high priority, the maximum interrupt response time is dependent on the maximum number of clock cycles it takes to handle any of the other interrupt requests.

Using the Input Capture unit in any mode of operation when the TOP value (resolution) is actively changed during operation, is not recommended.

Measurement of an external signal's duty cycle requires that the trigger edge is changed after each capture. Changing the edge sensing must be done as early as possible after the ICR1 Register has been read. After a change of the edge, the Input Capture Flag (ICF1) must be cleared by software (writing a logical one to the I/O bit location). For measuring frequency only, the clearing of the ICF1 Flag is not required (if an interrupt handler is used).

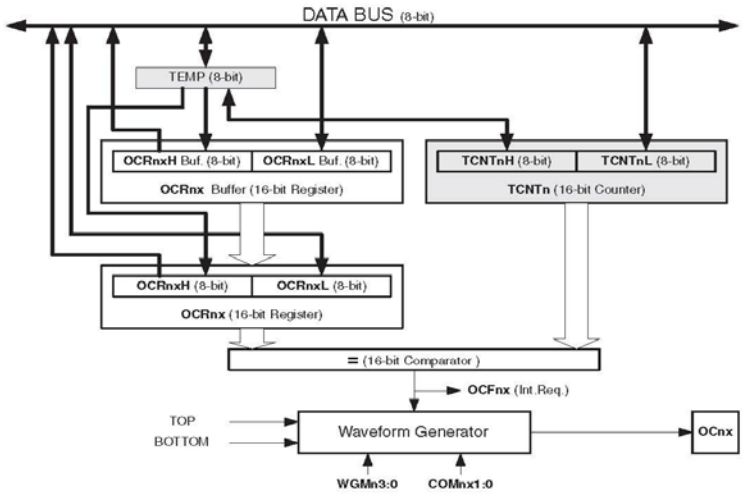
Output Compare Units

The 16-bit comparator continuously compares TCNT1 with the *Output Compare Register* (OCR1x). If TCNT equals OCR1x the comparator signals a match. A match will set the *Output Compare Flag* (OCF1x) at the next timer clock cycle. If enabled (OCIE1x = 1), the Output Compare Flag generates an output compare interrupt. The OCF1x Flag is automatically cleared when the interrupt is executed. Alternatively the OCF1x Flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the *Waveform Generation mode* (WGM13:0) bits and *Compare Output mode* (COM1x1:0) bits. The TOP and BOTTOM signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (See "Modes of Operation" on page 101.)

A special feature of output compare unit A allows it to define the Timer/Counter TOP value (that is, counter resolution). In addition to the counter resolution, the TOP value defines the period time for waveforms generated by the Waveform Generator.

Figure 43 shows a block diagram of the output compare unit. The small "n" in the register and bit names indicates the device number (n = 1 for Timer/Counter1), and the "x" indicates output compare unit (A/B). The elements of the block diagram that are not directly a part of the output compare unit are gray shaded.

Figure 43. Output Compare Unit, Block Diagram



The OCR1x Register is double buffered when using any of the twelve *Pulse Width Modulation* (PWM) modes. For the normal and *Clear Timer on Compare* (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR1x Compare Register to either TOP or BOTTOM of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR1x Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR1x Buffer Register, and if double buffering is disabled the CPU will access the OCR1x directly. The content of the OCR1x (Buffer or Compare) Register is only changed by a write operation (the Timer/Counter does not update this register automatically as the TCNT1 and ICR1 Register). Therefore OCR1x is not read via the High byte

temporary register (TEMP). However, it is a good practice to read the Low byte first as when accessing other 16-bit registers. Writing the OCR1x Registers must be done via the TEMP Register since the compare of all 16 bits is done continuously. The High byte (OCR1xH) has to be written first. When the High byte I/O location is written by the CPU, the TEMP Register will be updated by the value written. Then when the Low byte (OCR1xL) is written to the lower eight bits, the High byte will be copied into the upper 8-bits of either the OCR1x buffer or OCR1x Compare Register in the same system clock cycle.

For more information of how to access the 16-bit registers refer to "Accessing 16-bit Registers" on page 92.

Force Output Compare

In non-PWM Waveform Generation modes, the match output of the comparator can be forced by writing a one to the *Force Output Compare* (FOC1x) bit. Forcing compare match will not set the OCF1x Flag or reload/clear the timer, but the OC1x pin will be updated as if a real compare match had occurred (the COM1x1:0 bits settings define whether the OC1x pin is set, cleared or toggled).

Compare Match Blocking by TCNT1 Write

All CPU writes to the TCNT1 Register will block any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows OCR1x to be initialized to the same value as TCNT1 without triggering an interrupt when the Timer/Counter clock is enabled.

Using the Output Compare Unit

Since writing TCNT1 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT1 when using any of the output compare units, independent of whether the Timer/Counter is running or not. If the value written to TCNT1 equals the OCR1x value, the compare match will be missed, resulting in incorrect waveform generation. Do not write the TCNT1 equal to TOP in PWM modes with variable TOP values. The compare match for the TOP will be ignored and the counter will continue to 0xFFFF. Similarly, do not write the TCNT1 value equal to BOTTOM when the counter is downcounting.

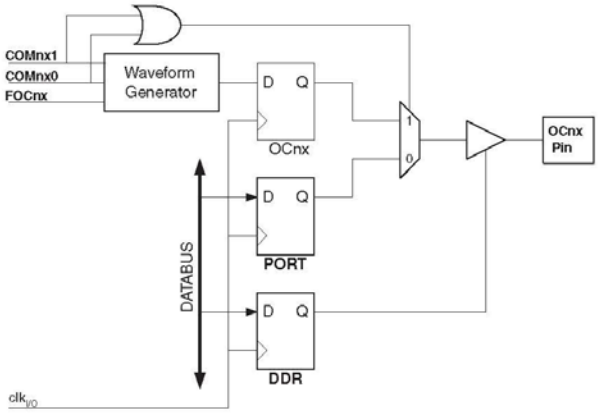
The setup of the OC1x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC1x value is to use the force output compare (FOC1x) strobe bits in Normal mode. The OC1x Register keeps its value even when changing between waveform generation modes.

Be aware that the COM1x1:0 bits are not double buffered together with the compare value. Changing the COM1x1:0 bits will take effect immediately.

Compare Match Output Unit

The *Compare Output mode* (COM1x1:0) bits have two functions. The Waveform Generator uses the COM1x1:0 bits for defining the Output Compare (OC1x) state at the next compare match. Secondly the COM1x1:0 bits control the OC1x pin output source. Figure 44 shows a simplified schematic of the logic affected by the COM1x1:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) that are affected by the COM1x1:0 bits are shown. When referring to the OC1x state, the reference is for the internal OC1x Register, not the OC1x pin. If a System Reset occur, the OC1x Register is reset to "0".

Figure 44. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC1x) from the Waveform Generator if either of the COM1x1:0 bits are set. However, the OC1x pin direction (input or output) is still controlled by the *Data Direction Register* (DDR) for the port pin. The Data Direction Register bit for the OC1x pin (DDR_OC1x) must be set as output before the OC1x value is visible on the pin. The port override function is generally independent of the Waveform Generation mode, but there are some exceptions. Refer to Table 44, Table 45 and Table 46 for details.

The design of the output compare pin logic allows initialization of the OC1x state before the output is enabled. Note that some COM1x1:0 bit settings are reserved for certain modes of operation. See "16-bit Timer/Counter Register Description" on page 110.

The COM1x1:0 bits have no effect on the Input Capture unit.

Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM1x1:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM1x1:0 = 0 tells the Waveform Generator that no action on the OC1x Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 44 on page 110. For fast PWM mode refer to Table 45 on page 111, and for phase correct and phase and frequency correct PWM refer to Table 46 on page 111.

A change of the COM1x1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC1x strobe bits.

Modes of Operation

The mode of operation, that is, the behavior of the Timer/Counter and the output compare pins, is defined by the combination of the *Waveform Generation mode* (WGM13:0) and *Compare Output mode* (COM1x1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM1x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM1x1:0 bits control whether the output should be set, cleared or toggle at a compare match (See "Compare Match Output Unit" on page 100.)

For detailed timing information refer to "Timer/Counter Timing Diagrams" on page 108.

Normal Mode

The simplest mode of operation is the *Normal* mode (WGM13:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 16-bit value (MAX = 0xFFFF) and then restarts from the BOTTOM (0x0000). In normal operation the *Timer/Counter Overflow Flag* (TOV1) will be set in the same timer clock cycle as the TCNT1 becomes zero. The TOV1 Flag in this case behaves like a 17th bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV1 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

The Input Capture unit is easy to use in Normal mode. However, observe that the maximum interval between the external events must not exceed the resolution of the counter. If the interval between events are too long, the timer overflow interrupt or the prescaler must be used to extend the resolution for the capture unit.

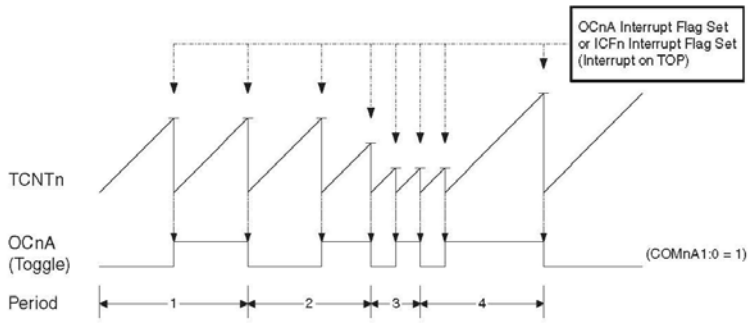
The output compare units can be used to generate interrupts at some given time. Using the output compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

Clear Timer on Compare Match (CTC) Mode

In *Clear Timer on Compare* or CTC mode (WGM13:0 = 4 or 12), the OCR1A or ICR1 Register are used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT1) matches either the OCR1A (WGM13:0 = 4) or the ICR1 (WGM13:0 = 12). The OCR1A or ICR1 define the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in Figure 45. The counter value (TCNT1) increases until a compare match occurs with either OCR1A or ICR1, and then counter (TCNT1) is cleared.

Figure 45. CTC Mode, Timing Diagram



An interrupt can be generated at each time the counter value reaches the TOP value by either using the OCF1A or ICF1 Flag according to the register used to define the TOP value. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing the TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR1A or ICR1 is lower than the current value of TCNT1, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. In many cases this feature is not desirable. An alternative will then be to use the fast PWM mode using OCR1A for defining TOP (WGM13:0 = 15) since the OCR1A then will be double buffered.

For generating a waveform output in CTC mode, the OC1A output can be set to toggle its logical level on each compare match by setting the compare output mode bits to toggle mode (COM1A1:0 = 1). The OC1A value will not be visible on the port pin unless the data direction for the pin is set to output (DDR_OC1A = 1). The waveform generated will have a maximum frequency of $f_{OC1A} = f_{clk\_I/O}/2$ when OCR1A is set to zero (0x0000). The waveform frequency is defined by the following equation:

$$f_{OCnA} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

The N variable represents the prescaler factor (1, 8, 64, 256, or 1024).

As for the Normal mode of operation, the TOV1 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x0000.

Fast PWM Mode

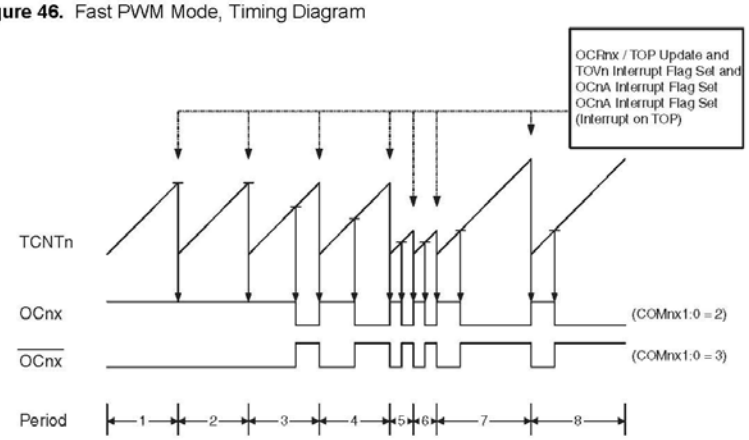
The *fast Pulse Width Modulation* or fast PWM mode (WGM13:0 = 5,6,7,14, or 15) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM options by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x, and set at BOTTOM. In inverting Compare Output mode output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct and phase and frequency correct PWM modes that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), hence reduces total system cost.

The PWM resolution for fast PWM can be fixed to 8-bit, 9-bit, or 10-bit, or defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated by using the following equation:

$$R_{FPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In fast PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM13:0 = 5, 6, or 7), the value in ICR1 (WGM13:0 = 14), or the value in OCR1A (WGM13:0 = 15). The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 46. The figure shows fast PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.

Figure 46. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches TOP. In addition the OC1A or ICF1 Flag is set at the same timer clock cycle as TOV1 is set when either OCR1A or ICR1 is used for defining the TOP value. If one of the interrupts are enabled, the interrupt handler routine can be used for updating the TOP and compare values.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1x. Note that when using fixed TOP values the unused bits are masked to zero when any of the OCR1x Registers are written.

The procedure for updating ICR1 differs from updating OCR1A when used for defining the TOP value. The ICR1 Register is not double buffered. This means that if ICR1 is changed to a low value when the counter is running with none or a low prescaler value, there is a risk that the new ICR1 value written is lower than the current value of TCNT1. The result will then be that the counter will miss the compare match at the TOP value. The counter will then have to count to the MAX value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. The OCR1A Register however, is double buffered. This feature allows the OCR1A I/O location to be written anytime. When the OCR1A I/O location is written the value written will be put into the OCR1A Buffer Register. The OCR1A Compare Register will then be updated with the value in the Buffer Register at the next timer clock cycle the TCNT1 matches TOP. The update is done at the same timer clock cycle as the TCNT1 is cleared and the TOV1 Flag is set.

Using the ICR1 Register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A Register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed (by changing the TOP value), using the OCR1A as TOP is clearly a better choice due to its double buffer feature.

In fast PWM mode, the compare units allow generation of PWM waveforms on the OC1x pins. Setting the COM1x1:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to 3 (See Table 44 on page 110). The actual OC1x value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1, and clearing (or setting) the OC1x Register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot (1 + TOP)}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR1x is set equal to BOTTOM (0x0000) the output will be a narrow spike for each TOP+1 timer clock cycle. Setting the OCR1x equal to TOP will result in a constant high or low output (depending on the polarity of the output set by the COM1x1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC1A to toggle its logical level on each compare match (COM1A1:0 = 1). This applies only if OCR1A is used to define the TOP value (WGM13:0 = 15). The waveform generated will have a maximum frequency of $f_{OC1A} = f_{clk\_I/O}/2$ when OCR1A is set to zero (0x0000). This feature is similar to the OC1A toggle in CTC mode, except the double buffer feature of the output compare unit is enabled in the fast PWM mode.

Phase Correct PWM Mode

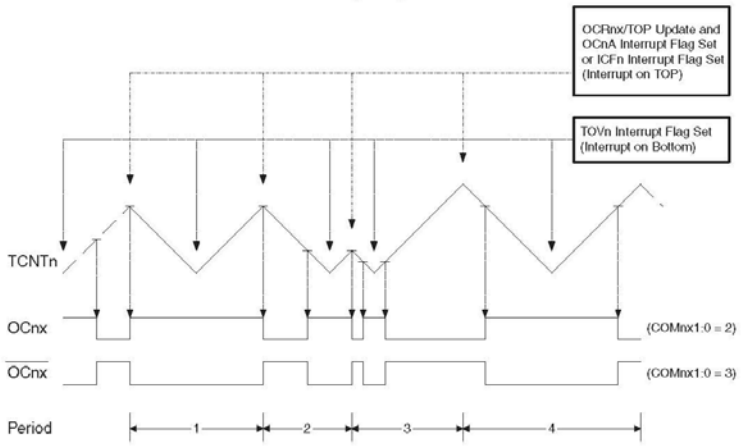
The *phase correct Pulse Width Modulation* or phase correct PWM mode (WGM13:0 = 1,2,3,10, or 11) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is, like the phase and frequency correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The PWM resolution for the phase correct PWM mode can be fixed to 8-bit, 9-bit, or 10-bit, or defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated by using the following equation:

$$R_{PCPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In phase correct PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM13:0 = 1, 2, or 3), the value in ICR1 (WGM13:0 = 10), or the value in OCR1A (WGM13:0 = 11). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 47. The figure shows phase correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.

Figure 47. Phase Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set each time the counter reaches BOTTOM. When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 Flag is set accordingly at the same timer clock cycle as the OCR1x Registers are updated with the double buffer value (at TOP). The Interrupt Flags can be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1x. Note that when using fixed TOP values, the unused bits are masked to zero when any of the OCR1x Registers are written. As the third period shown in Figure 47 illustrates, changing the TOP actively while the Timer/Counter is running in the phase correct mode can result in an unsymmetrical output. The reason for this can be found in the time of update of the OCR1x Register. Since the OCR1x update occurs at TOP, the PWM period starts and ends at TOP. This implies that the length of the falling slope is determined by the previous TOP value, while the length of the rising slope is determined by the new TOP value. When these two values differ the two slopes of the period will differ in length. The difference in length gives the unsymmetrical result on the output.

It is recommended to use the phase and frequency correct mode instead of the phase correct mode when changing the TOP value while the Timer/Counter is running. When using a static TOP value there are practically no differences between the two modes of operation.

In phase correct PWM mode, the compare units allow generation of PWM waveforms on the OC1x pins. Setting the COM1x1:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to 3 (See Table 44 on page 110). The actual OC1x value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1 when the counter increments, and clearing (or setting) the OC1x Register at compare match between OCR1x and TCNT1 when

the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot TOP}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR1x is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values. If OCR1A is used to define the TOP value (WGM13:0 = 11) and COM1A1:0 = 1, the OC1A output will toggle with a 50% duty cycle.

Phase and Frequency Correct PWM Mode

The *phase and frequency correct Pulse Width Modulation*, or phase and frequency correct PWM mode (WGM13:0 = 8 or 9) provides a high resolution phase and frequency correct PWM waveform generation option. The phase and frequency correct PWM mode is, like the phase correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the compare match between TCNT1 and OCR1x while upcounting, and set on the compare match while downcounting. In inverting Compare Output mode, the operation is inverted. The dual-slope operation gives a lower maximum operation frequency compared to the single-slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

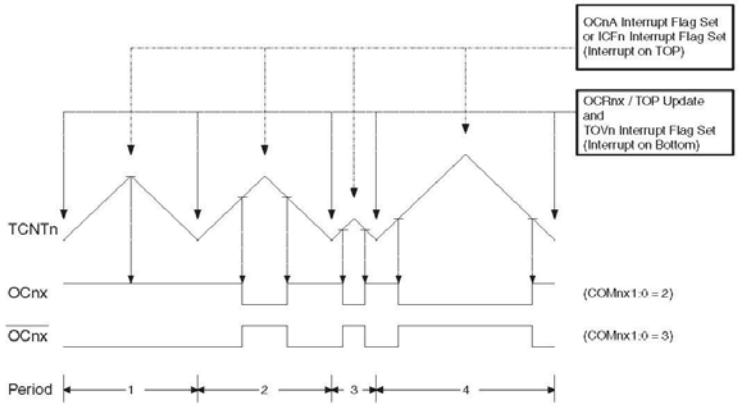
The main difference between the phase correct, and the phase and frequency correct PWM mode is the time the OCR1x Register is updated by the OCR1x Buffer Register, (see Figure 47 and Figure 48).

The PWM resolution for the phase and frequency correct PWM mode can be defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated using the following equation:

$$R_{PFCPWM} = \frac{\log(TOP+1)}{\log(2)}$$

In phase and frequency correct PWM mode the counter is incremented until the counter value matches either the value in ICR1 (WGM13:0 = 8), or the value in OCR1A (WGM13:0 = 9). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct and frequency correct PWM mode is shown on Figure 48. The figure shows phase and frequency correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.

Figure 48. Phase and Frequency Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set at the same timer clock cycle as the OCR1x Registers are updated with the double buffer value (at BOTTOM). When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 Flag set when TCNT1 has reached TOP. The Interrupt Flags can then be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1x.

As Figure 48 shows the output generated is, in contrast to the phase correct mode, symmetrical in all periods. Since the OCR1x Registers are updated at BOTTOM, the length of the rising and the falling slopes will always be equal. This gives symmetrical output pulses and is therefore frequency correct.

Using the ICR1 Register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A Register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed by changing the TOP value, using the OCR1A as TOP is clearly a better choice due to its double buffer feature.

In phase and frequency correct PWM mode, the compare units allow generation of PWM waveforms on the OC1x pins. Setting the COM1x1:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to 3 (See Table on page 111). The actual OC1x value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1 when the counter increments, and clearing (or setting) the OC1x Register at compare match between OCR1x and TCNT1 when the counter decrements. The PWM frequency for the output when using phase and frequency correct PWM can be calculated by the following equation:

$$f_{OCnxPFCPWM} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot TOP}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1x Register represents special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR1x is set equal to BOTTOM the

output will be continuously low and if set equal to TOP the output will be set to high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values. If OCR1A is used to define the TOP value (WGM13:0 = 9) and COM1A1:0 = 1, the OC1A output will toggle with a 50% duty cycle.

Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T1}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt Flags are set, and when the OCR1x Register is updated with the OCR1x buffer value (only for modes utilizing double buffering). Figure 49 shows a timing diagram for the setting of OCF1x.

Figure 49. Timer/Counter Timing Diagram, Setting of OCF1x, No Prescaling

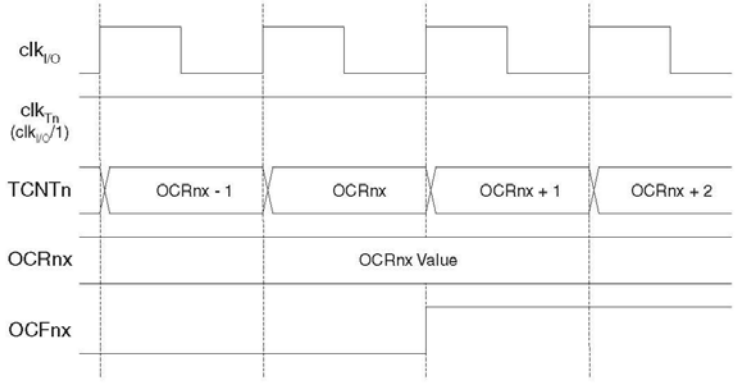


Figure 50 shows the same timing data, but with the prescaler enabled.

Figure 50. Timer/Counter Timing Diagram, Setting of OCF1x, with Prescaler ($f_{clk\_I/O}/8$)

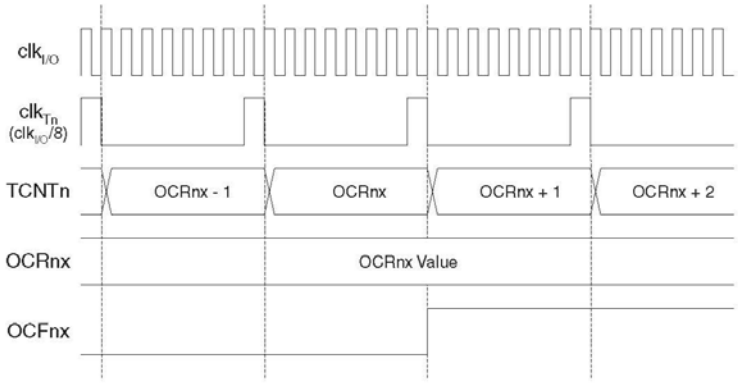


Figure 51 shows the count sequence close to TOP in various modes. When using phase and frequency correct PWM mode the OCR1x Register is updated at BOTTOM. The timing diagrams

will be the same, but TOP should be replaced by BOTTOM, TOP-1 by BOTTOM+1 and so on. The same renaming applies for modes that set the TOV1 Flag at BOTTOM.

Figure 51. Timer/Counter Timing Diagram, no Prescaling

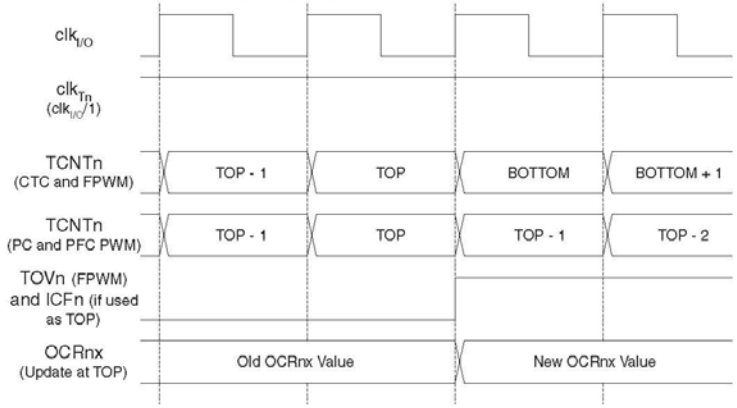
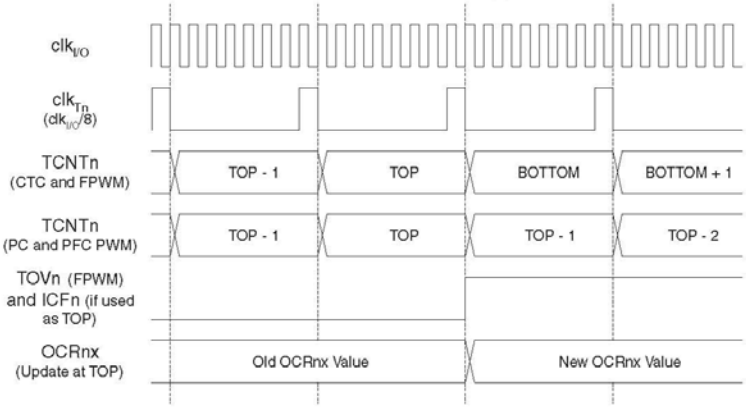


Figure 52 shows the same timing data, but with the prescaler enabled.

Figure 52. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)



16-bit Timer/Counter Register Description

Timer/Counter1 Control Register A – TCCR1A

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A**

- **Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B**

The COM1A1:0 and COM1B1:0 control the Output Compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the *Data Direction Register* (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. Table 44 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a normal or a CTC mode (non-PWM).

Table 44. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on compare match
1	0	Clear OC1A/OC1B on compare match (Set output to low level)
1	1	Set OC1A/OC1B on compare match (Set output to high level)

Table 45 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

Table 45. Compare Output Mode, Fast PWM[1]

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM13:0 settings, normal port operation, OCnA/OCnB disconnected.
1	0	Clear OC1A/OC1B on compare match, set OC1A/OC1B at BOTTOM, (non-inverting mode)
1	1	Set OC1A/OC1B on compare match, clear OC1A/OC1B at BOTTOM, (inverting mode)

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 102. for more details.

Table 46 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the phase correct or the phase and frequency correct, PWM mode.

Table 46. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM [1]

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 9 or 14: Toggle OCnA on Compare Match, OCnB disconnected (normal port operation). For all other WGM13:0 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on compare match when up-counting. Set OC1A/OC1B on compare match when downcounting.
1	1	Set OC1A/OC1B on compare match when up-counting. Clear OC1A/OC1B on compare match when downcounting.

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. See "Phase Correct PWM Mode" on page 104. for more details.

- **Bit 3 – FOC1A: Force Output Compare for Channel A**
- **Bit 2 – FOC1B: Force Output Compare for Channel B**

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-PWM mode. However, for ensuring compatibility with future devices, these bits must be set to zero when TCCR1A is written when operating in a PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the Waveform Generation unit. The OC1A/OC1B output is changed according to its COM1x1:0 bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1x1:0 bits that determine the effect of the forced compare.

A FOC1A/FOC1B strobe will not generate any interrupt nor will it clear the timer in Clear Timer on Compare match (CTC) mode using OCR1A as TOP.

The FOC1A/FOC1B bits are always read as zero.

- **Bit 1:0 – WGM11:0: Waveform Generation Mode**

Combined with the WGM13:2 bits found in the TCCR1B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 47. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare match (CTC) mode, and three types of Pulse Width Modulation (PWM) modes. (See "Modes of Operation" on page 101.)

Table 47. Waveform Generation Mode Bit Description[1]

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1X	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

Timer/Counter1 Control Register B – TCCR1B

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNC1: Input Capture Noise Canceler**

Setting this bit (to one) activates the Input Capture Noise Canceler. When the Noise Canceler is activated, the input from the Input Capture Pin (ICP1) is filtered. The filter function requires four successive equal valued samples of the ICP1 pin for changing its output. The Input Capture is therefore delayed by four Oscillator cycles when the Noise Canceler is enabled.

- **Bit 6 – ICES1: Input Capture Edge Select**

This bit selects which edge on the Input Capture Pin (ICP1) that is used to trigger a capture event. When the ICES1 bit is written to zero, a falling (negative) edge is used as trigger, and when the ICES1 bit is written to one, a rising (positive) edge will trigger the capture.

When a capture is triggered according to the ICES1 setting, the counter value is copied into the Input Capture Register (ICR1). The event will also set the Input Capture Flag (ICF1), and this can be used to cause an Input Capture Interrupt, if this interrupt is enabled.

When the ICR1 is used as TOP value (see description of the WGM13:0 bits located in the TCCR1A and the TCCR1B Register), the ICP1 is disconnected and consequently the Input Capture function is disabled.

- **Bit 5 – Reserved Bit**

This bit is reserved for future use. For ensuring compatibility with future devices, this bit must be written to zero when TCCR1B is written.

- **Bit 4:3 – WGM13:2: Waveform Generation Mode**

See TCCR1A Register description.

- **Bit 2:0 – CS12:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter, see Figure 49 and Figure 50.

Table 48. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter1, transitions on the T1 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

Timer/Counter1 – TCNT1H and TCNT1L

Bit	7	6	5	4	3	2	1	0	
	TCNT1[15:8]								TCNT1H
	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The two *Timer/Counter* I/O locations (TCNT1H and TCNT1L, combined TCNT1) give direct access, both for read and for write operations, to the Timer/Counter unit 16-bit counter. To ensure that both the high and Low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See "Accessing 16-bit Registers" on page 92.

Modifying the counter (TCNT1) while the counter is running introduces a risk of missing a compare match between TCNT1 and one of the OCR1x Registers.

Writing to the TCNT1 Register blocks (removes) the compare match on the following timer clock for all compare units.

Output Compare Register 1 A – OCR1AH and OCR1AL

Bit	7	6	5	4	3	2	1	0	
	OCR1A[15:8]								OCR1AH
	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Output Compare Register 1 B – OCR1BH and OCR1BL

Bit	7	6	5	4	3	2	1	0	
	OCR1B[15:8]								OCR1BH
	OCR1B[7:0]								OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Registers contain a 16-bit value that is continuously compared with the counter value (TCNT1). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC1x pin.

The Output Compare Registers are 16-bit in size. To ensure that both the high and Low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See "Accessing 16-bit Registers" on page 92.

Input Capture Register 1 – ICR1H and ICR1L

Bit	7	6	5	4	3	2	1	0	
	ICR1[15:8]								ICR1H
	ICR1[7:0]								ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Input Capture is updated with the counter (TCNT1) value each time an event occurs on the ICP1 pin (or optionally on the analog comparator output for Timer/Counter1). The Input Capture can be used for defining the counter TOP value.

The Input Capture Register is 16-bit in size. To ensure that both the high and Low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See "Accessing 16-bit Registers" on page 92.

Timer/Counter Interrupt Mask Register – TIMSK[1]

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. This register contains interrupt control bits for several Timer/Counters, but only Timer1 bits are described in this section. The remaining bits are described in their respective timer sections.

- **Bit 5 – TICIE1: Timer/Counter1, Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Input Capture Interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the ICF1 Flag, located in TIFR, is set.

- **Bit 4 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare A match interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the OCF1A Flag, located in TIFR, is set.

- **Bit 3 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare B match interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the OCF1B Flag, located in TIFR, is set.

- **Bit 2 – TOIE1: Timer/Counter1, Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Overflow Interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 45.) is executed when the TOV1 Flag, located in TIFR, is set.

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Note: This register contains flag bits for several Timer/Counters, but only Timer1 bits are described in this section. The remaining bits are described in their respective timer sections.

- **Bit 5 – ICF1: Timer/Counter1, Input Capture Flag**

This flag is set when a capture event occurs on the ICP1 pin. When the Input Capture Register (ICR1) is set by the WGM13:0 to be used as the TOP value, the ICF1 Flag is set when the counter reaches the TOP value.

ICF1 is automatically cleared when the Input Capture Interrupt Vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- **Bit 4 – OCF1A: Timer/Counter1, Output Compare A Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register A (OCR1A).

Note that a Forced Output Compare (FOC1A) strobe will not set the OCF1A Flag.

OCF1A is automatically cleared when the Output Compare Match A Interrupt Vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- **Bit 3 – OCF1B: Timer/Counter1, Output Compare B Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register B (OCR1B).

Note that a forced output compare (FOC1B) strobe will not set the OCF1B Flag.

OCF1B is automatically cleared when the Output Compare Match B Interrupt Vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

- **Bit 2 – TOV1: Timer/Counter1, Overflow Flag**

The setting of this flag is dependent of the WGM13:0 bits setting. In normal and CTC modes, the TOV1 Flag is set when the timer overflows. Refer to Table 47 on page 112 for the TOV1 Flag behavior when using another WGM13:0 bit setting.

TOV1 is automatically cleared when the Timer/Counter1 Overflow interrupt vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

8-bit Timer/Counter2 with PWM and Asynchronous Operation

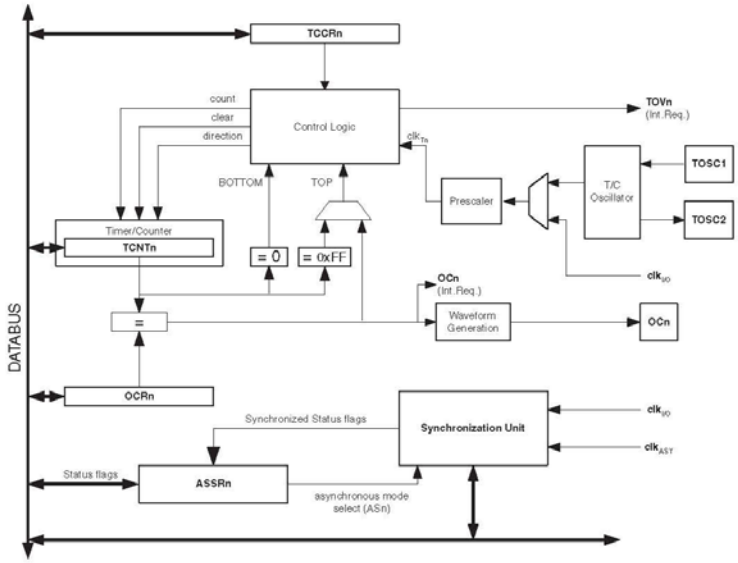
Timer/Counter2 is a general purpose, single compare unit, 8-bit Timer/Counter module. The main features are:

- **Single Compare unit Counter**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch-free, Phase Correct Pulse Width Modulator (PWM)**
- **Frequency Generator**
- **10-bit Clock Prescaler**
- **Overflow and Compare Match Interrupt Sources (TOV2 and OCF2)**
- **Allows clocking from External 32 kHz Watch Crystal Independent of the I/O Clock**

Overview

A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 53. For the actual placement of I/O pins, refer to "Pinout ATmega16" on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "8-bit Timer/Counter Register Description" on page 128.

Figure 53. 8-bit Timer/Counter Block Diagram



Registers

The Timer/Counter (TCNT2) and Output Compare Register (OCR2) are 8-bit registers. Interrupt request (shorten as Int.Req.) signals are all visible in the Timer Interrupt Flag Register (TIFR). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK). TIFR and TIMSK are not shown in the figure since these registers are shared by other timer units.

The Timer/Counter can be clocked internally, via the prescaler, or asynchronously clocked from the TOSC1/2 pins, as detailed later in this section. The asynchronous operation is controlled by the Asynchronous Status Register (ASSR). The Clock Select logic block controls which clock source the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T2}).

The double buffered Output Compare Register (OCR2) is compared with the Timer/Counter value at all times. The result of the compare can be used by the waveform generator to generate a PWM or variable frequency output on the Output Compare Pin (OC2). See "Output Compare Unit" on page 119. for details. The compare match event will also set the Compare Flag (OCF2) which can be used to generate an output compare interrupt request.

Definitions

Many register and bit references in this document are written in general form. A lower case "n" replaces the Timer/Counter number, in this case 2. However, when using the register or bit defines in a program, the precise form must be used (that is, TCNT2 for accessing Timer/Counter2 counter value and so on). The definitions in Table 49 are also used extensively throughout the document.

Table 49. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes zero (0x00).
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR2 Register. The assignment is dependent on the mode of operation.

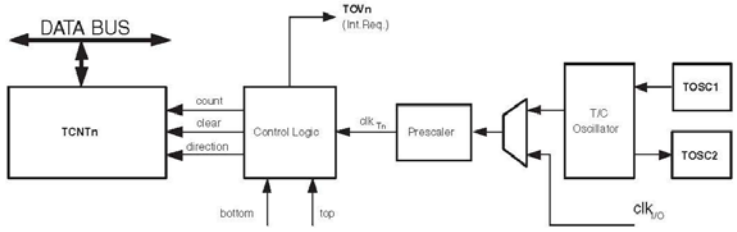
Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal synchronous or an external asynchronous clock source. The clock source clk_{T2} is by default equal to the MCU clock, $clk_{I/O}$. When the AS2 bit in the ASSR Register is written to logic one, the clock source is taken from the Timer/Counter Oscillator connected to TOSC1 and TOSC2. For details on asynchronous operation, see "Asynchronous Status Register – ASSR" on page 131. For details on clock sources and prescaler, see "Timer/Counter Prescaler" on page 134.

Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 54 shows a block diagram of the counter and its surrounding environment.

Figure 54. Counter Unit Block Diagram



Signal description (internal signals):

count	Increment or decrement TCNT2 by 1.
direction	Selects between increment and decrement.
clear	Clear TCNT2 (set all bits to zero).
clk_{T2}	Timer/Counter clock.
top	Signalizes that TCNT2 has reached maximum value.

bottom Signalizes that TCNT2 has reached minimum value (zero).

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T2}). clk_{T2} can be generated from an external or internal clock source, selected by the Clock Select bits (CS22:0). When no clock source is selected (CS22:0 = 0) the timer is stopped. However, the TCNT2 value can be accessed by the CPU, regardless of whether clk_{T2} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

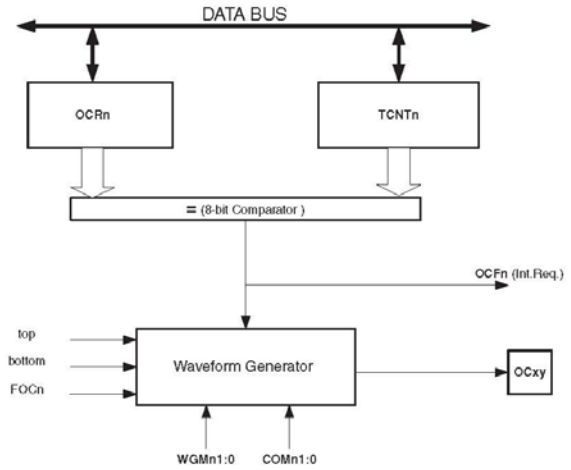
The counting sequence is determined by the setting of the WGM21 and WGM20 bits located in the Timer/Counter Control Register (TCCR2). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare output OC2. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 122.

The Timer/Counter Overflow (TOV2) Flag is set according to the mode of operation selected by the WGM21:0 bits. TOV2 can be used for generating a CPU interrupt.

Output Compare Unit

The 8-bit comparator continuously compares TCNT2 with the Output Compare Register (OCR2). Whenever TCNT2 equals OCR2, the comparator signals a match. A match will set the Output Compare Flag (OCF2) at the next timer clock cycle. If enabled (OCIE2 = 1), the Output Compare Flag generates an output compare interrupt. The OCF2 Flag is automatically cleared when the interrupt is executed. Alternatively, the OCF2 Flag can be cleared by software by writing a logical one to its I/O bit location. The waveform generator uses the match signal to generate an output according to operating mode set by the WGM21:0 bits and Compare Output mode (COM21:0) bits. The max and bottom signals are used by the waveform generator for handling the special cases of the extreme values in some modes of operation ("Modes of Operation" on page 122). Figure 55 shows a block diagram of the output compare unit.

Figure 55. Output Compare Unit, Block Diagram



The OCR2 Register is double buffered when using any of the Pulse Width Modulation (PWM) modes. For the normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR2 Compare Register

to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR2 Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR2 Buffer Register, and if double buffering is disabled the CPU will access the OCR2 directly.

Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC2) bit. Forcing compare match will not set the OCF2 Flag or reload/clear the timer, but the OC2 pin will be updated as if a real compare match had occurred (the COM21:0 bits settings define whether the OC2 pin is set, cleared or toggled).

Compare Match Blocking by TCNT2 Write

All CPU write operations to the TCNT2 Register will block any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows OCR2 to be initialized to the same value as TCNT2 without triggering an interrupt when the Timer/Counter clock is enabled.

Using the Output Compare Unit

Since writing TCNT2 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT2 when using the output compare unit, independently of whether the Timer/Counter is running or not. If the value written to TCNT2 equals the OCR2 value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT2 value equal to BOTTOM when the counter is downcounting.

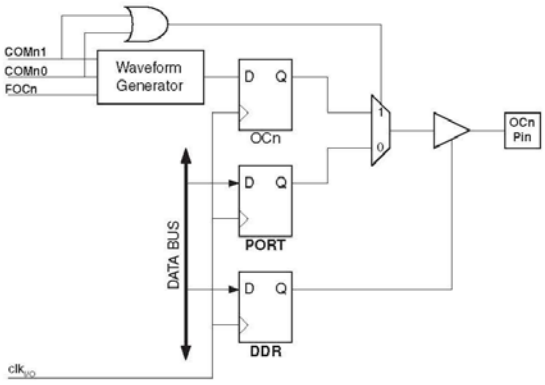
The setup of the OC2 should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC2 value is to use the Force Output Compare (FOC2) strobe bit in Normal mode. The OC2 Register keeps its value even when changing between Waveform Generation modes.

Be aware that the COM21:0 bits are not double buffered together with the compare value. Changing the COM21:0 bits will take effect immediately.

Compare Match Output Unit

The Compare Output mode (COM21:0) bits have two functions. The Waveform Generator uses the COM21:0 bits for defining the Output Compare (OC2) state at the next compare match. Also, the COM21:0 bits control the OC2 pin output source. Figure 56 shows a simplified schematic of the logic affected by the COM21:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) that are affected by the COM21:0 bits are shown. When referring to the OC2 state, the reference is for the internal OC2 Register, not the OC2 pin.

Figure 56. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC2) from the waveform generator if either of the COM21:0 bits are set. However, the OC2 pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC2 pin (DDR_OC2) must be set as output before the OC2 value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the output compare pin logic allows initialization of the OC2 state before the output is enabled. Note that some COM21:0 bit settings are reserved for certain modes of operation. See "8-bit Timer/Counter Register Description" on page 128.

Compare Output Mode and Waveform Generation

The waveform generator uses the COM21:0 bits differently in Normal, CTC, and PWM modes. For all modes, setting the COM21:0 = 0 tells the Waveform Generator that no action on the OC2 Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 51 on page 129. For fast PWM mode, refer to Table 52 on page 129, and for phase correct PWM refer to Table 53 on page 129.

A change of the COM21:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC2 strobe bits.

Modes of Operation

The mode of operation, that is, the behavior of the Timer/Counter and the output compare pins, is defined by the combination of the Waveform Generation mode (WGM21:0) and Compare Output mode (COM21:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM21:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM21:0 bits control whether the output should be set, cleared, or toggled at a compare match (See "Compare Match Output Unit" on page 121.).

For detailed timing information refer to "Timer/Counter Timing Diagrams" on page 126.

Normal Mode

The simplest mode of operation is the Normal mode (WGM21:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (TOP = 0xFF) and then restarts from the bottom (0x00). In normal operation the Timer/Counter Overflow Flag (TOV2) will be set in the same timer clock cycle as the TCNT2 becomes zero. The TOV2 Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV2 Flag, the timer resolution can be increased by software. There are no special cases to consider in the normal mode, a new counter value can be written anytime.

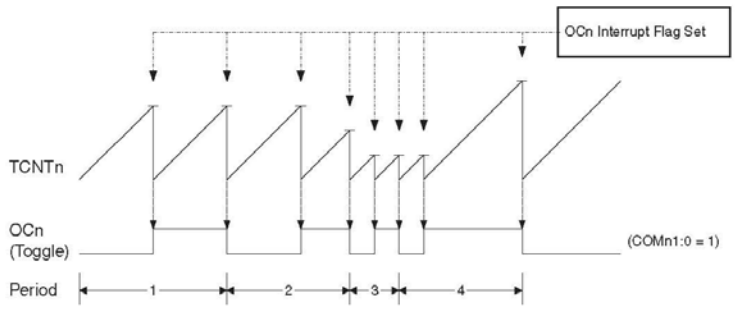
The Output Compare unit can be used to generate interrupts at some given time. Using the output compare to generate waveforms in normal mode is not recommended, since this will occupy too much of the CPU time.

Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM21:0 = 2), the OCR2 Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT2) matches the OCR2. The OCR2 defines the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in Figure 57. The counter value (TCNT2) increases until a compare match occurs between TCNT2 and OCR2, and then counter (TCNT2) is cleared.

Figure 57. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF2 Flag. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing the TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR2 is lower than the current

value of TCNT2, the counter will miss the compare match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the compare match can occur.

For generating a waveform output in CTC mode, the OC2 output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM21:0 = 1). The OC2 value will not be visible on the port pin unless the data direction for the pin is set to output. The waveform generated will have a maximum frequency of $f_{OC2} = f_{clk\_I/O}/2$ when OCR2 is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRn)}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

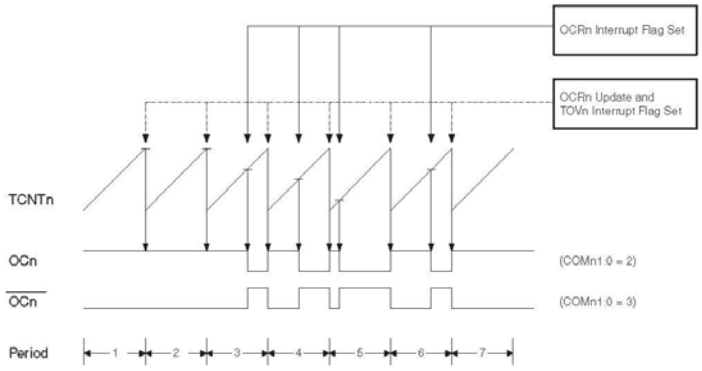
As for the Normal mode of operation, the TOV2 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM21:0 = 3) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to MAX then restarts from BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC2) is cleared on the compare match between TCNT2 and OCR2, and set at BOTTOM. In inverting Compare Output mode, the output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that uses dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the MAX value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 58. The TCNT2 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT2 slopes represent compare matches between OCR2 and TCNT2.

Figure 58. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV2) is set each time the counter reaches MAX. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC2 pin. Setting the COM21:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM21:0 to 3 (see Table 52 on page 129). The actual OC2 value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC2 Register at the compare match between OCR2 and TCNT2, and clearing (or setting) the OC2 Register at the timer clock cycle the counter is cleared (changes from MAX to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

The extreme values for the OCR2 Register represent special cases when generating a PWM waveform output in the fast PWM mode. If the OCR2 is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR2 equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM21:0 bits.)

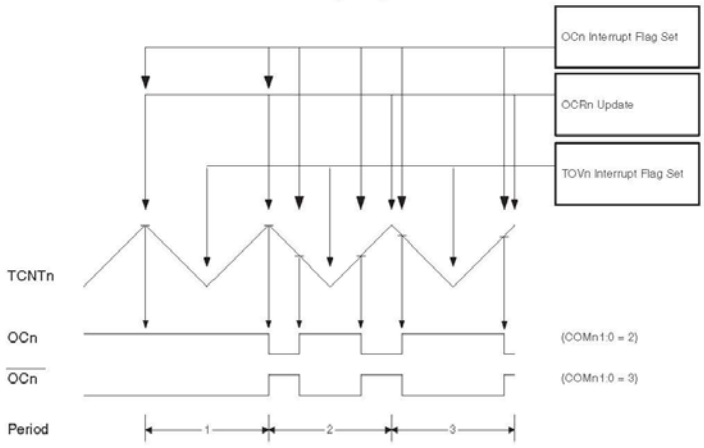
A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC2 to toggle its logical level on each compare match (COM21:0 = 1). The waveform generated will have a maximum frequency of $f_{oc2} = f_{clk\_I/O}/2$ when OCR2 is set to zero. This feature is similar to the OC2 toggle in CTC mode, except the double buffer feature of the output compare unit is enabled in the fast PWM mode.

Phase Correct PWM Mode

The phase correct PWM mode (WGM21:0 = 1) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to MAX and then from MAX to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC2) is cleared on the compare match between TCNT2 and OCR2 while upcounting, and set on the compare match while downcounting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The PWM resolution for the phase correct PWM mode is fixed to 8 bits. In phase correct PWM mode the counter is incremented until the counter value matches MAX. When the counter reaches MAX, it changes the count direction. The TCNT2 value will be equal to MAX for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 59. The TCNT2 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT2 slopes represent compare matches between OCR2 and TCNT2.

Figure 59. Phase Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV2) is set each time the counter reaches BOTTOM. The Interrupt Flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

In phase correct PWM mode, the compare unit allows generation of PWM waveforms on the OC2 pin. Setting the COM21:0 bits to 2 will produce a non-inverted PWM. An inverted PWM output can be generated by setting the COM21:0 to 3 (see Table 53 on page 129). The actual OC2 value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by clearing (or setting) the OC2 Register at the compare match between OCR2 and TCNT2 when the counter increments, and setting (or clearing) the OC2 Register at compare match between OCR2 and TCNT2 when the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

The extreme values for the OCR2 Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR2 is set equal to BOTTOM, the output will be continuously low and if set equal to MAX the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

At the very start of Period 2 in Figure 59 OCn has a transition from high to l ow even though there is no Compare Match. The point of this transition is to guarantee symmetry around BOTTOM. There are two cases that will give transition without Compare Match:

- OCR2A changes its value from Max, like in Figure 59. When the OCR2A value is MAX the OCn pin value is the same as the result of a down-counting Compare Match. To ensure symmetry around BOTTOM the OCn value at MAX must be correspond the the result of an up-counting Compare Match.
- The Timer starts counting from a value higher than the one in OCR2A, and for that reason misses the Compare Match and hence the OCn that would have happened on the way up.

Timer/Counter Timing Diagrams

The following figures show the Timer/Counter in Synchronous mode, and the timer clock (clk_{T2}) is therefore shown as a clock enable signal. In Asynchronous mode, $clk_{I/O}$ should be replaced by the Timer/Counter Oscillator clock. The figures include information on when Interrupt Flags are set. Figure 60 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 60. Timer/Counter Timing Diagram, no Prescaling

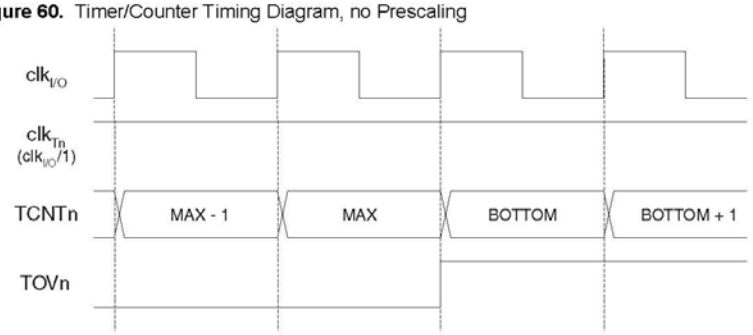


Figure 61 shows the same timing data, but with the prescaler enabled.

Figure 61. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

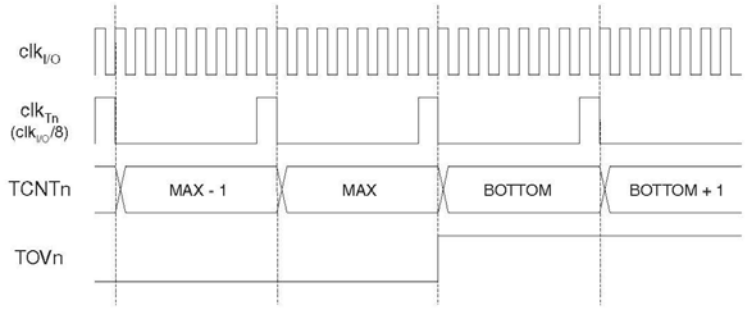


Figure 62 shows the setting of OCF2 in all modes except CTC mode.

Figure 62. Timer/Counter Timing Diagram, Setting of OCF2, with Prescaler ($f_{clk\_I/O}/8$)

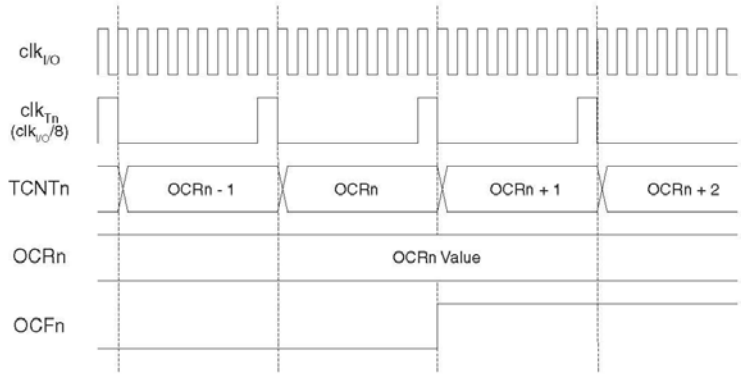
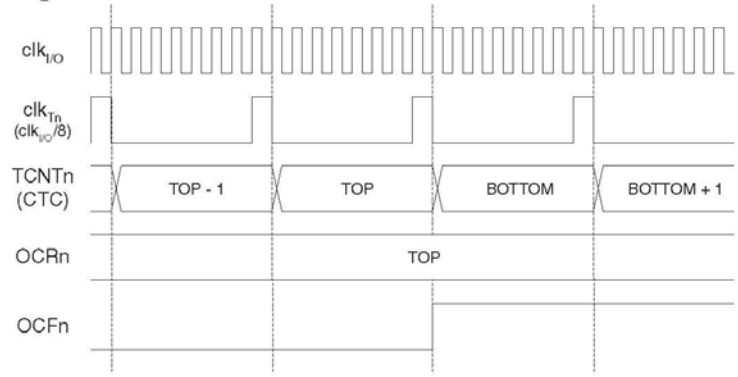


Figure 63 shows the setting of OCF2 and the clearing of TCNT2 in CTC mode.

Figure 63. Timer/Counter Timing Diagram, Clear Timer on Compare Match Mode, with Prescaler ($f_{clk\_I/O}/8$)



8-bit Timer/Counter Register Description

Timer/Counter Control Register – TCCR2

Bit	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC2: Force Output Compare**

The FOC2 bit is only active when the WGM bits specify a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR2 is written when operating in PWM mode. When writing a logical one to the FOC2 bit, an immediate compare match is forced on the waveform generation unit. The OC2 output is changed according to its COM21:0 bits setting. Note that the FOC2 bit is implemented as a strobe. Therefore it is the value present in the COM21:0 bits that determines the effect of the forced compare.

A FOC2 strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR2 as TOP.

The FOC2 bit is always read as zero.

- **Bit 3, 6 – WGM21:0: Waveform Generation Mode**

These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of waveform generation to be used. Modes of operation supported by the Timer/Counter unit are: Normal mode, Clear Timer on Compare match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes. See Table 50 and "Modes of Operation" on page 122.

Table 50. Waveform Generation Mode Bit Description[1]

Mode	WGM21 (CTC2)	WGM20 (PWM2)	Timer/Counter Mode of Operation	TOP	Update of OCR2	TOV2 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR2	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Note: 1. The CTC2 and PWM2 bit definition names are now obsolete. Use the WGM21:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

- **Bit 5:4 – COM21:0: Compare Match Output Mode**

These bits control the Output Compare pin (OC2) behavior. If one or both of the COM21:0 bits are set, the OC2 output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to OC2 pin must be set in order to enable the output driver.

When OC2 is connected to the pin, the function of the COM21:0 bits depends on the WGM21:0 bit setting. Table 51 shows the COM21:0 bit functionality when the WGM21:0 bits are set to a normal or CTC mode (non-PWM).

Table 51. Compare Output Mode, non-PWM Mode

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Toggle OC2 on compare match
1	0	Clear OC2 on compare match
1	1	Set OC2 on compare match

Table 52 shows the COM21:0 bit functionality when the WGM21:0 bits are set to fast PWM mode.

Table 52. Compare Output Mode, Fast PWM Mode(1)

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on compare match, set OC2 at BOTTOM, (non-inverting mode)
1	1	Set OC2 on compare match, clear OC2 at BOTTOM, (inverting mode)

Note: 1. A special case occurs when OCR2 equals TOP and COM21 is set. In this case, the compare match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 123 for more details.

Table 53 shows the COM21:0 bit functionality when the WGM21:0 bits are set to phase correct PWM mode

Table 53. Compare Output Mode, Phase Correct PWM Mode(1)

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on compare match when up-counting. Set OC2 on compare match when downcounting.
1	1	Set OC2 on compare match when up-counting. Clear OC2 on compare match when downcounting.

Note: 1. A special case occurs when OCR2 equals TOP and COM21 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 124 for more details.

- **Bit 2:0 – CS22:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter, see Table 54.

Table 54. Clock Select Bit Description

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk_{T2S}/(No prescaling)
0	1	0	clk_{T2S}/8 (From prescaler)
0	1	1	clk_{T2S}/32 (From prescaler)
1	0	0	clk_{T2S}/64 (From prescaler)
1	0	1	clk_{T2S}/128 (From prescaler)
1	1	0	clk_{T2S}/256 (From prescaler)
1	1	1	clk_{T2S}/1024 (From prescaler)

Timer/Counter Register – TCNT2

Bit	7	6	5	4	3	2	1	0	
	TCNT2[7:0]								TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT2 Register blocks (removes) the compare match on the following timer clock. Modifying the counter (TCNT2) while the counter is running, introduces a risk of missing a compare match between TCNT2 and the OCR2 Register.

Output Compare Register – OCR2

Bit	7	6	5	4	3	2	1	0	
	OCR2[7:0]								OCR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register contains an 8-bit value that is continuously compared with the counter value (TCNT2). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC2 pin.

Asynchronous Operation of the Timer/Counter

Asynchronous Status Register – ASSR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 3 – AS2: Asynchronous Timer/Counter2**

When AS2 is written to zero, Timer/Counter 2 is clocked from the I/O clock, $clk_{I/O}$. When AS2 is written to one, Timer/Counter2 is clocked from a Crystal Oscillator connected to the Timer Oscillator 1 (TOSC1) pin. When the value of AS2 is changed, the contents of TCNT2, OCR2, and TCCR2 might be corrupted.

- **Bit 2 – TCN2UB: Timer/Counter2 Update Busy**

When Timer/Counter2 operates asynchronously and TCNT2 is written, this bit becomes set. When TCNT2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCNT2 is ready to be updated with a new value.

- **Bit 1 – OCR2UB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2 is written, this bit becomes set. When OCR2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2 is ready to be updated with a new value.

- **Bit 0 – TCR2UB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2 is written, this bit becomes set. When TCCR2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2 is ready to be updated with a new value.

If a write is performed to any of the three Timer/Counter2 Registers while its update busy flag is set, the updated value might get corrupted and cause an unintentional interrupt to occur.

The mechanisms for reading TCNT2, OCR2, and TCCR2 are different. When reading TCNT2, the actual timer value is read. When reading OCR2 or TCCR2, the value in the temporary storage register is read.

Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken.

- Warning: When switching between asynchronous and synchronous clocking of Timer/Counter2, the Timer Registers TCNT2, OCR2, and TCCR2 might be corrupted. A safe procedure for switching clock source is:
  1. Disable the Timer/Counter2 interrupts by clearing OCIE2 and TOIE2.
  2. Select clock source by setting AS2 as appropriate.
  3. Write new values to TCNT2, OCR2, and TCCR2.
  4. To switch to asynchronous operation: Wait for TCN2UB, OCR2UB, and TCR2UB.
  5. Clear the Timer/Counter2 Interrupt Flags.
  6. Enable interrupts, if needed.

- The Oscillator is optimized for use with a 32.768 kHz watch crystal. Applying an external clock to the TOSC1 pin may result in incorrect Timer/Counter2 operation. The CPU main clock frequency must be more than four times the Oscillator frequency.
- When writing to one of the registers TCNT2, OCR2, or TCCR2, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means for example that writing to TCNT2 does not disturb an OCR2 write in progress. To detect that a transfer to the destination register has taken place, the Asynchronous Status Register – ASSR has been implemented.
- When entering Power-save or Extended Standby mode after having written to TCNT2, OCR2, or TCCR2, the user must wait until the written register has been updated if Timer/Counter2 is used to wake up the device. Otherwise, the MCU will enter sleep mode before the changes are effective. This is particularly important if the Output Compare2 interrupt is used to wake up the device, since the output compare function is disabled during writing to OCR2 or TCNT2. If the write cycle is not finished, and the MCU enters sleep mode before the OCR2UB bit returns to zero, the device will never receive a compare match interrupt, and the MCU will not wake up.
- If Timer/Counter2 is used to wake the device up from Power-save or Extended Standby mode, precautions must be taken if the user wants to re-enter one of these modes: The interrupt logic needs one TOSC1 cycle to be reset. If the time between wake-up and re-entering sleep mode is less than one TOSC1 cycle, the interrupt will not occur, and the device will fail to wake up. If the user is in doubt whether the time before re-entering Power-save or Extended Standby mode is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
  1. Write a value to TCCR2, TCNT2, or OCR2.
  2. Wait until the corresponding Update Busy Flag in ASSR returns to zero.
  3. Enter Power-save or Extended Standby mode.
- When the asynchronous operation is selected, the 32.768 kHz Oscillator for Timer/Counter2 is always running, except in Power-down and Standby modes. After a Power-up Reset or wake-up from Power-down or Standby mode, the user should be aware of the fact that this Oscillator might take as long as one second to stabilize. The user is advised to wait for at least one second before using Timer/Counter2 after power-up or wake-up from Power-down or Standby mode. The contents of all Timer/Counter2 Registers must be considered lost after a wake-up from Power-down or Standby mode due to unstable clock signal upon start-up, no matter whether the Oscillator is in use or a clock signal is applied to the TOSC1 pin.
- Description of wake up from Power-save or Extended Standby mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. After wake-up, the MCU is halted for four cycles, it executes the interrupt routine, and resumes execution from the instruction following SLEEP.
- Reading of the TCNT2 Register shortly after wake-up from Power-save may give an incorrect result. Since TCNT2 is clocked on the asynchronous TOSC clock, reading TCNT2 must be done through a register synchronized to the internal I/O clock domain. Synchronization takes place for every rising TOSC1 edge. When waking up from Power-save mode, and the I/O clock ($clk_{I/O}$) again becomes active, TCNT2 will read as the previous value (before entering sleep) until the next rising TOSC1 edge. The phase of the TOSC clock after waking up from Power-save mode is essentially unpredictable, as it depends on the wake-up time. The recommended procedure for reading TCNT2 is thus as follows:

1. Write any value to either of the registers OCR2 or TCCR2.
2. Wait for the corresponding Update Busy Flag to be cleared.
3. Read TCNT2.

- During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes three processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the Interrupt Flag. The output compare pin is changed on the timer clock and is not synchronized to the processor clock.

Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – OCIE2: Timer/Counter2 Output Compare Match Interrupt Enable**

When the OCIE2 bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter2 occurs, that is, when the OCF2 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 6 – TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter2 occurs, that is, when the TOV2 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – OCF2: Output Compare Flag 2**

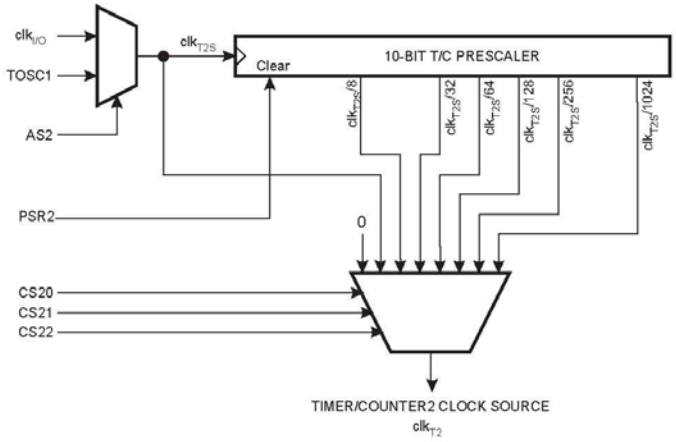
The OCF2 bit is set (one) when a compare match occurs between the Timer/Counter2 and the data in OCR2 – Output Compare Register2. OCF2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2 is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE2 (Timer/Counter2 Compare match Interrupt Enable), and OCF2 are set (one), the Timer/Counter2 Compare match Interrupt is executed.

- **Bit 6 – TOV2: Timer/Counter2 Overflow Flag**

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV2 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE2 (Timer/Counter2 Overflow Interrupt Enable), and TOV2 are set (one), the Timer/Counter2 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter2 changes counting direction at $00.

Timer/Counter Prescaler

Figure 64. Prescaler for Timer/Counter2



The clock source for Timer/Counter2 is named clk_{T2S}. clk_{T2S} is by default connected to the main system I/O clock clk_{IO}. By setting the AS2 bit in ASSR, Timer/Counter2 is asynchronously clocked from the TOSC1 pin. This enables use of Timer/Counter2 as a Real Time Counter (RTC). When AS2 is set, pins TOSC1 and TOSC2 are disconnected from Port C. A crystal can then be connected between the TOSC1 and TOSC2 pins to serve as an independent clock source for Timer/Counter2. The Oscillator is optimized for use with a 32.768 kHz crystal. Applying an external clock source to TOSC1 is not recommended.

For Timer/Counter2, the possible prescaled selections are: $clk_{T2S}/8$, $clk_{T2S}/32$, $clk_{T2S}/64$, $clk_{T2S}/128$, $clk_{T2S}/256$, and $clk_{T2S}/1024$. Additionally, clk_{T2S} as well as 0 (stop) may be selected. Setting the PSR2 bit in SFIOR resets the prescaler. This allows the user to operate with a predictable prescaler.

Special Function IO Register – SFIOR

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – PSR2: Prescaler Reset Timer/Counter2**

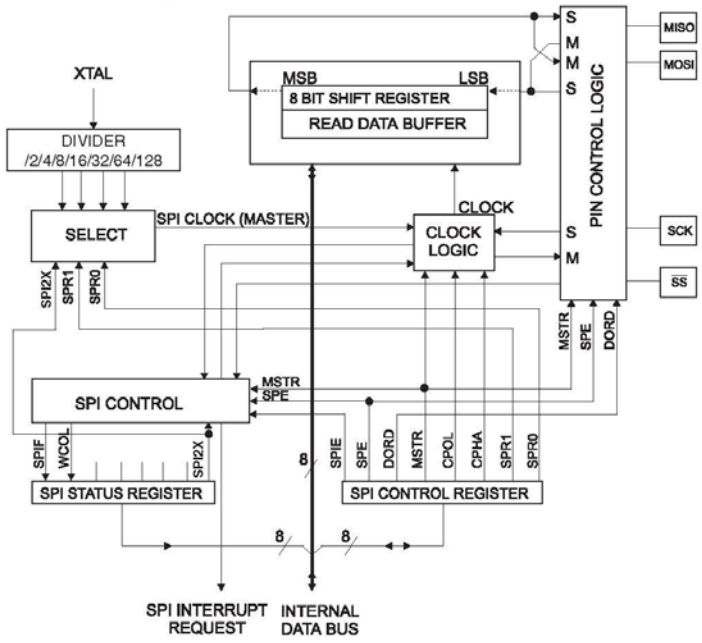
When this bit is written to one, the Timer/Counter2 prescaler will be reset. The bit will be cleared by hardware after the operation is performed. Writing a zero to this bit will have no effect. This bit will always be read as zero if Timer/Counter2 is clocked by the internal CPU clock. If this bit is written when Timer/Counter2 is operating in asynchronous mode, the bit will remain one until the prescaler has been reset.

Serial Peripheral Interface – SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega16 and peripheral devices or between several AVR devices. The ATmega16 SPI includes the following features:

- **Full-duplex, Three-wire Synchronous Data Transfer**
- **Master or Slave Operation**
- **LSB First or MSB First Data Transfer**
- **Seven Programmable Bit Rates**
- **End of Transmission Interrupt Flag**
- **Write Collision Flag Protection**
- **Wake-up from Idle Mode**
- **Double Speed (CK/2) Master SPI Mode**

Figure 65. SPI Block Diagram[1]



Note: 1. Refer to Figure 1 on page 2, and Table 25 on page 58 for SPI pin placement.

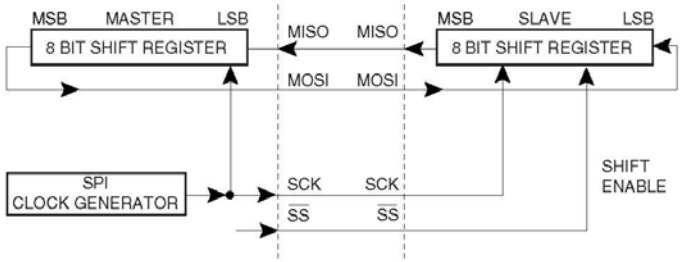
The interconnection between Master and Slave CPUs with SPI is shown in Figure 66. The system consists of two Shift Registers, and a Master clock generator. The SPI Master initiates the communication cycle when pulling low the Slave Select $\overline{SS}$ pin of the desired Slave. Master and Slave prepare the data to be sent in their respective Shift Registers, and the Master generates the required clock pulses on the SCK line to interchange data. Data is always shifted from Master to Slave on the Master Out – Slave In, MOSI, line, and from Slave to Master on the Master In – Slave Out, MISO, line. After each data packet, the Master will synchronize the Slave by pulling high the Slave Select, $\overline{SS}$, line.

When configured as a Master, the SPI interface has no automatic control of the $\overline{SS}$ line. This must be handled by user software before communication can start. When this is done, writing a

byte to the SPI Data Register starts the SPI clock generator, and the hardware shifts the eight bits into the Slave. After shifting one byte, the SPI clock generator stops, setting the end of Transmission Flag (SPIF). If the SPI Interrupt Enable bit (SPIE) in the SPCR Register is set, an interrupt is requested. The Master may continue to shift the next byte by writing it into SPDR, or signal the end of packet by pulling high the Slave Select, $\overline{SS}$ line. The last incoming byte will be kept in the Buffer Register for later use.

When configured as a Slave, the SPI interface will remain sleeping with MISO tri-stated as long as the $\overline{SS}$ pin is driven high. In this state, software may update the contents of the SPI Data Register, SPDR, but the data will not be shifted out by incoming clock pulses on the SCK pin until the $\overline{SS}$ pin is driven low. As one byte has been completely shifted, the end of Transmission Flag, SPIF is set. If the SPI Interrupt Enable bit, SPIE, in the SPCR Register is set, an interrupt is requested. The Slave may continue to place new data to be sent into SPDR before reading the incoming data. The last incoming byte will be kept in the Buffer Register for later use.

Figure 66. SPI Master-Slave Interconnection



The system is single buffered in the transmit direction and double buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received character must be read from the SPI Data Register before the next character has been completely shifted in. Otherwise, the first byte is lost.

In SPI Slave mode, the control logic will sample the incoming signal of the SCK pin. To ensure correct sampling of the clock signal, the minimum low and high periods should be:

Low periods: Longer than 2 CPU clock cycles.

High periods: Longer than 2 CPU clock cycles.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK, and $\overline{SS}$ pins is overridden according to Table 55 on page 136. For more details on automatic port overrides, refer to "Alternate Port Functions" on page 55.

Table 55. SPI Pin Overrides

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

Note: See "Alternate Functions of Port B" on page 58 for a detailed description of how to define the direction of the user defined SPI pins.

The following code examples show how to initialize the SPI as a Master and how to perform a simple transmission. DDR_SPI in the examples must be replaced by the actual Data Direction Register controlling the SPI pins. DD_MOSI, DD_MISO and DD_SCK must be replaced by the actual data direction bits for these pins. For example if MOSI is placed on pin PB5, replace DD_MOSI with DDB5 and DDR_SPI with DDRB.

Assembly Code Example[1]

```
SPI_MasterInit:
  ; Set MOSI and SCK output, all others input
  ldi   r17,(1<<DD_MOSI)|(1<<DD_SCK)
  out   DDR_SPI,r17
  ; Enable SPI, Master, set clock rate fck/16
  ldi   r17,(1<<SPE)|(1<<MSTR)|(1<<SPR0)
  out   SPCR,r17
  ret

SPI_MasterTransmit:
  ; Start transmission of data (r16)
  out   SPDR,r16
Wait_Transmit:
  ; Wait for transmission complete
  sbis  SPSR,SPIF
  rjmp  Wait_Transmit
  ret
```

C Code Example[1]

```
void SPI_MasterInit(void)
{
  /* Set MOSI and SCK output, all others input */
  DDR_SPI = (1<<DD_MOSI)|(1<<DD_SCK);
  /* Enable SPI, Master, set clock rate fck/16 */
  SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);
}

void SPI_MasterTransmit(char cData)
{
  /* Start transmission */
  SPDR = cData;
  /* Wait for transmission complete */
  while(!(SPSR & (1<<SPIF)))
    ;
}
```

Note: 1. See "About Code Examples" on page 7.

The following code examples show how to initialize the SPI as a Slave and how to perform a simple reception.

Assembly Code Example[1]

```
SPI_SlaveInit:
  ; Set MISO output, all others input
  ldi   r17,(1<<DD_MISO)
  out   DDR_SPI,r17
  ; Enable SPI
  ldi   r17,(1<<SPE)
  out   SPCR,r17
  ret

SPI_SlaveReceive:
  ; Wait for reception complete
  sbis  SPSR,SPIF
  rjmp  SPI_SlaveReceive
  ; Read received data and return
  in    r16,SPDR
  ret
```

C Code Example[1]

```
void SPI_SlaveInit(void)
{
  /* Set MISO output, all others input */
  DDR_SPI = (1<<DD_MISO);
  /* Enable SPI */
  SPCR = (1<<SPE);
}

char SPI_SlaveReceive(void)
{
  /* Wait for reception complete */
  while(!(SPSR & (1<<SPIF)))
    ;
  /* Return data register */
  return SPDR;
}
```

Note: 1. See "About Code Examples" on page 7.

$\overline{SS}$ Pin Functionality

Slave Mode

When the SPI is configured as a Slave, the Slave Select ($\overline{SS}$) pin is always input. When $\overline{SS}$ is held low, the SPI is activated, and MISO becomes an output if configured so by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs except MISO which can be user configured as an output, and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ pin is driven high.

The $\overline{SS}$ pin is useful for packet/byte synchronization to keep the Slave Bit Counter synchronous with the Master Clock generator. When the $\overline{SS}$ pin is driven high, the SPI Slave will immediately reset the send and receive logic, and drop any partially received data in the Shift Register.

Master Mode

When the SPI is configured as a Master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin.

If $\overline{SS}$ is configured as an output, the pin is a general output pin which does not affect the SPI system. Typically, the pin will be driving the $\overline{SS}$ pin of the SPI Slave.

If $\overline{SS}$ is configured as an input, it must be held high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as a Master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another Master selecting the SPI as a Slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a Slave. As a result of the SPI becoming a Slave, the MOSI and SCK pins become inputs.
2. The SPIF Flag in SPSR is set, and if the SPI interrupt is enabled, and the I-bit in SREG is set, the interrupt routine will be executed.

Thus, when interrupt-driven SPI transmission is used in Master mode, and there exists a possibility that $\overline{SS}$ is driven low, the interrupt should always check that the MSTR bit is still set. If the MSTR bit has been cleared by a Slave Select, it must be set by the user to re-enable SPI Master mode.

SPI Control Register – SPCR

Bit	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIE: SPI Interrupt Enable**

This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR Register is set and the if the global interrupt enable bit in SREG is set.

- **Bit 6 – SPE: SPI Enable**

When the SPE bit is written to one, the SPI is enabled. This bit must be set to enable any SPI operations.

- **Bit 5 – DORD: Data Order**

When the DORD bit is written to one, the LSB of the data word is transmitted first.

When the DORD bit is written to zero, the MSB of the data word is transmitted first.

- **Bit 4 – MSTR: Master/Slave Select**

This bit selects Master SPI mode when written to one, and Slave SPI mode when written logic zero. If $\overline{SS}$ is configured as an input and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI Master mode.

- **Bit 3 – CPOL: Clock Polarity**

When this bit is written to one, SCK is high when idle. When CPOL is written to zero, SCK is low when idle. Refer to Figure 67 and Figure 68 for an example. The CPOL functionality is summarized below:

Table 56. CPOL Functionality

CPOL	Leading Edge	Trailing Edge
0	Rising	Falling
1	Falling	Rising

- **Bit 2 – CPHA: Clock Phase**

The settings of the Clock Phase bit (CPHA) determine if data is sampled on the leading (first) or trailing (last) edge of SCK. Refer to Figure 67 and Figure 68 for an example. The CPHA functionality is summarized below:

Table 57. CPHA Functionality

CPHA	Leading Edge	Trailing Edge
0	Sample	Setup
1	Setup	Sample

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the Oscillator Clock frequency f_{osc} is shown in the following table:

Table 58. Relationship Between SCK and the Oscillator Frequency

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

SPI Status Register – SPSR

Bit	7	6	5	4	3	2	1	0	
	SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIF: SPI Interrupt Flag**

When a serial transfer is complete, the SPIF Flag is set. An interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. If $\overline{SS}$ is an input and is driven low when the SPI is in Master mode, this will also set the SPIF Flag. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI Status Register with SPIF set, then accessing the SPI Data Register (SPDR).

- **Bit 6 – WCOL: Write COLlision Flag**

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) are cleared by first reading the SPI Status Register with WCOL set, and then accessing the SPI Data Register.

- **Bit 5..1 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16 and will always read as zero.

- **Bit 0 – SPI2X: Double SPI Speed Bit**

When this bit is written logic one the SPI speed (SCK Frequency) will be doubled when the SPI is in Master mode (see Table 58). This means that the minimum SCK period will be two CPU clock periods. When the SPI is configured as Slave, the SPI is only guaranteed to work at $f_{osc}/4$ or lower.

The SPI interface on the ATmega16 is also used for program memory and EEPROM downloading or uploading. See page 273 for SPI Serial Programming and Verification.

SPI Data Register – SPDR

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

The SPI Data Register is a read/write register used for data transfer between the Register File and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 67 and Figure 68. Data bits are shifted out and latched in on opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize. This is clearly seen by summarizing Table 56 and Table 57, as done below:

Table 59. CPOL and CPHA Functionality

	Leading Edge	Trailing Edge	SPI Mode
CPOL = 0, CPHA = 0	Sample (Rising)	Setup (Falling)	0
CPOL = 0, CPHA = 1	Setup (Rising)	Sample (Falling)	1
CPOL = 1, CPHA = 0	Sample (Falling)	Setup (Rising)	2
CPOL = 1, CPHA = 1	Setup (Falling)	Sample (Rising)	3

Figure 67. SPI Transfer Format with CPHA = 0

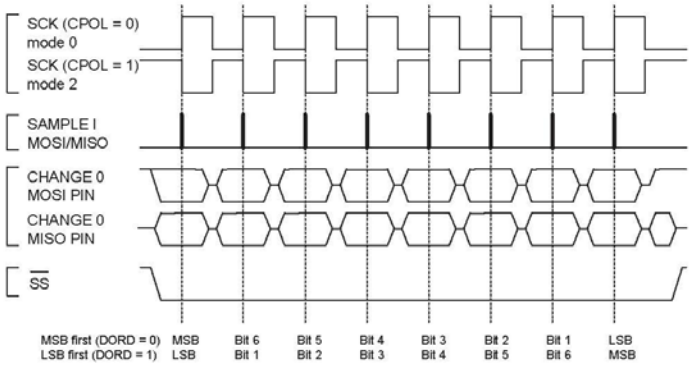
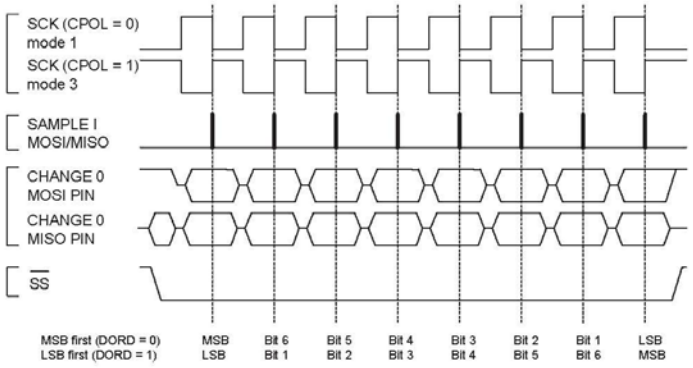


Figure 68. SPI Transfer Format with CPHA = 1



USART

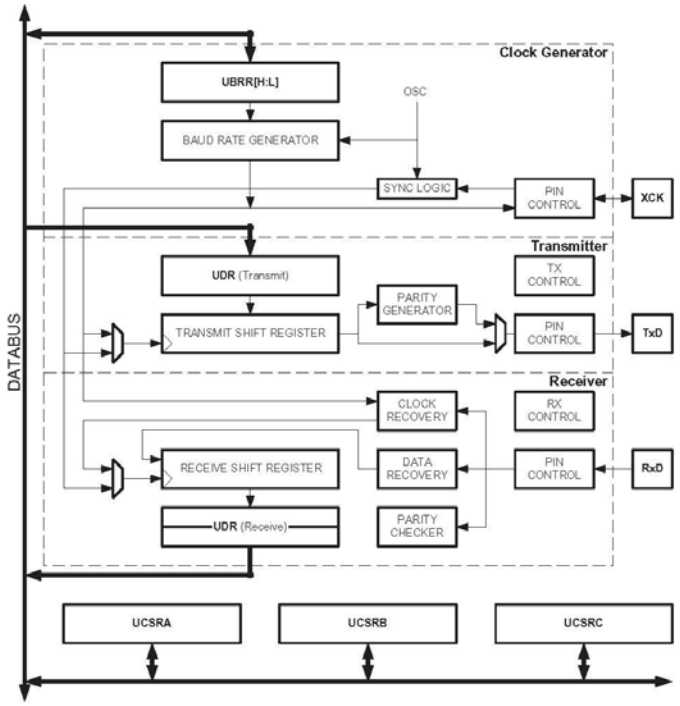
The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are:

- **Full Duplex Operation (Independent Serial Receive and Transmit Registers)**
- **Asynchronous or Synchronous Operation**
- **Master or Slave Clocked Synchronous Operation**
- **High Resolution Baud Rate Generator**
- **Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits**
- **Odd or Even Parity Generation and Parity Check Supported by Hardware**
- **Data OverRun Detection**
- **Framing Error Detection**
- **Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter**
- **Three Separate Interrupts on TX Complete, TX Data Register Empty, and RX Complete**
- **Multi-processor Communication Mode**
- **Double Speed Asynchronous Communication Mode**

Overview

A simplified block diagram of the USART transmitter is shown in Figure 69. CPU accessible I/O Registers and I/O pins are shown in bold.

Figure 69. USART Block Diagram[1]



Note: 1. Refer to Figure 1 on page 2, Table 33 on page 65, and Table 27 on page 60 for USART pin placement.

The dashed boxes in the block diagram separate the three main parts of the USART (listed from the top): Clock Generator, Transmitter and Receiver. Control Registers are shared by all units. The clock generation logic consists of synchronization logic for external clock input used by synchronous Slave operation, and the baud rate generator. The XCK (Transfer Clock) pin is only used by Synchronous Transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery units are used for asynchronous data reception. In addition to the recovery units, the receiver includes a parity checker, control logic, a Shift Register and a two level receive buffer (UDR). The receiver supports the same frame formats as the transmitter, and can detect frame error, data overrun and parity errors.

AVR USART vs. AVR UART – Compatibility

The USART is fully compatible with the AVR UART regarding:

- Bit locations inside all USART Registers
- Baud Rate Generation
- Transmitter Operation
- Transmit Buffer Functionality
- Receiver Operation

However, the receive buffering has two improvements that will affect the compatibility in some special cases:

- A second Buffer Register has been added. The two Buffer Registers operate as a circular FIFO buffer. Therefore the UDR must only be read once for each incoming data! More important is the fact that the Error Flags (FE and DOR) and the 9th data bit (RXB8) are buffered with the data in the receive buffer. Therefore the status bits must always be read before the UDR Register is read. Otherwise the error status will be lost since the buffer state is lost.
- The receiver Shift Register can now act as a third buffer level. This is done by allowing the received data to remain in the serial Shift Register (see Figure 69) if the Buffer Registers are full, until a new start bit is detected. The USART is therefore more resistant to Data OverRun (DOR) error conditions.

The following control bits have changed name, but have same functionality and register location:

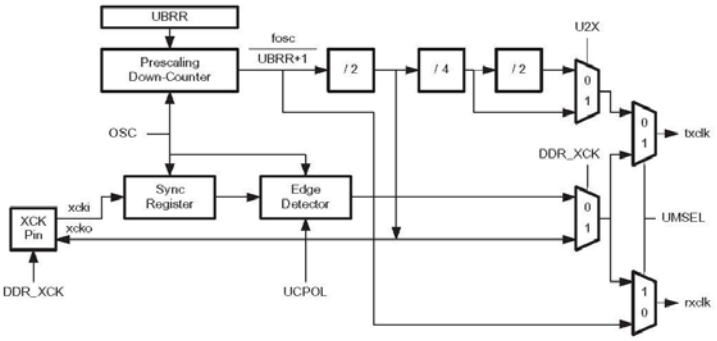
- CHR9 is changed to UCSZ2
- OR is changed to DOR

Clock Generation

The clock generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation: Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous mode. The UMSEL bit in USART Control and Status Register C (UCSRC) selects between asynchronous and synchronous operation. Double Speed (Asynchronous mode only) is controlled by the U2X found in the UCSRA Register. When using Synchronous mode (UMSEL = 1), the Data Direction Register for the XCK pin (DDR_XCK) controls whether the clock source is internal (Master mode) or external (Slave mode). The XCK pin is only active when using Synchronous mode.

Figure 70 shows a block diagram of the clock generation logic.

Figure 70. Clock Generation Logic, Block Diagram



Signal description:

txclk Transmitter clock (Internal Signal).

rxclk Receiver base clock (Internal Signal).

xcki Input from XCK pin (Internal Signal). Used for synchronous Slave operation.

xcko Clock output to XCK pin (Internal Signal). Used for synchronous Master operation.

fosc XTAL pin frequency (System Clock).

Internal Clock Generation – The Baud Rate Generator

Internal clock generation is used for the asynchronous and the synchronous Master modes of operation. The description in this section refers to Figure 70.

The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (fosc), is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output (= fosc/(UBRR+1)). The Transmitter divides the baud rate generator clock output by 2, 8 or 16 depending on mode. The baud rate generator output is used directly by the receiver's clock and data recovery units. However, the recovery units use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSEL, U2X and DDR_XCK bits.

Table 60 contains equations for calculating the baud rate (in bits per second) and for calculating the UBRR value for each mode of operation using an internally generated clock source.

Table 60. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second, bps)

f_{osc} System Oscillator clock frequency

UBRR Contents of the UBRRH and UBRRL Registers, (0 - 4095)

Some examples of UBRR values for some system clock frequencies are found in Table 68 (see page 168).

Double Speed Operation (U2X)

The transfer rate can be doubled by setting the U2X bit in UCSRA. Setting this bit only has effect for the asynchronous operation. Set this bit to zero when using synchronous operation.

Setting this bit will reduce the divisor of the baud rate divider from 16 to 8, effectively doubling the transfer rate for asynchronous communication. Note however that the receiver will in this case only use half the number of samples (reduced from 16 to 8) for data sampling and clock recovery, and therefore a more accurate baud rate setting and system clock are required when this mode is used. For the Transmitter, there are no downsides.

External Clock

External clocking is used by the synchronous Slave modes of operation. The description in this section refers to Figure 70 for details.

External clock input from the XCK pin is sampled by a synchronization register to minimize the chance of meta-stability. The output from the synchronization register must then pass through an edge detector before it can be used by the Transmitter and receiver. This process introduces a two CPU clock period delay and therefore the maximum external XCK clock frequency is limited by the following equation:

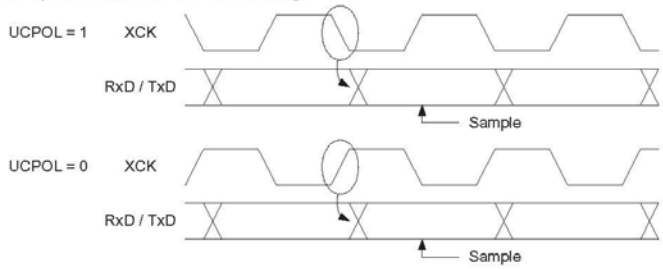
$$f_{XCK} < \frac{f_{OSC}}{4}$$

Note that f_{osc} depends on the stability of the system clock source. It is therefore recommended to add some margin to avoid possible loss of data due to frequency variations.

Synchronous Clock Operation

When Synchronous mode is used (UMSEL = 1), the XCK pin will be used as either clock input (Slave) or clock output (Master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input (on RxD) is sampled at the opposite XCK clock edge of the edge the data output (TxD) is changed.

Figure 71. Synchronous Mode XCK Timing.



The UCPOL bit UCRSC selects which XCK clock edge is used for data sampling and which is used for data change. As Figure 71 shows, when UCPOL is zero the data will be changed at rising XCK edge and sampled at falling XCK edge. If UCPOL is set, the data will be changed at falling XCK edge and sampled at rising XCK edge.

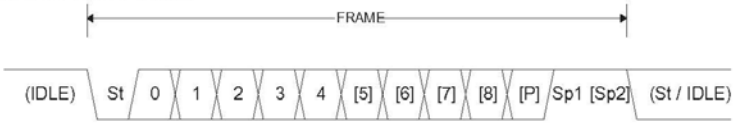
Frame Formats

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking. The USART accepts all 30 combinations of the following as valid frame formats:

- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit. Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit. If enabled, the parity bit is inserted after the data bits, before the stop bits. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state. Figure 72 illustrates the possible combinations of the frame formats. Bits inside brackets are optional.

Figure 72. Frame Formats



St Start bit, always low.

(n) Data bits (0 to 8).

P Parity bit. Can be odd or even.

Sp Stop bit, always high.

IDLE No transfers on the communication line (RxD or TxD). An IDLE line must be high.

The frame format used by the USART is set by the UCSZ2:0, UPM1:0, and USBS bits in UCSRB and UCSRC. The Receiver and Transmitter use the same setting. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter.

The USART Character SiZe (UCSZ2:0) bits select the number of data bits in the frame. The USART Parity mode (UPM1:0) bits enable and set the type of parity bit. The selection between one or two stop bits is done by the USART Stop Bit Select (USBS) bit. The receiver ignores the second stop bit. An FE (Frame Error) will therefore only be detected in the cases where the first stop bit is zero.

Parity Bit Calculation

The parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive or is inverted. The relation between the parity bit and data bits is as follows::

$$P_{even} = d_{n-1} \oplus \ldots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$
$$P_{odd} = d_{n-1} \oplus \ldots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1$$

P_{even} Parity bit using even parity

P_{odd} Parity bit using odd parity

d_n Data bit n of the character

If used, the parity bit is located between the last data bit and first stop bit of a serial frame.

USART Initialization

The USART has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and interrupts globally disabled) when doing the initialization.

Before doing a re-initialization with changed baud rate or frame format, be sure that there are no ongoing transmissions during the period the registers are changed. The TXC Flag can be used to check that the Transmitter has completed all transfers, and the RXC Flag can be used to check that there are no unread data in the receive buffer. Note that the TXC Flag must be cleared before each transmission (before UDR is written) if it is used for this purpose.

The following simple USART initialization code examples show one assembly and one C function that are equal in functionality. The examples assume asynchronous operation using polling (no interrupts enabled) and a fixed frame format. The baud rate is given as a function parameter. For the assembly code, the baud rate parameter is assumed to be stored in the r17:r16 registers. When the function writes to the UCSRC Register, the URSEL bit (MSB) must be set due to the sharing of I/O location by UBRRH and UCSRC.

Assembly Code Example[1]

```
USART_Init:
  ; Set baud rate
  out   UBRRH, r17
  out   UBRRL, r16
  ; Enable receiver and transmitter
  ldi   r16, (1<<RXEN)|(1<<TXEN)
  out   UCSRB,r16
  ; Set frame format: 8data, 2stop bit
  ldi   r16, (1<<URSEL)|(1<<USBS)|(3<<UCSZ0)
  out   UCSRC,r16
  ret
```

C Code Example[1]

```
#define FOSC 1843200// Clock Speed
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
void main( void )
{
  ...
  USART_Init ( MYUBRR );
  ...
}
void USART_Init( unsigned int ubrr)
{
  /* Set baud rate */
  UBRRH = (unsigned char)(ubrr>>8);
  UBRRL = (unsigned char)ubrr;
  /* Enable receiver and transmitter */
  UCSRB = (1<<RXEN)|(1<<TXEN);
  /* Set frame format: 8data, 2stop bit */
  UCSRC = (1<<URSEL)|(1<<USBS)|(3<<UCSZ0);
}
```

Note: 1. See "About Code Examples" on page 7.

More advanced initialization routines can be made that include frame format as parameters, disable interrupts and so on. However, many applications use a fixed setting of the Baud and Control Registers, and for these types of applications the initialization code can be placed directly in the main routine, or be combined with initialization code for other I/O modules.

Data Transmission – The USART Transmitter

The USART Transmitter is enabled by setting the *Transmit Enable* (TXEN) bit in the UCSRB Register. When the Transmitter is enabled, the normal port operation of the TxD pin is overridden by the USART and given the function as the transmitter's serial output. The baud rate, mode of operation and frame format must be set up once before doing any transmissions. If synchronous operation is used, the clock on the XCK pin will be overridden and used as transmission clock.

Sending Frames with 5 to 8 Data Bit

A data transmission is initiated by loading the transmit buffer with the data to be transmitted. The CPU can load the transmit buffer by writing to the UDR I/O location. The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is ready to send a new

frame. The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted. When the Shift Register is loaded with new data, it will transfer one complete frame at the rate given by the Baud Register, U2X bit or by XCK depending on mode of operation.

The following code examples show a simple USART transmit function based on polling of the *Data Register Empty* (UDRE) Flag. When using frames with less than eight bits, the most significant bits written to the UDR are ignored. The USART has to be initialized before the function can be used. For the assembly code, the data to be sent is assumed to be stored in Register R16

Assembly Code Example[1]

```
USART_Transmit:
  ; Wait for empty transmit buffer
  sbis  UCSRA,UDRE
  rjmp  USART_Transmit
  ; Put data (r16) into buffer, sends the data
  out   UDR,r16
  ret
```

C Code Example[1]

```
void USART_Transmit( unsigned char data )
{
  /* Wait for empty transmit buffer */
  while ( !( UCSRA & (1<<UDRE)) )
        ;
  /* Put data into buffer, sends the data */
  UDR = data;
}
```

Note: 1. See "About Code Examples" on page 7.

The function simply waits for the transmit buffer to be empty by checking the UDRE Flag, before loading it with new data to be transmitted. If the Data Register Empty Interrupt is utilized, the interrupt routine writes the data into the buffer.

Sending Frames with 9 Data Bit

If 9-bit characters are used (UCSZ = 7), the ninth bit must be written to the TXB8 bit in UCSRB before the Low byte of the character is written to UDR. The following code examples show a transmit function that handles 9-bit characters. For the assembly code, the data to be sent is assumed to be stored in Registers R17:R16.

Assembly Code Example[1]

```
USART_Transmit:
  ; Wait for empty transmit buffer
  sbis  UCSRA,UDRE
  rjmp  USART_Transmit
  ; Copy 9th bit from r17 to TXB8
  cbi   UCSRB,TXB8
  sbrc  r17,0
  sbi   UCSRB,TXB8
  ; Put LSB data (r16) into buffer, sends the data
  out   UDR,r16
  ret
```

C Code Example[1]

```
void USART_Transmit( unsigned int data )
{
  /* Wait for empty transmit buffer */
  while ( !( UCSRA & (1<<UDRE)) )
        ;
  /* Copy 9th bit to TXB8 */
  UCSRB &= ~(1<<TXB8);
  if ( data & 0x0100 )
    UCSRB |= (1<<TXB8);
  /* Put data into buffer, sends the data */
  UDR = data;
}
```

Note: 1. These transmit functions are written to be general functions. They can be optimized if the contents of the UCSRB is static. (that is, only the TXB8 bit of the UCSRB Register is used after initialization).

The ninth bit can be used for indicating an address frame when using multi processor communication mode or for other protocol handling as for example synchronization.

Transmitter Flags and Interrupts

The USART transmitter has two flags that indicate its state: USART Data Register Empty (UDRE) and Transmit Complete (TXC). Both flags can be used for generating interrupts.

The Data Register Empty (UDRE) Flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty, and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the Shift Register. For compatibility with future devices, always write this bit to zero when writing the UCSRA Register.

When the Data Register empty Interrupt Enable (UDRIE) bit in UCSRB is written to one, the USART Data Register Empty Interrupt will be executed as long as UDRE is set (provided that global interrupts are enabled). UDRE is cleared by writing UDR. When interrupt-driven data transmission is used, the Data Register Empty Interrupt routine must either write new data to UDR in order to clear UDRE or disable the Data Register empty Interrupt, otherwise a new interrupt will occur once the interrupt routine terminates.

The Transmit Complete (TXC) Flag bit is set one when the entire frame in the transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer. The TXC Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXC Flag is useful in half-duplex communication interfaces (like the RS485 standard), where a transmitting application must enter receive mode and free the communication bus immediately after completing the transmission.

When the Transmit Compete Interrupt Enable (TXCIE) bit in UCSRB is set, the USART Transmit Complete Interrupt will be executed when the TXC Flag becomes set (provided that global interrupts are enabled). When the transmit complete interrupt is used, the interrupt handling routine does not have to clear the TXC Flag, this is done automatically when the interrupt is executed.

Parity Generator

The parity generator calculates the parity bit for the serial frame data. When parity bit is enabled (UPM1 = 1), the transmitter control logic inserts the parity bit between the last data bit and the first stop bit of the frame that is sent.

Disabling the Transmitter

The disabling of the transmitter (setting the TXEN to zero) will not become effective until ongoing and pending transmissions are completed, that is, when the transmit Shift Register and transmit Buffer Register do not contain data to be transmitted. When disabled, the transmitter will no longer override the TxD pin.

Data Reception – The USART Receiver

The USART Receiver is enabled by writing the Receive Enable (RXEN) bit in the UCSRB Register to one. When the receiver is enabled, the normal pin operation of the RxD pin is overridden by the USART and given the function as the receiver's serial input. The baud rate, mode of operation and frame format must be set up once before any serial reception can be done. If synchronous operation is used, the clock on the XCK pin will be used as transfer clock.

Receiving Frames with 5 to 8 Data Bits

The receiver starts data reception when it detects a valid start bit. Each bit that follows the start bit will be sampled at the baud rate or XCK clock, and shifted into the receive Shift Register until the first stop bit of a frame is received. A second stop bit will be ignored by the receiver. When the first stop bit is received, that is, a complete serial frame is present in the receive Shift Register, the contents of the Shift Register will be moved into the receive buffer. The receive buffer can then be read by reading the UDR I/O location.

The following code example shows a simple USART receive function based on polling of the Receive Complete (RXC) Flag. When using frames with less than eight bits the most significant bits of the data read from the UDR will be masked to zero. The USART has to be initialized before the function can be used.

Assembly Code Example[1]

```
USART_Receive:
  ; Wait for data to be received
  sbis  UCSRA, RXC
  rjmp  USART_Receive
  ; Get and return received data from buffer
  in    r16, UDR
  ret
```

C Code Example[1]

```
unsigned char USART_Receive( void )
{
  /* Wait for data to be received */
  while ( !(UCSRA & (1<<RXC)) )
        ;
  /* Get and return received data from buffer */
  return UDR;
}
```

Note: 1. See "About Code Examples" on page 7.

The function simply waits for data to be present in the receive buffer by checking the RXC Flag, before reading the buffer and returning the value.

Receiving Frames with 9 Databits

If 9 bit characters are used (UCSZ=7) the ninth bit must be read from the RXB8 bit in UCSRB **before** reading the low bits from the UDR. This rule applies to the FE, DOR and PE status Flags as well. Read status from UCSRA, then data from UDR. Reading the UDR I/O location will change the state of the receive buffer FIFO and consequently the TXB8, FE, DOR and PE bits, which all are stored in the FIFO, will change.

The following code example shows a simple USART receive function that handles both 9-bit characters and the status bits.

Assembly Code Example[1]

```
USART_Receive:
  ; Wait for data to be received
  sbis  UCSRA, RXC
  rjmp  USART_Receive
  ; Get status and 9th bit, then data from buffer
  in    r18, UCSRA
  in    r17, UCSRB
  in    r16, UDR
  ; If error, return -1
  andi  r18,(1<<FE)|(1<<DOR)|(1<<PE)
  breq  USART_ReceiveNoError
  ldi   r17, HIGH(-1)
  ldi   r16, LOW(-1)
USART_ReceiveNoError:
  ; Filter the 9th bit, then return
  lsr   r17
  andi  r17, 0x01
  ret
```

C Code Example[1]

```
unsigned int USART_Receive( void )
{
  unsigned char status, resh, resl;
  /* Wait for data to be received */
  while ( !(UCSRA & (1<<RXC)) )
        ;
  /* Get status and 9th bit, then data */
  /* from buffer */
  status = UCSRA;
  resh = UCSRB;
  resl = UDR;
  /* If error, return -1 */
  if ( status & (1<<FE)|(1<<DOR)|(1<<PE) )
    return -1;
  /* Filter the 9th bit, then return */
  resh = (resh >> 1) & 0x01;
  return ((resh << 8) | resl);
}
```

Note: 1. See "About Code Examples" on page 7.

The receive function example reads all the I/O Registers into the Register File before any computation is done. This gives an optimal receive buffer utilization since the buffer location read will be free to accept new data as early as possible.

Receive Compete Flag and Interrupt

The USART Receiver has one flag that indicates the receiver state.

The Receive Complete (RXC) Flag indicates if there are unread data present in the receive buffer. This flag is one when unread data exist in the receive buffer, and zero when the receive buffer is empty (that is, does not contain any unread data). If the receiver is disabled (RXEN = 0), the receive buffer will be flushed and consequently the RXC bit will become zero.

When the Receive Complete Interrupt Enable (RXCIE) in UCSRB is set, the USART Receive Complete Interrupt will be executed as long as the RXC Flag is set (provided that global interrupts are enabled). When interrupt-driven data reception is used, the receive complete routine must read the received data from UDR in order to clear the RXC Flag, otherwise a new interrupt will occur once the interrupt routine terminates.

Receiver Error Flags

The USART Receiver has three Error Flags: Frame Error (FE), Data OverRun (DOR) and Parity Error (PE). All can be accessed by reading UCSRA. Common for the Error Flags is that they are located in the receive buffer together with the frame for which they indicate the error status. Due to the buffering of the Error Flags, the UCSRA must be read before the receive buffer (UDR), since reading the UDR I/O location changes the buffer read location. Another equality for the Error Flags is that they can not be altered by software doing a write to the flag location. However, all flags must be set to zero when the UCSRA is written for upward compatibility of future USART implementations. None of the Error Flags can generate interrupts.

The Frame Error (FE) Flag indicates the state of the first stop bit of the next readable frame stored in the receive buffer. The FE Flag is zero when the stop bit was correctly read (as one), and the FE Flag will be one when the stop bit was incorrect (zero). This flag can be used for detecting out-of-sync conditions, detecting break conditions and protocol handling. The FE Flag is not affected by the setting of the USBS bit in UCSRC since the receiver ignores all, except for the first, stop bits. For compatibility with future devices, always set this bit to zero when writing to UCSRA.

The Data OverRun (DOR) Flag indicates data loss due to a receiver buffer full condition. A Data OverRun occurs when the receive buffer is full (two characters), it is a new character waiting in the receive Shift Register, and a new start bit is detected. If the DOR Flag is set there was one or more serial frame lost between the frame last read from UDR, and the next frame read from UDR. For compatibility with future devices, always write this bit to zero when writing to UCSRA. The DOR Flag is cleared when the frame received was successfully moved from the Shift Register to the receive buffer.

The Parity Error (PE) Flag indicates that the next frame in the receive buffer had a parity error when received. If parity check is not enabled the PE bit will always be read zero. For compatibility with future devices, always set this bit to zero when writing to UCSRA. For more details see "Parity Bit Calculation" on page 149 and "Parity Checker" on page 156.

Parity Checker

The Parity Checker is active when the high USART Parity mode (UPM1) bit is set. Type of parity check to be performed (odd or even) is selected by the UPM0 bit. When enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the parity bit from the serial frame. The result of the check is stored in the receive buffer together with the received data and stop bits. The Parity Error (PE) Flag can then be read by software to check if the frame had a parity error.

The PE bit is set if the next character that can be read from the receive buffer had a parity error when received and the parity checking was enabled at that point (UPM1 = 1). This bit is valid until the receive buffer (UDR) is read.

Disabling the Receiver

In contrast to the Transmitter, disabling of the Receiver will be immediate. Data from ongoing receptions will therefore be lost. When disabled (that is, the RXEN is set to zero) the Receiver will no longer override the normal function of the RxD port pin. The receiver buffer FIFO will be flushed when the receiver is disabled. Remaining data in the buffer will be lost

Flushing the Receive Buffer

The receiver buffer FIFO will be flushed when the Receiver is disabled, that is, the buffer will be emptied of its contents. Unread data will be lost. If the buffer has to be flushed during normal operation, due to for instance an error condition, read the UDR I/O location until the RXC Flag is cleared. The following code example shows how to flush the receive buffer.

Assembly Code Example[1]

```
USART_Flush:
  sbis  UCSRA, RXC
  ret
  in    r16, UDR
  rjmp  USART_Flush
```

C Code Example[1]

```
void USART_Flush( void )
{
  unsigned char dummy;
  while ( UCSRA & (1<<RXC) ) dummy = UDR;
}
```

Note: 1. See "About Code Examples" on page 7.

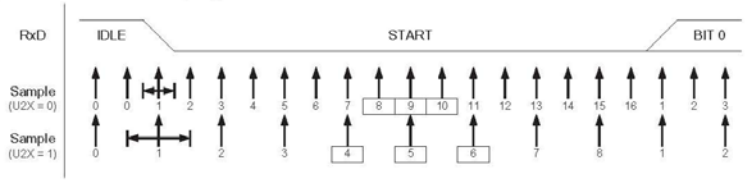
Asynchronous Data Reception

The USART includes a clock recovery and a data recovery unit for handling asynchronous data reception. The clock recovery logic is used for synchronizing the internally generated baud rate clock to the incoming asynchronous serial frames at the RxD pin. The data recovery logic samples and low pass filters each incoming bit, thereby improving the noise immunity of the receiver. The asynchronous reception operational range depends on the accuracy of the internal baud rate clock, the rate of the incoming frames, and the frame size in number of bits.

Asynchronous Clock Recovery

The clock recovery logic synchronizes internal clock to the incoming serial frames. Figure 73 illustrates the sampling process of the start bit of an incoming frame. The sample rate is 16 times the baud rate for Normal mode, and 8 times the baud rate for Double Speed mode. The horizontal arrows illustrate the synchronization variation due to the sampling process. Note the larger time variation when using the double speed mode (U2X = 1) of operation. Samples denoted zero are samples done when the RxD line is idle (that is, no communication activity).

Figure 73. Start Bit Sampling



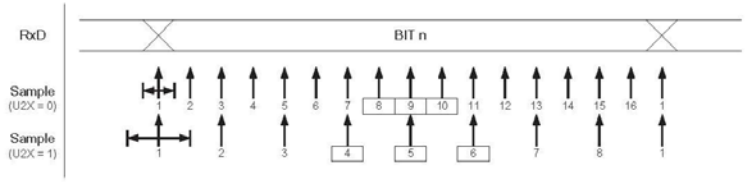
When the clock recovery logic detects a high (idle) to low (start) transition on the RxD line, the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample as shown in the figure. The clock recovery logic then uses samples 8, 9, and 10 for Normal mode, and samples 4, 5, and 6 for Double Speed mode (indicated with sample numbers inside boxes on the

figure), to decide if a valid start bit is received. If two or more of these three samples have logical high levels (the majority wins), the start bit is rejected as a noise spike and the receiver starts looking for the next high to low-transition. If however, a valid start bit is detected, the clock recovery logic is synchronized and the data recovery can begin. The synchronization process is repeated for each start bit.

Asynchronous Data Recovery

When the receiver clock is synchronized to the start bit, the data recovery can begin. The data recovery unit uses a state machine that has 16 states for each bit in normal mode and 8 states for each bit in Double Speed mode. Figure 74 shows the sampling of the data bits and the parity bit. Each of the samples is given a number that is equal to the state of the recovery unit.

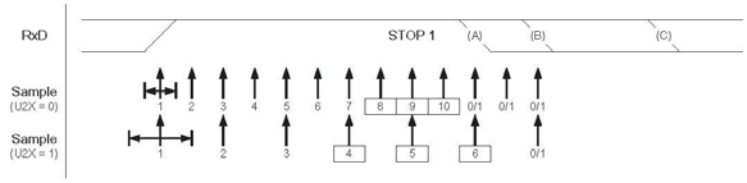
Figure 74. Sampling of Data and Parity Bit



The decision of the logic level of the received bit is taken by doing a majority voting of the logic value to the three samples in the center of the received bit. The center samples are emphasized on the figure by having the sample number inside boxes. The majority voting process is done as follows: If two or all three samples have high levels, the received bit is registered to be a logic 1. If two or all three samples have low levels, the received bit is registered to be a logic 0. This majority voting process acts as a low pass filter for the incoming signal on the RxD pin. The recovery process is then repeated until a complete frame is received. Including the first stop bit. Note that the receiver only uses the first stop bit of a frame.

Figure 75 shows the sampling of the stop bit and the earliest possible beginning of the start bit of the next frame.

Figure 75. Stop Bit Sampling and Next Start Bit Sampling



The same majority voting is done to the stop bit as done for the other bits in the frame. If the stop bit is registered to have a logic 0 value, the Frame Error (FE) Flag will be set.

A new high to low transition indicating the start bit of a new frame can come right after the last of the bits used for majority voting. For Normal Speed mode, the first low level sample can be at point marked (A) in Figure 75. For Double Speed mode the first low level must be delayed to (B). (C) marks a stop bit of full length. The early start bit detection influences the operational range of the receiver.

Asynchronous Operational Range

The operational range of the receiver is dependent on the mismatch between the received bit rate and the internally generated baud rate. If the Transmitter is sending frames at too fast or too slow bit rates, or the internally generated baud rate of the receiver does not have a similar (see Table 61) base frequency, the receiver will not be able to synchronize the frames to the start bit.

The following equations can be used to calculate the ratio of the incoming data rate and internal receiver baud rate.

$$R_{slow} = \frac{(D+1)S}{S-1+D\cdot S+S_F}$$

$$R_{fast} = \frac{(D+2)S}{(D+1)S+S_M}$$

D Sum of character size and parity size (D = 5 to 10 bit)

S Samples per bit. S = 16 for Normal Speed mode and S = 8 for Double Speed mode.

S_F First sample number used for majority voting. S_F = 8 for Normal Speed and S_F = 4 for Double Speed mode.

S_M Middle sample number used for majority voting. S_M = 9 for Normal Speed and S_M = 5 for Double Speed mode.

R_{slow} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate. R_{fast} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate.

Table 61 and Table 62 list the maximum receiver baud rate error that can be tolerated. Note that Normal Speed mode has higher toleration of baud rate variations.

Table 61. Recommended Maximum Receiver Baud Rate Error for Normal Speed Mode (U2X = 0)

D # (Data+Parity Bit)	R_{slow} (%)	R_{fast}(%)	Max Total Error (%)	Recommended Max Receiver Error (%)
5	93.20	106.67	+6.67/-6.8	±3.0
6	94.12	105.79	+5.79/-5.88	±2.5
7	94.81	105.11	+5.11/-5.19	±2.0
8	95.36	104.58	+4.58/-4.54	±2.0
9	95.81	104.14	+4.14/-4.19	±1.5
10	96.17	103.78	+3.78/-3.83	±1.5

Table 62. Recommended Maximum Receiver Baud Rate Error for Double Speed Mode (U2X = 1)

D # (Data+Parity Bit)	R_{slow} (%)	R_{fast} (%)	Max Total Error (%)	Recommended Max Receiver Error (%)
5	94.12	105.66	+5.66/-5.88	±2.5
6	94.92	104.92	+4.92/-5.08	±2.0
7	95.52	104.35	+4.35/-4.48	±1.5

Table 62. Recommended Maximum Receiver Baud Rate Error for Double Speed Mode (U2X = 1)

D # (Data+Parity Bit)	R_{slow} (%)	R_{fast} (%)	Max Total Error (%)	Recommended Max Receiver Error (%)
8	96.00	103.90	+3.90/-4.00	±1.5
9	96.39	103.53	+3.53/-3.61	±1.5
10	96.70	103.23	+3.23/-3.30	±1.0

The recommendations of the maximum receiver baud rate error was made under the assumption that the receiver and transmitter equally divides the maximum total error.

There are two possible sources for the receivers baud rate error. The receiver's system clock (XTAL) will always have some minor instability over the supply voltage range and the temperature range. When using a crystal to generate the system clock, this is rarely a problem, but for a resonator the system clock may differ more than 2% depending of the resonators tolerance. The second source for the error is more controllable. The baud rate generator can not always do an exact division of the system frequency to get the baud rate wanted. In this case an UBRR value that gives an acceptable low error can be used if possible.

Multi-processor Communication Mode

Setting the Multi-processor Communication mode (MPCM) bit in UCSRA enables a filtering function of incoming frames received by the USART Receiver. Frames that do not contain address information will be ignored and not put into the receive buffer. This effectively reduces the number of incoming frames that has to be handled by the CPU, in a system with multiple MCUs that communicate via the same serial bus. The Transmitter is unaffected by the MPCM setting, but has to be used differently when it is a part of a system utilizing the Multi-processor Communication mode.

If the receiver is set up to receive frames that contain 5 to 8 data bits, then the first stop bit indicates if the frame contains data or address information. If the receiver is set up for frames with nine data bits, then the ninth bit (RXB8) is used for identifying address and data frames. When the frame type bit (the first stop or the ninth bit) is one, the frame contains an address. When the frame type bit is zero the frame is a data frame.

The Multi-processor Communication mode enables several Slave MCUs to receive data from a Master MCU. This is done by first decoding an address frame to find out which MCU has been addressed. If a particular Slave MCU has been addressed, it will receive the following data frames as normal, while the other Slave MCUs will ignore the received frames until another address frame is received.

Using MPCM

For an MCU to act as a Master MCU, it can use a 9-bit character frame format (UCSZ = 7). The ninth bit (TXB8) must be set when an address frame (TXB8 = 1) or cleared when a data frame (TXB = 0) is being transmitted. The Slave MCUs must in this case be set to use a 9-bit character frame format.

The following procedure should be used to exchange data in Multi-processor Communication mode:

1. All Slave MCUs are in Multi-processor Communication mode (MPCM in UCSRA is set).
2. The Master MCU sends an address frame, and all Slaves receive and read this frame. In the Slave MCUs, the RXC Flag in UCSRA will be set as normal.
3. Each Slave MCU reads the UDR Register and determines if it has been selected. If so, it clears the MPCM bit in UCSRA, otherwise it waits for the next address byte and keeps the MPCM setting.
4. The addressed MCU will receive all data frames until a new address frame is received. The other Slave MCUs, which still have the MPCM bit set, will ignore the data frames.
5. When the last data frame is received by the addressed MCU, the addressed MCU sets the MPCM bit and waits for a new address frame from Master. The process then repeats from 2.

Using any of the 5-bit to 8-bit character frame formats is possible, but impractical since the receiver must change between using n and n+1 character frame formats. This makes full-duplex operation difficult since the transmitter and receiver uses the same character size setting. If 5-bit to 8-bit character frames are used, the transmitter must be set to use two stop bit (USBS = 1) since the first stop bit is used for indicating the frame type.

Do not use Read-Modify-Write instructions (SBI and CBI) to set or clear the MPCM bit. The MPCM bit shares the same I/O location as the TXC Flag and this might accidentally be cleared when using SBI or CBI instructions.

Accessing UBRRH/ UCSRC Registers

The UBRRH Register shares the same I/O location as the UCSRC Register. Therefore some special consideration must be taken when accessing this I/O location.

Write Access

When doing a write access of this I/O location, the high bit of the value written, the USART Register Select (URSEL) bit, controls which one of the two registers that will be written. If URSEL is zero during a write operation, the UBRRH value will be updated. If URSEL is one, the UCSRC setting will be updated.

The following code examples show how to access the two registers.

Assembly Code Example[1]

```
...
; Set UBRRH to 2
ldi r16,0x02
out UBRRH,r16
...
; Set the USBS and the UCSZ1 bit to one, and
; the remaining bits to zero.
ldi r16,(1<<URSEL)|(1<<USBS)|(1<<UCSZ1)
out UCSRC,r16
...
```

C Code Example[1]

```
...
/* Set UBRRH to 2 */
UBRRH = 0x02;
...
/* Set the USBS and the UCSZ1 bit to one, and */
/* the remaining bits to zero. */
UCSRC = (1<<URSEL)|(1<<USBS)|(1<<UCSZ1);
...
```

Note: 1. See "About Code Examples" on page 7.

As the code examples illustrate, write accesses of the two registers are relatively unaffected of the sharing of I/O location.

Read Access

Doing a read access to the UBRRH or the UCSRC Register is a more complex operation. However, in most applications, it is rarely necessary to read any of these registers.

The read access is controlled by a timed sequence. Reading the I/O location once returns the UBRRH Register contents. If the register location was read in previous system clock cycle, reading the register in the current clock cycle will return the UCSRC contents. Note that the timed sequence for reading the UCSRC is an atomic operation. Interrupts must therefore be controlled (for example by disabling interrupts globally) during the read operation.

The following code example shows how to read the UCSRC Register contents.

Assembly Code Example[1]

```
USART_ReadUCSRC:
  ; Read UCSRC
  in  r16,UBRRH
  in  r16,UCSRC
  ret
```

C Code Example[1]

```
unsigned char USART_ReadUCSRC( void )
{
  unsigned char ucsrc;
  /* Read UCSRC */
  ucsrc = UBRRH;
  ucsrc = UCSRC;
  return ucsrc;
}
```

Note: 1. See "About Code Examples" on page 7.

The assembly code example returns the UCSRC value in r16.

Reading the UBRRH contents is not an atomic operation and therefore it can be read as an ordinary register, as long as the previous instruction did not access the register location.

USART Register Description

USART I/O Data Register – UDR

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The USART Transmit Data Buffer Register and USART Receive Data Buffer Registers share the same I/O address referred to as USART Data Register or UDR. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDR Register location. Reading the UDR Register location will return the contents of the Receive Data Buffer Register (RXB).

For 5-bit, 6-bit, or 7-bit characters the upper unused bits will be ignored by the Transmitter and set to zero by the Receiver.

The transmit buffer can only be written when the UDRE Flag in the UCSRA Register is set. Data written to UDR when the UDRE Flag is not set, will be ignored by the USART Transmitter. When data is written to the transmit buffer, and the Transmitter is enabled, the Transmitter will load the data into the transmit Shift Register when the Shift Register is empty. Then the data will be serially transmitted on the TxD pin.

The receive buffer consists of a two level FIFO. The FIFO will change its state whenever the receive buffer is accessed. Due to this behavior of the receive buffer, do not use read modify write instructions (SBI and CBI) on this location. Be careful when using bit test instructions (SBIC and SBIS), since these also will change the state of the FIFO.

USART Control and Status Register A – UCSRA

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXC: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (that is, does not contain any unread data). If the receiver is disabled, the receive buffer will be flushed and consequently the RXC bit will become zero. The RXC Flag can be used to generate a Receive Complete interrupt (see description of the RXCIE bit).

- **Bit 6 – TXC: USART Transmit Complete**

This flag bit is set when the entire frame in the transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDR). The TXC Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXC Flag can generate a Transmit Complete interrupt (see description of the TXCIE bit).

- **Bit 5 – UDRE: USART Data Register Empty**

The UDRE Flag indicates if the transmit buffer (UDR) is ready to receive new data. If UDRE is one, the buffer is empty, and therefore ready to be written. The UDRE Flag can generate a Data Register empty Interrupt (see description of the UDRIE bit).

UDRE is set after a reset to indicate that the transmitter is ready.

- **Bit 4 – FE: Frame Error**

This bit is set if the next character in the receive buffer had a Frame Error when received. that is, when the first stop bit of the next character in the receive buffer is zero. This bit is valid until the receive buffer (UDR) is read. The FE bit is zero when the stop bit of received data is one. Always set this bit to zero when writing to UCSRA.

- **Bit 3 – DOR: Data OverRun**

This bit is set if a Data OverRun condition is detected. A Data OverRun occurs when the receive buffer is full (two characters), it is a new character waiting in the receive Shift Register, and a new start bit is detected. This bit is valid until the receive buffer (UDR) is read. Always set this bit to zero when writing to UCSRA.

- **Bit 2 – PE: Parity Error**

This bit is set if the next character in the receive buffer had a Parity Error when received and the parity checking was enabled at that point (UPM1 = 1). This bit is valid until the receive buffer (UDR) is read. Always set this bit to zero when writing to UCSRA.

- **Bit 1 – U2X: Double the USART Transmission Speed**

This bit only has effect for the asynchronous operation. Write this bit to zero when using synchronous operation.

Writing this bit to one will reduce the divisor of the baud rate divider from 16 to 8 effectively doubling the transfer rate for asynchronous communication.

- **Bit 0 – MPCM: Multi-processor Communication Mode**

This bit enables the Multi-processor Communication mode. When the MPCM bit is written to one, all the incoming frames received by the USART receiver that do not contain address information will be ignored. The transmitter is unaffected by the MPCM setting. For more detailed information see "Multi-processor Communication Mode" on page 161.

USART Control and Status Register B – UCSRB

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIE: RX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the RXC Flag. A USART Receive Complete Interrupt will be generated only if the RXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXC bit in UCSRA is set.

- **Bit 6 – TXCIE: TX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the TXC Flag. A USART Transmit Complete Interrupt will be generated only if the TXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXC bit in UCSRA is set.

- **Bit 5 – UDRIE: USART Data Register Empty Interrupt Enable**

Writing this bit to one enables interrupt on the UDRE Flag. A Data Register Empty Interrupt will be generated only if the UDRIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDRE bit in UCSRA is set.

- **Bit 4 – RXEN: Receiver Enable**

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxD pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FE, DOR, and PE Flags.

- **Bit 3 – TXEN: Transmitter Enable**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxD pin when enabled. The disabling of the Transmitter (writing TXEN to zero) will not become effective until ongoing and pending transmissions are completed, that is, when the transmit Shift Register and transmit Buffer Register do not contain data to be transmitted. When disabled, the transmitter will no longer override the TxD port.

- **Bit 2 – UCSZ2: Character Size**

The UCSZ2 bits combined with the UCSZ1:0 bit in UCSRC sets the number of data bits (Character Size) in a frame the receiver and transmitter use.

- **Bit 1 – RXB8: Receive Data Bit 8**

RXB8 is the ninth data bit of the received character when operating with serial frames with nine data bits. Must be read before reading the low bits from UDR.

- **Bit 0 – TXB8: Transmit Data Bit 8**

TXB8 is the ninth data bit in the character to be transmitted when operating with serial frames with nine data bits. Must be written before writing the low bits to UDR.

USART Control and Status Register C – UCSRC

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

The UCSRC Register shares the same I/O location as the UBRRH Register. See the "Accessing UBRRH/ UCSRC Registers" on page 162 section which describes how to access this register.

- **Bit 7 – URSEL: Register Select**

This bit selects between accessing the UCSRC or the UBRRH Register. It is read as one when reading UCSRC. The URSEL must be one when writing the UCSRC.

- **Bit 6 – UMSEL: USART Mode Select**

This bit selects between Asynchronous and Synchronous mode of operation.

Table 63. UMSEL Bit Settings

UMSEL	Mode
0	Asynchronous Operation
1	Synchronous Operation

- **Bit 5:4 – UPM1:0: Parity Mode**

These bits enable and set type of parity generation and check. If enabled, the transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The Receiver will generate a parity value for the incoming data and compare it to the UPM0 setting. If a mismatch is detected, the PE Flag in UCSRA will be set.

Table 64. UPM Bits Settings

UPM1	UPM0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

- **Bit 3 – USBS: Stop Bit Select**

This bit selects the number of Stop Bits to be inserted by the Transmitter. The Receiver ignores this setting.

Table 65. USBS Bit Settings

USBS	Stop Bit(s)
0	1-bit
1	2-bit

- **Bit 2:1 – UCSZ1:0: Character Size**

The UCSZ1:0 bits combined with the UCSZ2 bit in UCSRB sets the number of data bits (Character Size) in a frame the Receiver and Transmitter use.

Table 66. UCSZ Bits Settings

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

- **Bit 0 – UCPOL: Clock Polarity**

This bit is used for Synchronous mode only. Write this bit to zero when Asynchronous mode is used. The UCPOL bit sets the relationship between data output change and data input sample, and the synchronous clock (XCK).

Table 67. UCPOL Bit Settings

UCPOL	Transmitted Data Changed (Output of TxD Pin)	Received Data Sampled (Input on RxD Pin)
0	Rising XCK Edge	Falling XCK Edge
1	Falling XCK Edge	Rising XCK Edge

USART Baud Rate Registers – UBRRL and UBRRH

Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The UBRRH Register shares the same I/O location as the UCSRC Register. See the "Accessing UBRRH/ UCSRC Registers" on page 162 section which describes how to access this register.

- **Bit 15 – URSEL: Register Select**

This bit selects between accessing the UBRRH or the UCSRC Register. It is read as zero when reading UBRRH. The URSEL must be zero when writing the UBRRH.

- **Bit 14:12 – Reserved Bits**

These bits are reserved for future use. For compatibility with future devices, these bit must be written to zero when UBRRH is written.

- **Bit 11:0 – UBRR11:0: USART Baud Rate Register**

This is a 12-bit register which contains the USART baud rate. The UBRRH contains the four most significant bits, and the UBRRL contains the 8 least significant bits of the USART baud rate. Ongoing transmissions by the transmitter and receiver will be corrupted if the baud rate is changed. Writing UBRRL will trigger an immediate update of the baud rate prescaler.

Examples of Baud Rate Setting

For standard crystal and resonator frequencies, the most commonly used baud rates for asynchronous operation can be generated by using the UBRR settings in Table 68. UBRR values which yield an actual baud rate differing less than 0.5% from the target baud rate, are bold in the table. Higher error ratings are acceptable, but the receiver will have less noise resistance when the error ratings are high, especially for large serial frames (see "Asynchronous Operational Range" on page 159). The error values are calculated using the following equation:

$$\text{Error}[\%] = \left(\frac{\text{BaudRate}_{\text{Closest Match}}}{\text{BaudRate}} - 1\right) \bullet 100\%$$

Table 68. Examples of UBRR Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	f_{osc} = 1.0000 MHz				f_{osc} = 1.8432 MHz				f_{osc} = 2.0000 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4k	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%
19.2k	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8k	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4k	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6k	0	8.5%	1	8.5%	1	0.0%	3	0.0%	1	8.5%	3	8.5%
76.8k	–	–	1	-18.6%	1	-25.0%	2	0.0%	1	-18.6%	2	8.5%
115.2k	–	–	0	8.5%	0	0.0%	1	0.0%	0	8.5%	1	8.5%
230.4k	–	–	–	–	–	–	0	0.0%	–	–	–	–
250k	–	–	–	–	–	–	–	–	–	–	0	0.0%
Max (1)	62.5 Kbps		125 Kbps		115.2 Kbps		230.4 Kbps		125 Kbps		250 Kbps	

1. UBRR = 0, Error = 0.0%

Table 69. Examples of UBRR Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	f_{osc} = 3.6864 MHz				f_{osc} = 4.0000 MHz				f_{osc} = 7.3728 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	95	0.0%	191	0.0%	103	0.2%	207	0.2%	191	0.0%	383	0.0%
4800	47	0.0%	95	0.0%	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	23	0.0%	47	0.0%	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4k	15	0.0%	31	0.0%	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2k	11	0.0%	23	0.0%	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8k	7	0.0%	15	0.0%	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4k	5	0.0%	11	0.0%	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6k	3	0.0%	7	0.0%	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8k	2	0.0%	5	0.0%	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2k	1	0.0%	3	0.0%	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4k	0	0.0%	1	0.0%	0	8.5%	1	8.5%	1	0.0%	3	0.0%
250k	0	-7.8%	1	-7.8%	0	0.0%	1	0.0%	1	-7.8%	3	-7.8%
0.5M	–	–	0	-7.8%	–	–	0	0.0%	0	-7.8%	1	-7.8%
1M	–	–	–	–	–	–	–	–	–	–	0	-7.8%
Max (1)	230.4 Kbps		460.8 Kbps		250 Kbps		0.5 Mbps		460.8 Kbps		921.6 Kbps	

1. UBRR = 0, Error = 0.0%

Table 70. Examples of UBRR Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	f_{osc} = 8.0000 MHz				f_{osc} = 11.0592 MHz				f_{osc} = 14.7456 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	–	–	2	-7.8%	1	-7.8%	3	-7.8%
1M	–	–	0	0.0%	–	–	–	–	0	-7.8%	1	-7.8%
Max (1)	0.5 Mbps		1 Mbps		691.2 Kbps		1.3824 Mbps		921.6 Kbps		1.8432 Mbps	

1. UBRR = 0, Error = 0.0%

Table 71. Examples of UBRR Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	f_{osc} = 16.0000 MHz				f_{osc} = 18.4320 MHz				f_{osc} = 20.0000 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	–	–	4	-7.8%	–	–	4	0.0%
1M	0	0.0%	1	0.0%	–	–	–	–	–	–	–	–
Max (1)	1 Mbps		2 Mbps		1.152 Mbps		2.304 Mbps		1.25 Mbps		2.5 Mbps	

1. UBRR = 0, Error = 0.0%

Two-wire Serial Interface

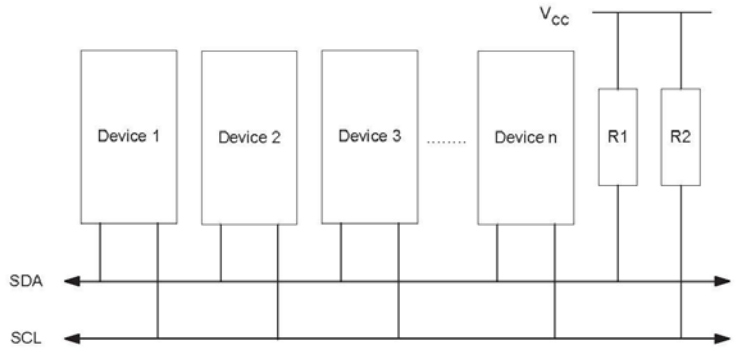
Features

- Simple Yet Powerful and Flexible Communication Interface, Only Two Bus Lines Needed
- Both Master and Slave Operation Supported
- Device Can Operate as Transmitter or Receiver
- 7-bit Address Space allows up to 128 Different Slave Addresses
- Multi-master Arbitration Support
- Up to 400 kHz Data Transfer Speed
- Slew-rate Limited Output Drivers
- Noise Suppression Circuitry Rejects Spikes on Bus Lines
- Fully Programmable Slave Address with General Call Support
- Address Recognition causes Wake-up when AVR is in Sleep Mode

Two-wire Serial Interface Bus Definition

The Two-wire Serial Interface (TWI) is ideally suited for typical microcontroller applications. The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA). The only external hardware needed to implement the bus is a single pull-up resistor for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the TWI protocol.

Figure 76. TWI Bus Interconnection



TWI Terminology

The following definitions are frequently encountered in this section.

Table 72. TWI Terminology

Term	Description
Master	The device that initiates and terminates a transmission. The Master also generates the SCL clock.
Slave	The device addressed by a Master.
Transmitter	The device placing data on the bus.
Receiver	The device reading data from the bus.

Electrical Interconnection

As depicted in Figure 76, both bus lines are connected to the positive supply voltage through pull-up resistors. The bus drivers of all TWI-compliant devices are open-drain or open-collector. This implements a wired-AND function which is essential to the operation of the interface. A low level on a TWI bus line is generated when one or more TWI devices output a zero. A high level is output when all TWI devices tri-state their outputs, allowing the pull-up resistors to pull the line high. Note that all AVR devices connected to the TWI bus must be powered in order to allow any bus operation.

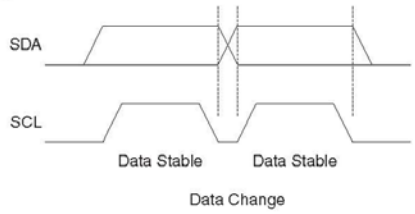
The number of devices that can be connected to the bus is only limited by the bus capacitance limit of 400 pF and the 7-bit Slave address space. A detailed specification of the electrical characteristics of the TWI is given in "Two-wire Serial Interface Characteristics" on page 294. Two different sets of specifications are presented there, one relevant for bus speeds below 100 kHz, and one valid for bus speeds up to 400 kHz.

Data Transfer and Frame Format

Transferring Bits

Each data bit transferred on the TWI bus is accompanied by a pulse on the clock line. The level of the data line must be stable when the clock line is high. The only exception to this rule is for generating start and stop conditions.

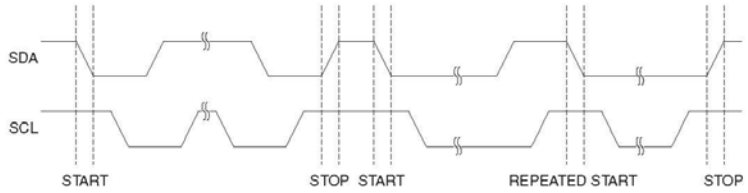
Figure 77. Data Validity



START and STOP Conditions

The Master initiates and terminates a data transmission. The transmission is initiated when the Master issues a START condition on the bus, and it is terminated when the Master issues a STOP condition. Between a START and a STOP condition, the bus is considered busy, and no other Master should try to seize control of the bus. A special case occurs when a new START condition is issued between a START and STOP condition. This is referred to as a REPEATED START condition, and is used when the Master wishes to initiate a new transfer without releasing control of the bus. After a REPEATED START, the bus is considered busy until the next STOP. This is identical to the START behavior, and therefore START is used to describe both START and REPEATED START for the remainder of this datasheet, unless otherwise noted. As depicted below, START and STOP conditions are signalled by changing the level of the SDA line when the SCL line is high.

Figure 78. START, REPEATED START, and STOP Conditions



Address Packet Format

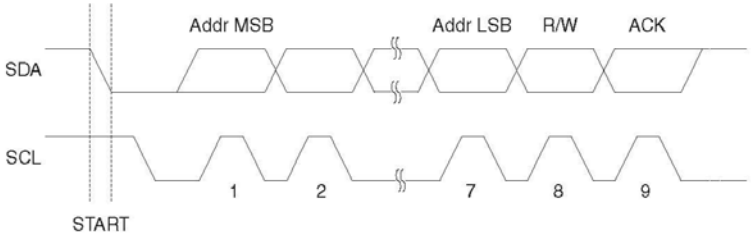
All address packets transmitted on the TWI bus are nine bits long, consisting of seven address bits, one READ/WRITE control bit and an acknowledge bit. If the READ/WRITE bit is set, a read operation is to be performed, otherwise a write operation should be performed. When a Slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle. If the addressed Slave is busy, or for some other reason can not service the Master's request, the SDA line should be left high in the ACK clock cycle. The Master can then transmit a STOP condition, or a REPEATED START condition to initiate a new transmission. An address packet consisting of a Slave address and a READ or a WRITE bit is called SLA+R or SLA+W, respectively.

The MSB of the address byte is transmitted first. Slave addresses can freely be allocated by the designer, but the address 0000 000 is reserved for a general call.

When a general call is issued, all Slaves should respond by pulling the SDA line low in the ACK cycle. A general call is used when a Master wishes to transmit the same message to several Slaves in the system. When the general call address followed by a Write bit is transmitted on the bus, all Slaves set up to acknowledge the general call will pull the SDA line low in the ack cycle. The following data packets will then be received by all the Slaves that acknowledged the general call. Note that transmitting the general call address followed by a Read bit is meaningless, as this would cause contention if several Slaves started transmitting different data.

All addresses of the format 1111 xxx should be reserved for future purposes.

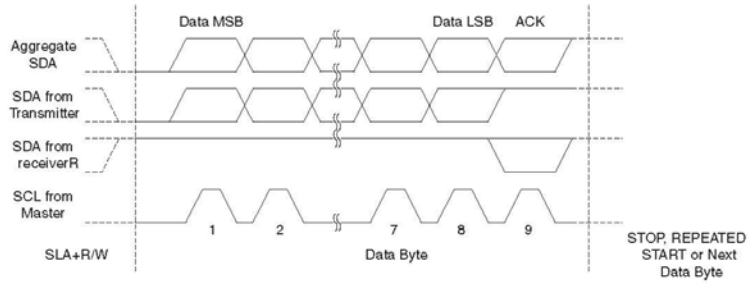
Figure 79. Address Packet Format



Data Packet Format

All data packets transmitted on the TWI bus are nine bits long, consisting of one data byte and an acknowledge bit. During a data transfer, the Master generates the clock and the START and STOP conditions, while the receiver is responsible for acknowledging the reception. An Acknowledge (ACK) is signalled by the receiver pulling the SDA line low during the ninth SCL cycle. If the receiver leaves the SDA line high, a NACK is signalled. When the receiver has received the last byte, or for some reason cannot receive any more bytes, it should inform the transmitter by sending a NACK after the final byte. The MSB of the data byte is transmitted first.

Figure 80. Data Packet Format

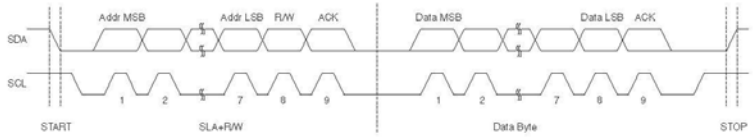


Combining Address and Data Packets into a Transmission

A transmission basically consists of a START condition, a SLA+R/W, one or more data packets and a STOP condition. An empty message, consisting of a START followed by a STOP condition, is illegal. Note that the wired-ANDing of the SCL line can be used to implement handshaking between the Master and the Slave. The Slave can extend the SCL low period by pulling the SCL line low. This is useful if the clock speed set up by the Master is too fast for the Slave, or the Slave needs extra time for processing between the data transmissions. The Slave extending the SCL low period will not affect the SCL high period, which is determined by the Master. As a consequence, the Slave can reduce the TWI data transfer speed by prolonging the SCL duty cycle.

Figure 81 shows a typical data transmission. Note that several data bytes can be transmitted between the SLA+R/W and the STOP condition, depending on the software protocol implemented by the application software.

Figure 81. Typical Data Transmission



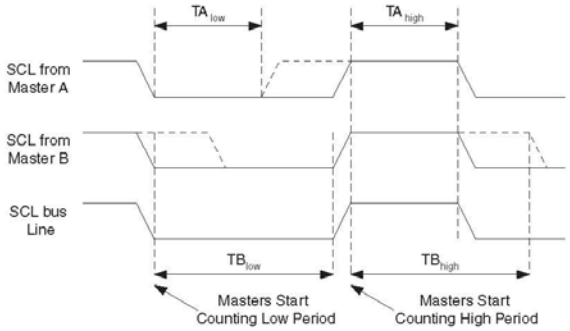
Multi-master Bus Systems, Arbitration and Synchronization

The TWI protocol allows bus systems with several Masters. Special concerns have been taken in order to ensure that transmissions will proceed as normal, even if two or more Masters initiate a transmission at the same time. Two problems arise in multi-master systems:

- An algorithm must be implemented allowing only one of the Masters to complete the transmission. All other Masters should cease transmission when they discover that they have lost the selection process. This selection process is called arbitration. When a contending Master discovers that it has lost the arbitration process, it should immediately switch to Slave mode to check whether it is being addressed by the winning Master. The fact that multiple Masters have started transmission at the same time should not be detectable to the Slaves, that is, the data being transferred on the bus must not be corrupted.
- Different Masters may use different SCL frequencies. A scheme must be devised to synchronize the serial clocks from all Masters, in order to let the transmission proceed in a lockstep fashion. This will facilitate the arbitration process.

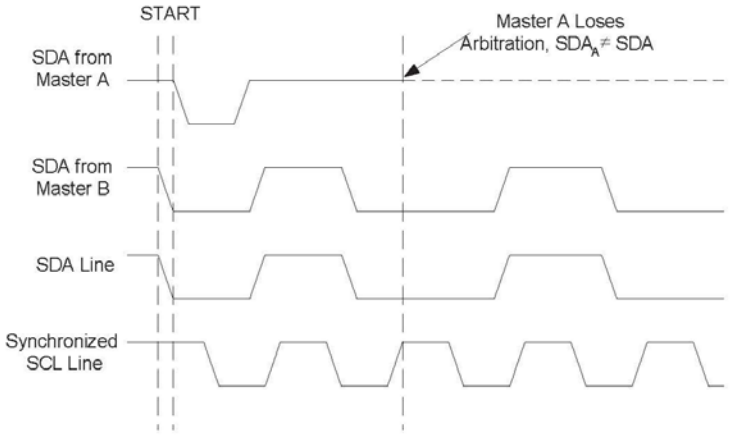
The wired-ANDing of the bus lines is used to solve both these problems. The serial clocks from all Masters will be wired-ANDed, yielding a combined clock with a high period equal to the one from the Master with the shortest high period. The low period of the combined clock is equal to the low period of the Master with the longest low period. Note that all Masters listen to the SCL line, effectively starting to count their SCL high and low time-out periods when the combined SCL line goes high or low, respectively.

Figure 82. SCL Synchronization between Multiple Masters



Arbitration is carried out by all Masters continuously monitoring the SDA line after outputting data. If the value read from the SDA line does not match the value the Master had output, it has lost the arbitration. Note that a Master can only lose arbitration when it outputs a high SDA value while another Master outputs a low value. The losing Master should immediately go to Slave mode, checking if it is being addressed by the winning Master. The SDA line should be left high, but losing Masters are allowed to generate a clock signal until the end of the current data or address packet. Arbitration will continue until only one Master remains, and this may take many bits. If several Masters are trying to address the same Slave, arbitration will continue into the data packet.

Figure 83. Arbitration between Two Masters



Note that arbitration is not allowed between:

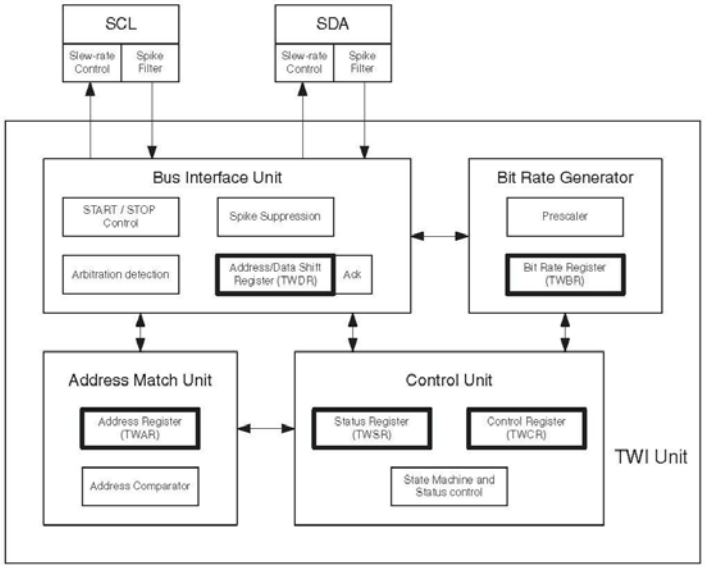
- A REPEATED START condition and a data bit
- A STOP condition and a data bit
- A REPEATED START and a STOP condition

It is the user software's responsibility to ensure that these illegal arbitration conditions never occur. This implies that in multi-master systems, all data transfers must use the same composition of SLA+R/W and data packets. In other words: All transmissions must contain the same number of data packets, otherwise the result of the arbitration is undefined.

Overview of the TWI Module

The TWI module is comprised of several submodules, as shown in Figure 84. All registers drawn in a thick line are accessible through the AVR data bus.

Figure 84. Overview of the TWI Module



SCL and SDA Pins

These pins interface the AVR TWI with the rest of the MCU system. The output drivers contain a slew-rate limiter in order to conform to the TWI specification. The input stages contain a spike suppression unit removing spikes shorter than 50 ns. Note that the internal pull-ups in the AVR pads can be enabled by setting the PORT bits corresponding to the SCL and SDA pins, as explained in the I/O Port section. The internal pull-ups can in some systems eliminate the need for external ones.

Bit Rate Generator Unit

This unit controls the period of SCL when operating in a Master mode. The SCL period is controlled by settings in the TWI Bit Rate Register (TWBR) and the Prescaler bits in the TWI Status Register (TWSR). Slave operation does not depend on Bit Rate or Prescaler settings, but the CPU clock frequency in the Slave must be at least 16 times higher than the SCL frequency. Note that Slaves may prolong the SCL low period, thereby reducing the average TWI bus clock period. The SCL frequency is generated according to the following equation:

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot 4^{TWPS}}$$

- TWBR = Value of the TWI Bit Rate Register
- TWPS = Value of the prescaler bits in the TWI Status Register

Note: Note: Pull-up resistor values should be selected according to the SCL frequency and the capacitive bus line load. See Table 120 on page 294 for value of pull-up resistor.

Bus Interface Unit

This unit contains the Data and Address Shift Register (TWDR), a START/STOP Controller and Arbitration detection hardware. The TWDR contains the address or data bytes to be transmitted,

or the address or data bytes received. In addition to the 8-bit TWDR, the Bus Interface Unit also contains a register containing the (N)ACK bit to be transmitted or received. This (N)ACK Register is not directly accessible by the application software. However, when receiving, it can be set or cleared by manipulating the TWI Control Register (TWCR). When in Transmitter mode, the value of the received (N)ACK bit can be determined by the value in the TWSR.

The START/STOP Controller is responsible for generation and detection of START, REPEATED START, and STOP conditions. The START/STOP controller is able to detect START and STOP conditions even when the AVR MCU is in one of the sleep modes, enabling the MCU to wake up if addressed by a Master.

If the TWI has initiated a transmission as Master, the Arbitration Detection hardware continuously monitors the transmission trying to determine if arbitration is in process. If the TWI has lost an arbitration, the Control Unit is informed. Correct action can then be taken and appropriate status codes generated.

Address Match Unit

The Address Match unit checks if received address bytes match the 7-bit address in the TWI Address Register (TWAR). If the TWI General Call Recognition Enable (TWGCE) bit in the TWAR is written to one, all incoming address bits will also be compared against the General Call address. Upon an address match, the Control Unit is informed, allowing correct action to be taken. The TWI may or may not acknowledge its address, depending on settings in the TWCR. The Address Match unit is able to compare addresses even when the AVR MCU is in sleep mode, enabling the MCU to wake up if addressed by a Master.

Control Unit

The Control unit monitors the TWI bus and generates responses corresponding to settings in the TWI Control Register (TWCR). When an event requiring the attention of the application occurs on the TWI bus, the TWI Interrupt Flag (TWINT) is asserted. In the next clock cycle, the TWI Status Register (TWSR) is updated with a status code identifying the event. The TWSR only contains relevant status information when the TWI Interrupt Flag is asserted. At all other times, the TWSR contains a special status code indicating that no relevant status information is available. As long as the TWINT Flag is set, the SCL line is held low. This allows the application software to complete its tasks before allowing the TWI transmission to continue.

The TWINT Flag is set in the following situations:

- After the TWI has transmitted a START/REPEATED START condition
- After the TWI has transmitted SLA+R/W
- After the TWI has transmitted an address byte
- After the TWI has lost arbitration
- After the TWI has been addressed by own Slave address or general call
- After the TWI has received a data byte
- After a STOP or REPEATED START has been received while still addressed as a Slave.
- When a bus error has occurred due to an illegal START or STOP condition

TWI Register Description

TWI Bit Rate Register – TWBR

Bit	7	6	5	4	3	2	1	0	
	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0	TWBR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..0 – TWI Bit Rate Register**

TWBR selects the division factor for the bit rate generator. The bit rate generator is a frequency divider which generates the SCL clock frequency in the Master modes. See "Bit Rate Generator Unit" on page 178 for calculating bit rates.

TWI Control Register – TWCR

Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The TWCR is used to control the operation of the TWI. It is used to enable the TWI, to initiate a Master access by applying a START condition to the bus, to generate a receiver acknowledge, to generate a stop condition, and to control halting of the bus while the data to be written to the bus are written to the TWDR. It also indicates a write collision if data is attempted written to TWDR while the register is inaccessible.

- **Bit 7 – TWINT: TWI Interrupt Flag**

This bit is set by hardware when the TWI has finished its current job and expects application software response. If the I-bit in SREG and TWIE in TWCR are set, the MCU will jump to the TWI Interrupt Vector. While the TWINT Flag is set, the SCL low period is stretched. The TWINT Flag must be cleared by software by writing a logic one to it. Note that this flag is not automatically cleared by hardware when executing the interrupt routine. Also note that clearing this flag starts the operation of the TWI, so all accesses to the TWI Address Register (TWAR), TWI Status Register (TWSR), and TWI Data Register (TWDR) must be complete before clearing this flag.

- **Bit 6 – TWEA: TWI Enable Acknowledge Bit**

The TWEA bit controls the generation of the acknowledge pulse. If the TWEA bit is written to one, the ACK pulse is generated on the TWI bus if the following conditions are met:

1. The device's own Slave address has been received.
2. A general call has been received, while the TWGCE bit in the TWAR is set.
3. A data byte has been received in Master Receiver or Slave Receiver mode.

By writing the TWEA bit to zero, the device can be virtually disconnected from the Two-wire Serial Bus temporarily. Address recognition can then be resumed by writing the TWEA bit to one again.

- **Bit 5 – TWSTA: TWI START Condition Bit**

The application writes the TWSTA bit to one when it desires to become a Master on the Two-wire Serial Bus. The TWI hardware checks if the bus is available, and generates a START condition on the bus if it is free. However, if the bus is not free, the TWI waits until a STOP condition

is detected, and then generates a new START condition to claim the bus Master status. TWSTA must be cleared by software when the START condition has been transmitted.

- **Bit 4 – TWSTO: TWI STOP Condition Bit**

Writing the TWSTO bit to one in Master mode will generate a STOP condition on the Two-wire Serial Bus. When the STOP condition is executed on the bus, the TWSTO bit is cleared automatically. In Slave mode, setting the TWSTO bit can be used to recover from an error condition. This will not generate a STOP condition, but the TWI returns to a well-defined unaddressed Slave mode and releases the SCL and SDA lines to a high impedance state.

- **Bit 3 – TWWC: TWI Write Collision Flag**

The TWWC bit is set when attempting to write to the TWI Data Register – TWDR when TWINT is low. This flag is cleared by writing the TWDR Register when TWINT is high.

- **Bit 2 – TWEN: TWI Enable Bit**

The TWEN bit enables TWI operation and activates the TWI interface. When TWEN is written to one, the TWI takes control over the I/O pins connected to the SCL and SDA pins, enabling the slew-rate limiters and spike filters. If this bit is written to zero, the TWI is switched off and all TWI transmissions are terminated, regardless of any ongoing operation.

- **Bit 1 – Res: Reserved Bit**

This bit is a reserved bit and will always read as zero.

- **Bit 0 – TWIE: TWI Interrupt Enable**

When this bit is written to one, and the I-bit in SREG is set, the TWI interrupt request will be activated for as long as the TWINT Flag is high.

TWI Status Register – TWSR

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

- **Bits 7..3 – TWS: TWI Status**

These five bits reflect the status of the TWI logic and the Two-wire Serial Bus. The different status codes are described later in this section. Note that the value read from TWSR contains both the 5-bit status value and the 2-bit prescaler value. The application designer should mask the prescaler bits to zero when checking the Status bits. This makes status checking independent of prescaler setting. This approach is used in this datasheet, unless otherwise noted.

- **Bit 2 – Res: Reserved Bit**

This bit is reserved and will always read as zero.

- **Bits 1..0 – TWPS: TWI Prescaler Bits**

These bits can be read and written, and control the bit rate prescaler.

Table 73. TWI Bit Rate Prescaler

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

To calculate bit rates, see "Bit Rate Generator Unit" on page 178. The value of TWPS1..0 is used in the equation.

TWI Data Register – TWDR

Bit	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

In Transmit mode, TWDR contains the next byte to be transmitted. In Receive mode, the TWDR contains the last byte received. It is writable while the TWI is not in the process of shifting a byte. This occurs when the TWI Interrupt Flag (TWINT) is set by hardware. Note that the Data Register cannot be initialized by the user before the first interrupt occurs. The data in TWDR remains stable as long as TWINT is set. While data is shifted out, data on the bus is simultaneously shifted in. TWDR always contains the last byte present on the bus, except after a wake up from a sleep mode by the TWI interrupt. In this case, the contents of TWDR is undefined. In the case of a lost bus arbitration, no data is lost in the transition from Master to Slave. Handling of the ACK bit is controlled automatically by the TWI logic, the CPU cannot access the ACK bit directly.

- **Bits 7..0 – TWD: TWI Data Register**

These eight bits contain the next data byte to be transmitted, or the latest data byte received on the Two-wire Serial Bus.

TWI (Slave) Address Register – TWAR

Bit	7	6	5	4	3	2	1	0	
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	0	

The TWAR should be loaded with the 7-bit Slave address (in the seven most significant bits of TWAR) to which the TWI will respond when programmed as a Slave Transmitter or receiver. In multi-master systems, TWAR must be set in Masters which can be addressed as Slaves by other Masters.

The LSB of TWAR is used to enable recognition of the general call address ($00). There is an associated address comparator that looks for the Slave address (or general call address if enabled) in the received serial address. If a match is found, an interrupt request is generated.

- **Bits 7..1 – TWA: TWI (Slave) Address Register**

These seven bits constitute the Slave address of the TWI unit.

- **Bit 0 – TWGCE: TWI General Call Recognition Enable Bit**

If set, this bit enables the recognition of a General Call given over the Two-wire Serial Bus.

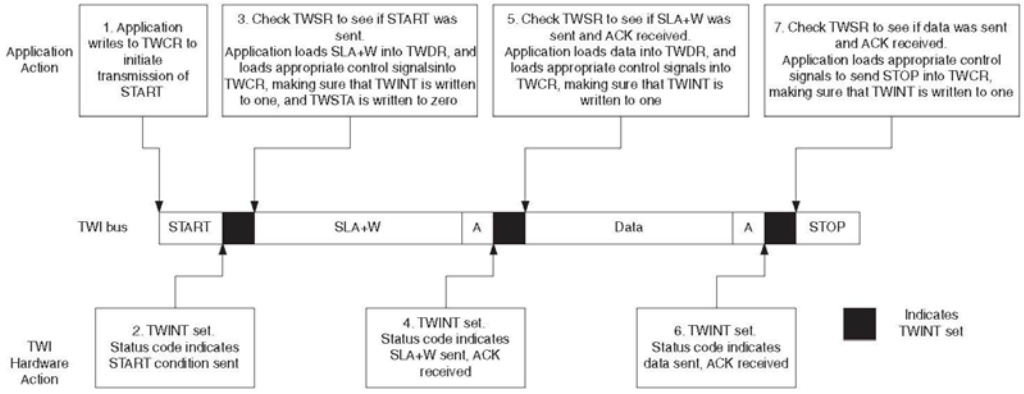
Using the TWI

The AVR TWI is byte-oriented and interrupt based. Interrupts are issued after all bus events, like reception of a byte or transmission of a START condition. Because the TWI is interrupt-based, the application software is free to carry on other operations during a TWI byte transfer. Note that the TWI Interrupt Enable (TWIE) bit in TWCR together with the Global Interrupt Enable bit in SREG allow the application to decide whether or not assertion of the TWINT Flag should generate an interrupt request. If the TWIE bit is cleared, the application must poll the TWINT Flag in order to detect actions on the TWI bus.

When the TWINT Flag is asserted, the TWI has finished an operation and awaits application response. In this case, the TWI Status Register (TWSR) contains a value indicating the current state of the TWI bus. The application software can then decide how the TWI should behave in the next TWI bus cycle by manipulating the TWCR and TWDR Registers.

Figure 85 is a simple example of how the application can interface to the TWI hardware. In this example, a Master wishes to transmit a single data byte to a Slave. This description is quite abstract, a more detailed explanation follows later in this section. A simple code example implementing the desired behavior is also presented.

Figure 85. Interfacing the Application to the TWI in a Typical Transmission



1. The first step in a TWI transmission is to transmit a START condition. This is done by writing a specific value into TWCR, instructing the TWI hardware to transmit a START condition. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the START condition.
2. When the START condition has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the START condition has successfully been sent.
3. The application software should now examine the value of TWSR, to make sure that the START condition was successfully transmitted. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load SLA+W into TWDR. Remember that TWDR is used both for address and data. After TWDR has been loaded with the

desired SLA+W, a specific value must be written to TWCR, instructing the TWI hardware to transmit the SLA+W present in TWDR. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the address packet.

4. When the address packet has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the address packet has successfully been sent. The status code will also reflect whether a Slave acknowledged the packet or not.
5. The application software should now examine the value of TWSR, to make sure that the address packet was successfully transmitted, and that the value of the ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load a data packet into TWDR. Subsequently, a specific value must be written to TWCR, instructing the TWI hardware to transmit the data packet present in TWDR. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the data packet.
6. When the data packet has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the data packet has successfully been sent. The status code will also reflect whether a Slave acknowledged the packet or not.
7. The application software should now examine the value of TWSR, to make sure that the data packet was successfully transmitted, and that the value of the ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must write a specific value to TWCR, instructing the TWI hardware to transmit a STOP condition. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the STOP condition. Note that TWINT is NOT set after a STOP condition has been sent.

Even though this example is simple, it shows the principles involved in all TWI transmissions. These can be summarized as follows:

- When the TWI has finished an operation and expects application response, the TWINT Flag is set. The SCL line is pulled low until TWINT is cleared.
- When the TWINT Flag is set, the user must update all TWI Registers with the value relevant for the next TWI bus cycle. As an example, TWDR must be loaded with the value to be transmitted in the next bus cycle.
- After all TWI Register updates and other pending application software tasks have been completed, TWCR is written. When writing TWCR, the TWINT bit should be set. Writing a one to TWINT clears the flag. The TWI will then commence executing whatever operation was specified by the TWCR setting.

In the following an assembly and C implementation of the example is given. Note that the code below assumes that several definitions have been made, for example by using include-files.

	Assembly code example	C example	Comments
1	`ldi r16, (1<<TWINT)|(1<<TWSTA)|(1<<TWEN)` `out TWCR, r16`	`TWCR = (1<<TWINT)|(1<<TWSTA)|(1<<TWEN)`	Send START condition

2	`wait1:`
	`in r16,TWCR`
	`sbrs r16,TWINT`
	`rjmp wait1`
		`while (!(TWCR & (1<<TWINT)))`
		`;`
			Wait for TWINT Flag set. This indicates that the START condition has been transmitted

3	`in r16,TWSR`
	`andi r16, 0xF8`
	`cpi r16, START`
	`brne ERROR`
		`if ((TWSR & 0xF8) != START)`
		`ERROR();`
			Check value of TWI Status Register. Mask prescaler bits. If status different from START go to ERROR

	`ldi r16, SLA_W`
	`out TWDR, r16`
	`ldi r16, (1<<TWINT) | (1<<TWEN)`
	`out TWCR, r16`
		`TWDR = SLA_W;`
		`TWCR = (1<<TWINT) | (1<<TWEN);`
			Load SLA_W into TWDR Register. Clear TWINT bit in TWCR to start transmission of address

4	`wait2:`
	`in r16,TWCR`
	`sbrs r16,TWINT`
	`rjmp wait2`
		`while (!(TWCR & (1<<TWINT)))`
		`;`
			Wait for TWINT Flag set. This indicates that the SLA+W has been transmitted, and ACK/NACK has been received.

5	`in r16,TWSR`
	`andi r16, 0xF8`
	`cpi r16, MT_SLA_ACK`
	`brne ERROR`
		`if ((TWSR & 0xF8) != MT_SLA_ACK)`
		`ERROR();`
			Check value of TWI Status Register. Mask prescaler bits. If status different from MT_SLA_ACK go to ERROR

	`ldi r16, DATA`
	`out TWDR, r16`
	`ldi r16, (1<<TWINT) | (1<<TWEN)`
	`out TWCR, r16`
		`TWDR = DATA;`
		`TWCR = (1<<TWINT) | (1<<TWEN);`
			Load DATA into TWDR Register. Clear TWINT bit in TWCR to start transmission of data

6	`wait3:`
	`in r16,TWCR`
	`sbrs r16,TWINT`
	`rjmp wait3`
		`while (!(TWCR & (1<<TWINT)))`
		`;`
			Wait for TWINT Flag set. This indicates that the DATA has been transmitted, and ACK/NACK has been received.

7	`in r16,TWSR`
	`andi r16, 0xF8`
	`cpi r16, MT_DATA_ACK`
	`brne ERROR`
		`if ((TWSR & 0xF8) != MT_DATA_ACK)`
		`ERROR();`
			Check value of TWI Status Register. Mask prescaler bits. If status different from MT_DATA_ACK go to ERROR

	`ldi r16, (1<<TWINT)|(1<<TWEN)|(1<<TWSTO)` `out TWCR, r16`	`TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTO);`	Transmit STOP condition

Transmission Modes

The TWI can operate in one of four major modes. These are named Master Transmitter (MT), Master Receiver (MR), Slave Transmitter (ST) and Slave Receiver (SR). Several of these modes can be used in the same application. As an example, the TWI can use MT mode to write data into a TWI EEPROM, MR mode to read the data back from the EEPROM. If other Masters are present in the system, some of these might transmit data to the TWI, and then SR mode would be used. It is the application software that decides which modes are legal.

The following sections describe each of these modes. Possible status codes are described along with figures detailing data transmission in each of the modes. These figures contain the following abbreviations:

S: START condition

Rs: REPEATED START condition

R: Read bit (high level at SDA)

W: Write bit (low level at SDA)

A: Acknowledge bit (low level at SDA)

$\overline{A}$: Not acknowledge bit (high level at SDA)

Data: 8-bit data byte

P: STOP condition

SLA: Slave Address

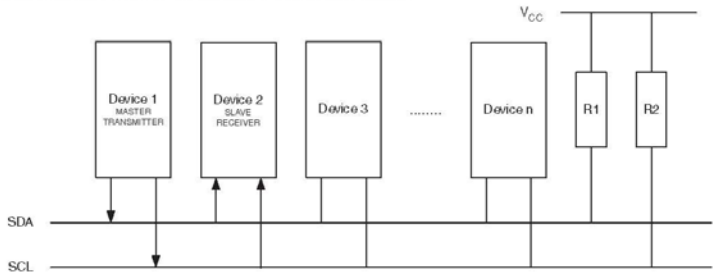
In Figure 87 to Figure 93, circles are used to indicate that the TWINT Flag is set. The numbers in the circles show the status code held in TWSR, with the prescaler bits masked to zero. At these points, actions must be taken by the application to continue or complete the TWI transfer. The TWI transfer is suspended until the TWINT Flag is cleared by software.

When the TWINT Flag is set, the status code in TWSR is used to determine the appropriate software action. For each status code, the required software action and details of the following serial transfer are given in Table 74 to Table 77. Note that the prescaler bits are masked to zero in these tables.

Master Transmitter Mode

In the Master Transmitter mode, a number of data bytes are transmitted to a Slave Receiver (see Figure 86). In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 86. Data Transfer in Master Transmitter Mode



A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

TWEN must be set to enable the Two-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be written to one to clear the TWINT Flag. The TWI will then test the Two-wire Serial Bus and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSR will be $08 (See Table 74). In order to enter MT mode, SLA+W must be transmitted. This is done by writing SLA+W to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

When SLA+W have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in Master mode are $18, $20, or $38. The appropriate action to be taken for each of these status codes is detailed in Table 74.

When SLA+W has been successfully transmitted, a data packet should be transmitted. This is done by writing the data byte to TWDR. TWDR must only be written when TWINT is high. If not, the access will be discarded, and the Write Collision bit (TWWC) will be set in the TWCR Register. After updating TWDR, the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

This scheme is repeated until the last byte has been sent and the transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	1	X	1	0	X

A REPEATED START condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

After a repeated START condition (state $10) the Two-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control of the bus.

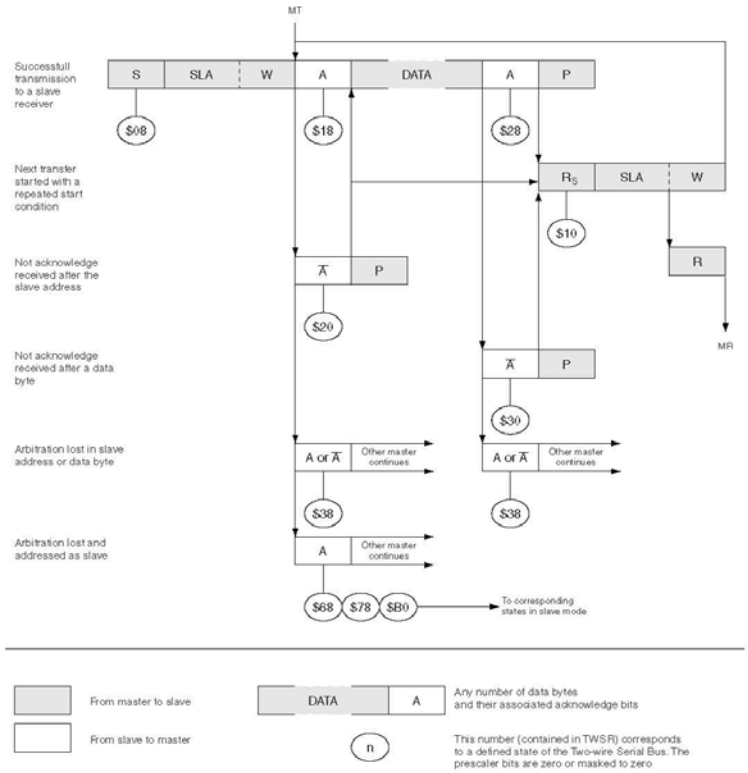
Table 74. Status Codes for Master Transmitter Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the Two-wire Serial Bus and Two-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
$08	A START condition has been transmitted	Load SLA+W	0	0	1	X	SLA+W will be transmitted; ACK or NOT ACK will be received
$10	A repeated START condition has been transmitted	Load SLA+W or	0	0	1	X	SLA+W will be transmitted; ACK or NOT ACK will be received
		Load SLA+R	0	0	1	X	SLA+R will be transmitted; Logic will switch to Master Receiver mode

Table 74. Status Codes for Master Transmitter Mode

$18	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received
		No TWDR action or	1	0	1	X	Repeated START will be transmitted
		No TWDR action or	0	1	1	X	STOP condition will be transmitted and TWSTO Flag will be Reset
		No TWDR action	1	1	1	X	STOP condition followed by a START condition will be transmitted and TWSTO Flag will be Reset
$20	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received
		No TWDR action or	1	0	1	X	Repeated START will be transmitted
		No TWDR action or	0	1	1	X	STOP condition will be transmitted and TWSTO Flag will be reset
		No TWDR action	1	1	1	X	STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
$28	Data byte has been transmitted; ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received
		No TWDR action or	1	0	1	X	Repeated START will be transmitted
		No TWDR action or	0	1	1	X	STOP condition will be transmitted and TWSTO Flag will be reset
		No TWDR action	1	1	1	X	STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
$30	Data byte has been transmitted; NOT ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be received
		No TWDR action or	1	0	1	X	Repeated START will be transmitted
		No TWDR action or	0	1	1	X	STOP condition will be transmitted and TWSTO Flag will be reset
		No TWDR action	1	1	1	X	STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
$38	Arbitration lost in SLA+W or data bytes	No TWDR action or	0	0	1	X	Two-wire Serial Bus will be released and not addressed Slave mode entered
		No TWDR action	1	0	1	X	A START condition will be transmitted when the bus becomes free

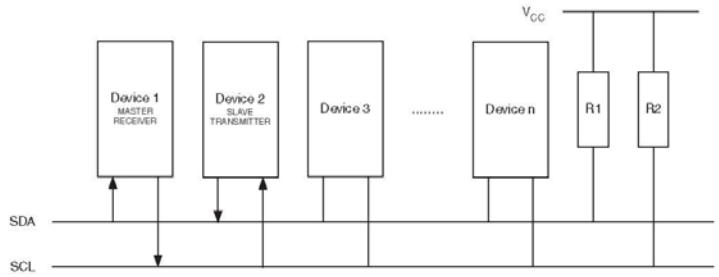
Figure 87. Formats and States in the Master Transmitter Mode



Master Receiver Mode

In the Master Receiver mode, a number of data bytes are received from a Slave Transmitter (see Figure 88). In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 88. Data Transfer in Master Receiver Mode



A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

TWEN must be written to one to enable the Two-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be set to clear the TWINT Flag. The TWI will then test the Two-wire Serial Bus and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSR will be $08 (See Table 74). In order to enter MR mode, SLA+R must be transmitted. This is done by writing SLA+R to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

When SLA+R have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in Master mode are $38, $40, or $48. The appropriate action to be taken for each of these status codes is detailed in Table 75. Received data can be read from the TWDR Register when the TWINT Flag is set high by hardware. This scheme is repeated until the last byte has been received. After the last byte has been received, the MR should inform the ST by sending a NACK after the last received data byte. The transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	1	X	1	0	X

A REPEATED START condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

After a repeated START condition (state $10) the Two-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control over the bus.

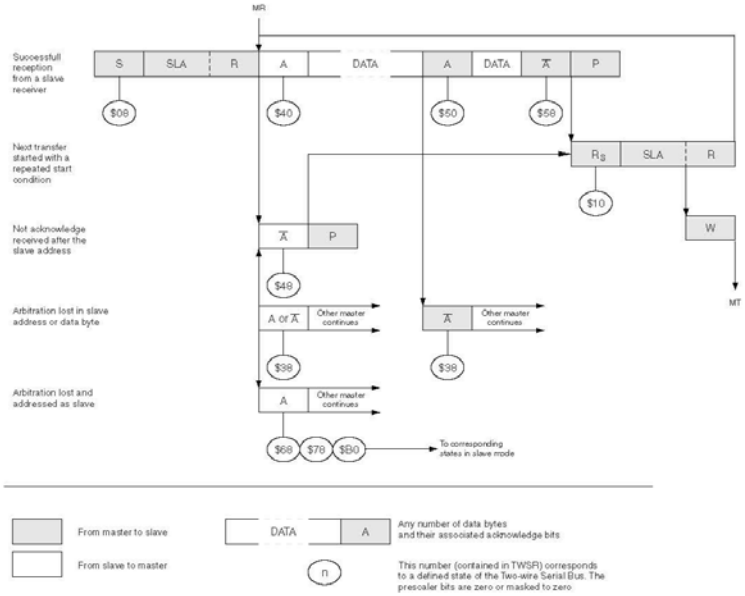
Table 75. Status Codes for Master Receiver Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the Two-wire Serial Bus and Two-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	

Table 75. Status Codes for Master Receiver Mode (Continued)

$08	A START condition has been transmitted	Load SLA+R	0	0	1	X	SLA+R will be transmitted ACK or NOT ACK will be received
$10	A repeated START condition has been transmitted	Load SLA+R or	0	0	1	X	SLA+R will be transmitted ACK or NOT ACK will be received
		Load SLA+W	0	0	1	X	SLA+W will be transmitted Logic will switch to masTer Transmitter mode
$38	Arbitration lost in SLA+R or NOT ACK bit	No TWDR action or	0	0	1	X	Two-wire Serial Bus will be released and not addressed Slave mode will be entered
		No TWDR action	1	0	1	X	A START condition will be transmitted when the bus becomes free
$40	SLA+R has been transmitted; ACK has been received	No TWDR action or	0	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	0	0	1	1	Data byte will be received and ACK will be returned
$48	SLA+R has been transmitted; NOT ACK has been received	No TWDR action or	1	0	1	X	Repeated START will be transmitted
		No TWDR action or	0	1	1	X	STOP condition will be transmitted and TWSTO Flag will be reset
		No TWDR action	1	1	1	X	STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset
$50	Data byte has been received; ACK has been returned	Read data byte or	0	0	1	0	Data byte will be received and NOT ACK will be returned
		Read data byte	0	0	1	1	Data byte will be received and ACK will be returned
$58	Data byte has been received; NOT ACK has been returned	Read data byte or	1	0	1	X	Repeated START will be transmitted
		Read data byte or	0	1	1	X	STOP condition will be transmitted and TWSTO Flag will be reset
		Read data byte	1	1	1	X	STOP condition followed by a START condition will be transmitted and TWSTO Flag will be reset

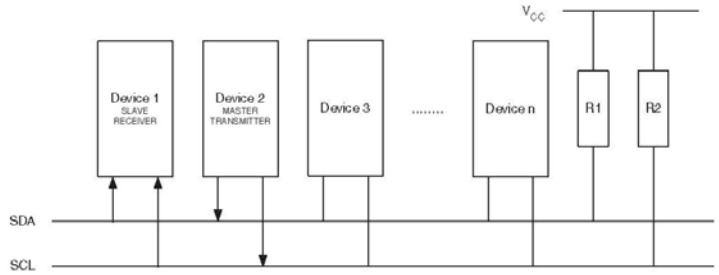
Figure 89. Formats and States in the Master Receiver Mode



Slave Receiver Mode

In the Slave Receiver mode, a number of data bytes are received from a Master Transmitter (see Figure 90). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 90. Data Transfer in Slave Receiver Mode



To initiate the Slave Receiver mode, TWAR and TWCR must be initialized as follows:

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Value	Device's Own Slave Address							

The upper seven bits are the address to which the Two-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address ($00), otherwise it will ignore the general call address.

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	0	1	0	0	0	1	0	X

TWEN must be written to one to enable the TWI. The TWEA bit must be written to one to enable the acknowledgement of the device's own Slave address or the general call address. TWSTA and TWSTO must be written to zero.

When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own Slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is "0" (write), the TWI will operate in SR mode, otherwise ST mode is entered. After its own Slave address and the write bit have been received, the TWINT Flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate soft-ware action. The appropriate action to be taken for each status code is detailed in Table 76. The Slave Receiver mode may also be entered if arbitration is lost while the TWI is in the Master mode (see states $68 and $78).

If the TWEA bit is reset during a transfer, the TWI will return a "Not Acknowledge" ("1") to SDA after the next received data byte. This can be used to indicate that the Slave is not able to receive any more bytes. While TWEA is zero, the TWI does not acknowledge its own Slave address. However, the Two-wire Serial Bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the Two-wire Serial Bus.

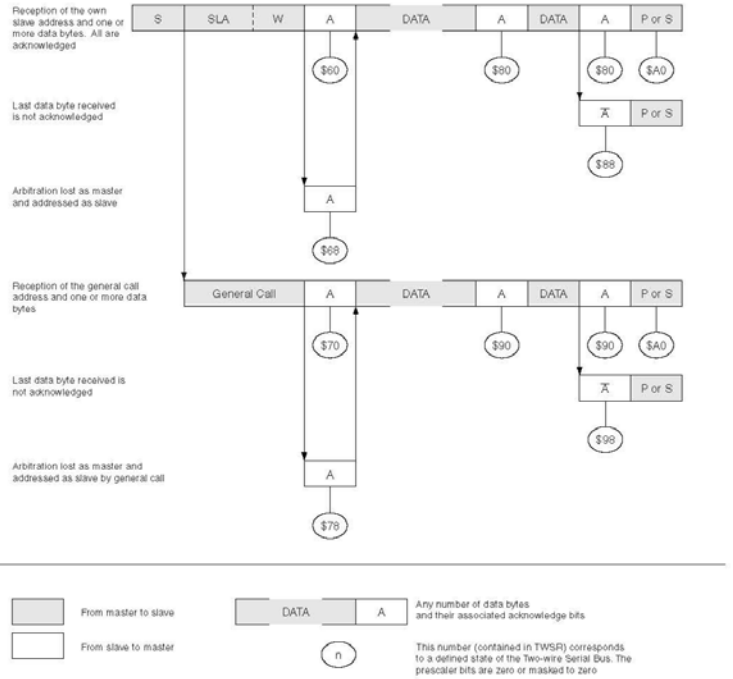
In all sleep modes other than Idle Mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own Slave address or the general call address by using the Two-wire Serial Bus clock as a clock source. The part will then wake up from sleep and the TWI will hold the SCL clock low during the wake up and until the TWINT Flag is cleared (by writing it to one). Further data reception will be carried out as normal, with the AVR clocks running as normal. Observe that if the AVR is set up with a long start-up time, the SCL line may be held low for a long time, blocking other data transmissions.

Note that the Two-wire Serial Interface Data Register – TWDR does not reflect the last byte present on the bus when waking up from these sleep modes.

Table 76. Status Codes for Slave Receiver Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the Two-wire Serial Bus and Two-wire Serial Interface Hardware	Application Software Response					
		To/from TWDR	To TWCR				Next Action Taken by TWI Hardware
			STA	STO	TWINT	TWEA	
$60	Own SLA+W has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$68	Arbitration lost in SLA+R/W as Master; own SLA+W has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$70	General call address has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$78	Arbitration lost in SLA+R/W as Master; General call address has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$80	Previously addressed with own SLA+W; data has been received; ACK has been returned	Read data byte or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		Read data byte	X	0	1	1	Data byte will be received and ACK will be returned
$88	Previously addressed with own SLA+W; data has been received; NOT ACK has been returned	Read data byte or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
		Read data byte or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"
		Read data byte or	1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		Read data byte	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
$90	Previously addressed with general call; data has been received; ACK has been returned	Read data byte or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		Read data byte	X	0	1	1	Data byte will be received and ACK will be returned
$98	Previously addressed with general call; data has been received; NOT ACK has been returned	Read data byte or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
		Read data byte or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"
		Read data byte or	1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		Read data byte	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
$A0	A STOP condition or repeated START condition has been received while still addressed as Slave	No action	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
			0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"
			1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
			1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free

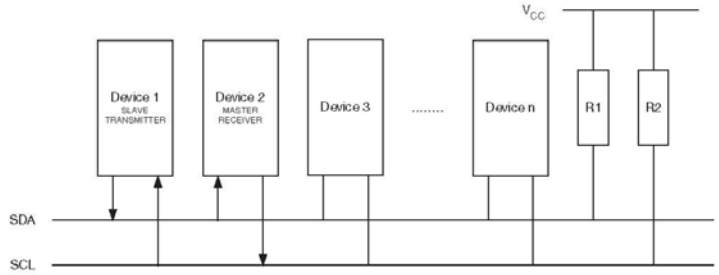
Figure 91. Formats and States in the Slave Receiver Mode



Slave Transmitter Mode

In the Slave Transmitter mode, a number of data bytes are transmitted to a Master Receiver (see Figure 92). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 92. Data Transfer in Slave Transmitter Mode



To initiate the Slave Transmitter mode, TWAR and TWCR must be initialized as follows:

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Value	Device's Own Slave Address							

The upper seven bits are the address to which the Two-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address ($00), otherwise it will ignore the general call address.

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	0	1	0	0	0	1	0	X

TWEN must be written to one to enable the TWI. The TWEA bit must be written to one to enable the acknowledgement of the device's own Slave address or the general call address. TWSTA and TWSTO must be written to zero.

When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own Slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is "1" (read), the TWI will operate in ST mode, otherwise SR mode is entered. After its own Slave address and the write bit have been received, the TWINT Flag is set and a valid status code can be read from TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in Table 77. The Slave Transmitter mode may also be entered if arbitration is lost while the TWI is in the Master mode (see state $B0).

If the TWEA bit is written to zero during a transfer, the TWI will transmit the last byte of the transfer. State $C0 or state $C8 will be entered, depending on whether the Master Receiver transmits a NACK or ACK after the final byte. The TWI is switched to the not addressed Slave mode, and will ignore the Master if it continues the transfer. Thus the Master Receiver receives all "1" as serial data. State $C8 is entered if the Master demands additional data bytes (by transmitting ACK), even though the Slave has transmitted the last byte (TWEA zero and expecting NACK from the Master).

While TWEA is zero, the TWI does not respond to its own Slave address. However, the Two-wire Serial Bus is still monitored and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the Two-wire Serial Bus.

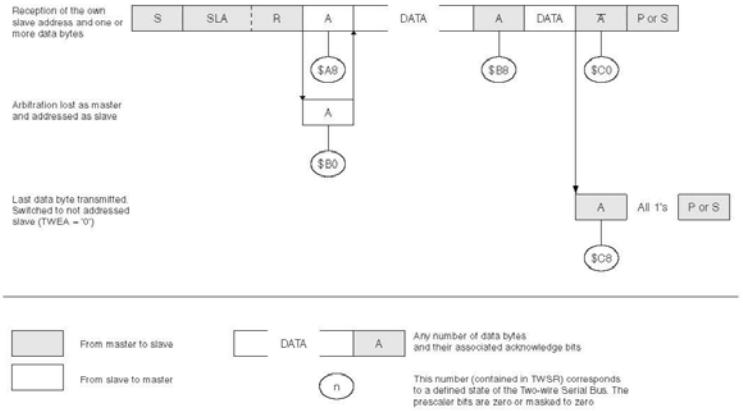
In all sleep modes other than Idle mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own Slave address or the general call address by using the Two-wire Serial Bus clock as a clock source. The part will then wake up from sleep and the TWI will hold the SCL clock will low during the wake up and until the TWINT Flag is cleared (by writing it to one). Further data transmission will be carried out as normal, with the AVR clocks running as normal. Observe that if the AVR is set up with a long start-up time, the SCL line may be held low for a long time, blocking other data transmissions.

Note that the Two-wire Serial Interface Data Register – TWDR does not reflect the last byte present on the bus when waking up from these sleep modes.

Table 77. Status Codes for Slave Transmitter Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the Two-wire Serial Bus and Two-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
$A8	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received
		Load data byte	X	0	1	1	Data byte will be transmitted and ACK should be received
$B0	Arbitration lost in SLA+R/W as Master; own SLA+R has been received; ACK has been returned	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received
		Load data byte	X	0	1	1	Data byte will be transmitted and ACK should be received
$B8	Data byte in TWDR has been transmitted; ACK has been received	Load data byte or	X	0	1	0	Last data byte will be transmitted and NOT ACK should be received
		Load data byte	X	0	1	1	Data byte will be transmitted and ACK should be received
$C0	Data byte in TWDR has been transmitted; NOT ACK has been received	No TWDR action or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
		No TWDR action or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"
		No TWDR action or	1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		No TWDR action	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free
$C8	Last data byte in TWDR has been transmitted (TWEA = "0"); ACK has been received	No TWDR action or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
		No TWDR action or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"
		No TWDR action or	1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		No TWDR action	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if TWGCE = "1"; a START condition will be transmitted when the bus becomes free

Figure 93. Formats and States in the Slave Transmitter Mode



Miscellaneous States

There are two status codes that do not correspond to a defined TWI state, see Table 78.

Status $F8 indicates that no relevant information is available because the TWINT Flag is not set. This occurs between other states, and when the TWI is not involved in a serial transfer.

Status $00 indicates that a bus error has occurred during a Two-wire Serial Bus transfer. A bus error occurs when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. When a bus error occurs, TWINT is set. To recover from a bus error, the TWSTO Flag must set and TWINT must be cleared by writing a logic one to it. This causes the TWI to enter the not addressed Slave mode and to clear the TWSTO Flag (no other bits in TWCR are affected). The SDA and SCL lines are released, and no STOP condition is transmitted.

Table 78. Miscellaneous States

Status Code (TWSR) Prescaler Bits are 0	Status of the Two-wire Serial Bus and Two-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
$F8	No relevant state information available; TWINT = "0"	No TWDR action	No TWCR action				Wait or proceed current transfer
$00	Bus error due to an illegal START or STOP condition	No TWDR action	0	1	1	X	Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and TWSTO is cleared.

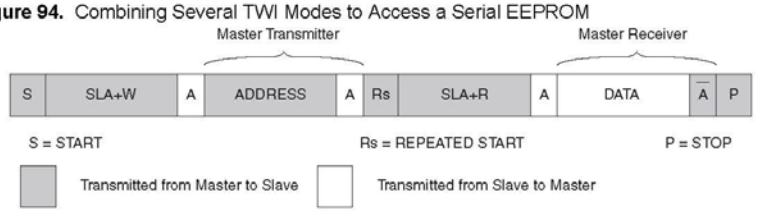
Combining Several TWI Modes

In some cases, several TWI modes must be combined in order to complete the desired action. Consider for example reading data from a serial EEPROM. Typically, such a transfer involves the following steps:

1. The transfer must be initiated
2. The EEPROM must be instructed what location should be read
3. The reading must be performed
4. The transfer must be finished

Note that data is transmitted both from Master to Slave and vice versa. The Master must instruct the Slave what location it wants to read, requiring the use of the MT mode. Subsequently, data must be read from the Slave, implying the use of the MR mode. Thus, the transfer direction must be changed. The Master must keep control of the bus during all these steps, and the steps should be carried out as an atomical operation. If this principle is violated in a multi-master system, another Master can alter the data pointer in the EEPROM between steps 2 and 3, and the Master will read the wrong data location. Such a change in transfer direction is accomplished by transmitting a REPEATED START between the transmission of the address byte and reception of the data. After a REPEATED START, the Master keeps ownership of the bus. The following figure shows the flow in this transfer.

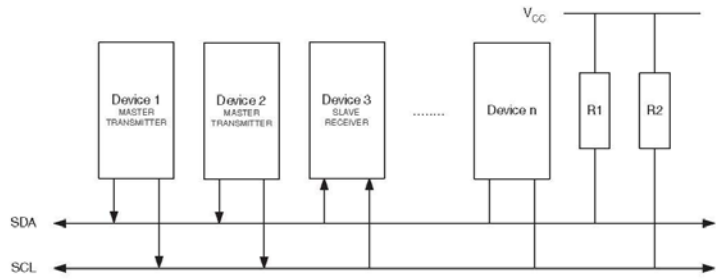
Figure 94. Combining Several TWI Modes to Access a Serial EEPROM



Multi-master Systems and Arbitration

If multiple Masters are connected to the same bus, transmissions may be initiated simultaneously by one or more of them. The TWI standard ensures that such situations are handled in such a way that one of the Masters will be allowed to proceed with the transfer, and that no data will be lost in the process. An example of an arbitration situation is depicted below, where two Masters are trying to transmit data to a Slave Receiver.

Figure 95. An Arbitration Example

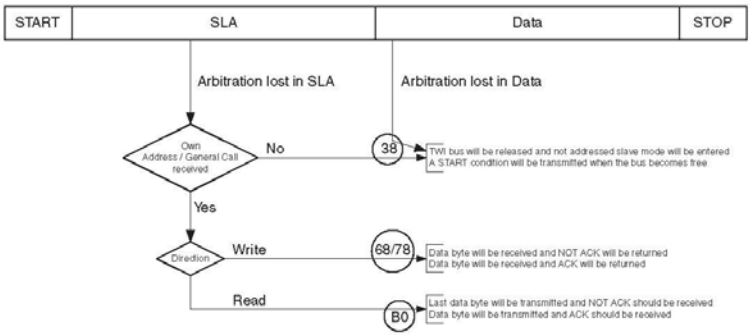


Several different scenarios may arise during arbitration, as described below:

- Two or more Masters are performing identical communication with the same Slave. In this case, neither the Slave nor any of the Masters will know about the bus contention.
- Two or more Masters are accessing the same Slave with different data or direction bit. In this case, arbitration will occur, either in the READ/WRITE bit or in the data bits. The Masters trying to output a one on SDA while another Master outputs a zero will lose the arbitration. Losing Masters will switch to not addressed Slave mode or wait until the bus is free and transmit a new START condition, depending on application software action.
- Two or more Masters are accessing different Slaves. In this case, arbitration will occur in the SLA bits. Masters trying to output a one on SDA while another Master outputs a zero will lose the arbitration. Masters losing arbitration in SLA will switch to Slave mode to check if they are being addressed by the winning Master. If addressed, they will switch to SR or ST mode, depending on the value of the READ/WRITE bit. If they are not being addressed, they will switch to not addressed Slave mode or wait until the bus is free and transmit a new START condition, depending on application software action.

This is summarized in Figure 96. Possible status values are given in circles.

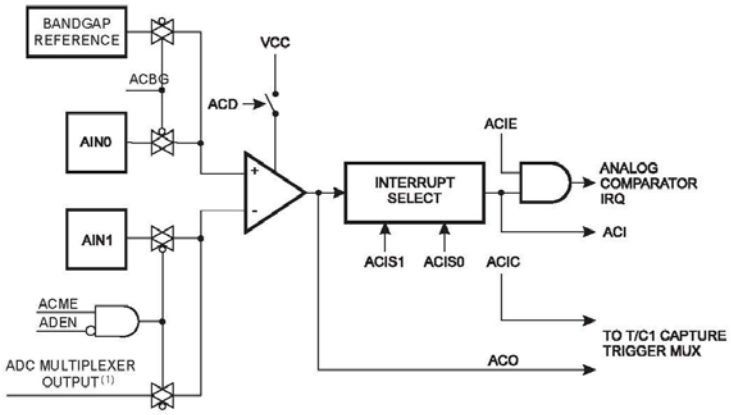
Figure 96. Possible Status Codes Caused by Arbitration



Analog Comparator

The Analog Comparator compares the input values on the positive pin AIN0 and negative pin AIN1. When the voltage on the positive pin AIN0 is higher than the voltage on the negative pin AIN1, the Analog Comparator Output, ACO, is set. The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 97.

Figure 97. Analog Comparator Block Diagram(2)



Notes: 1. See Table 80 on page 203.
2. Refer to Figure 1 on page 2 and Table 25 on page 58 for Analog Comparator pin placement.

Special Function IO Register – SFIOR

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 3 – ACME: Analog Comparator Multiplexer Enable**

When this bit is written logic one and the ADC is switched off (ADEN in ADCSRA is zero), the ADC multiplexer selects the negative input to the Analog Comparator. When this bit is written logic zero, AIN1 is applied to the negative input of the Analog Comparator. For a detailed description of this bit, see "Analog Comparator Multiplexed Input" on page 203.

Analog Comparator Control and Status Register – ACSR

Bit	7	6	5	4	3	2	1	0	
	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

- **Bit 7 – ACD: Analog Comparator Disable**

When this bit is written logic one, the power to the Analog Comparator is switched off. This bit can be set at any time to turn off the Analog Comparator. This will reduce power consumption in active and Idle mode. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – ACBG: Analog Comparator Bandgap Select**

When this bit is set, a fixed bandgap reference voltage replaces the positive input to the Analog Comparator. When this bit is cleared, AIN0 is applied to the positive input of the Analog Comparator. See "Internal Voltage Reference" on page 42.

- **Bit 5 – ACO: Analog Comparator Output**

The output of the Analog Comparator is synchronized and then directly connected to ACO. The synchronization introduces a delay of 1 - 2 clock cycles.

- **Bit 4 – ACI: Analog Comparator Interrupt Flag**

This bit is set by hardware when a comparator output event triggers the interrupt mode defined by ACIS1 and ACIS0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set and the I-bit in SREG is set. ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

- **Bit 3 – ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is written logic one and the I-bit in the Status Register is set, the Analog Comparator Interrupt is activated. When written logic zero, the interrupt is disabled.

- **Bit 2 – ACIC: Analog Comparator Input Capture Enable**

When written logic one, this bit enables the Input Capture function in Timer/Counter1 to be triggered by the Analog Comparator. The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When written logic zero, no connection between the Analog Comparator and the Input Capture function exists. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set.

- **Bits 1, 0 – ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in Table 79.

Table 79. ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge
1	1	Comparator Interrupt on Rising Output Edge

When changing the ACIS1/ACIS0 bits, the Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR Register. Otherwise an interrupt can occur when the bits are changed.

Analog Comparator Multiplexed Input

It is possible to select any of the ADC7..0 pins to replace the negative input to the Analog Comparator. The ADC multiplexer is used to select this input, and consequently, the ADC must be switched off to utilize this feature. If the Analog Comparator Multiplexer Enable bit (ACME in SFIOR) is set and the ADC is switched off (ADEN in ADCSRA is zero), MUX2..0 in ADMUX select the input pin to replace the negative input to the Analog Comparator, as shown in Table 80. If ACME is cleared or ADEN is set, AIN1 is applied to the negative input to the Analog Comparator.

Table 80. Analog Comparator Multiplexed Input

ACME	ADEN	MUX2..0	Analog Comparator Negative Input
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

Analog to Digital Converter

Features

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ±2 LSB Absolute Accuracy
- 13 µs- 260 µs Conversion Time
- Up to 15 kSPS at Maximum Resolution
- 8 Multiplexed Single Ended Input Channels
- 7 Differential Input Channels
- 2 Differential Input Channels with Optional Gain of 10x and 200x
- Optional Left adjustment for ADC Result Readout
- 0 - V_{CC} ADC Input Voltage Range
- Selectable 2.56V ADC Reference Voltage
- Free Running or Single Conversion Mode
- ADC Start Conversion by Auto Triggering on Interrupt Sources
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

The ATmega16 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows 8 single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).

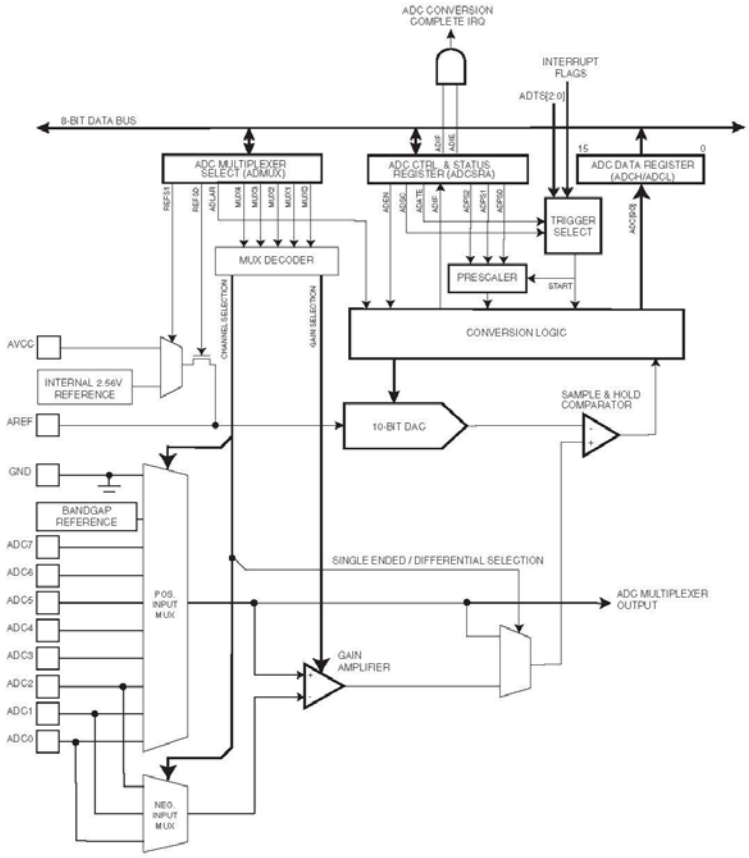
The device also supports 16 differential voltage input combinations. Two of the differential inputs (ADC1, ADC0 and ADC3, ADC2) are equipped with a programmable gain stage, providing amplification steps of 0 dB (1x), 20 dB (10x), or 46 dB (200x) on the differential input voltage before the A/D conversion. Seven differential analog input channels share a common negative terminal (ADC1), while any other ADC input can be selected as the positive input terminal. If 1x or 10x gain is used, 8-bit resolution can be expected. If 200x gain is used, 7-bit resolution can be expected.

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 98.

The ADC has a separate analog supply voltage pin, AVCC. AVCC must not differ more than ±0.3V from V_{CC}. See the paragraph "ADC Noise Canceler" on page 211 on how to connect this pin.

Internal reference voltages of nominally 2.56V or AVCC are provided On-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.

Figure 98. Analog to Digital Converter Block Schematic



Operation

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally, AVCC or an internal 2.56V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX Register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve noise immunity.

The analog input channel and differential gain are selected by writing to the MUX bits in ADMUX. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. A selection of ADC input pins can be selected as positive and negative inputs to the differential gain amplifier.

If differential channels are selected, the differential gain stage amplifies the voltage difference between the selected input channel pair by the selected gain factor. This amplified value then

becomes the analog input to the ADC. If single ended channels are used, the gain amplifier is bypassed altogether.

The ADC is enabled by setting the ADC Enable bit, ADEN in ADCSRA. Voltage reference and input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

The ADC generates a 10-bit result which is presented in the ADC Data Registers, ADCH and ADCL. By default, the result is presented right adjusted, but can optionally be presented left adjusted by setting the ADLAR bit in ADMUX.

If the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH, to ensure that the content of the Data Registers belongs to the same conversion. Once ADCL is read, ADC access to Data Registers is blocked. This means that if ADCL has been read, and a conversion completes before ADCH is read, neither register is updated and the result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL Registers is re-enabled.

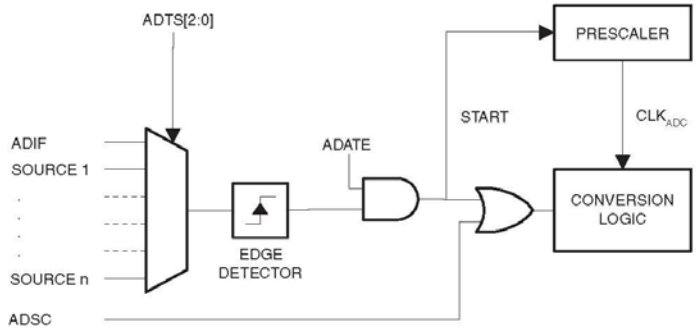
The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the Data Registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

Starting a Conversion

A single conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

Alternatively, a conversion can be triggered automatically by various sources. Auto Triggering is enabled by setting the ADC Auto Trigger Enable bit, ADATE in ADCSRA. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in SFIOR (see description of the ADTS bits for a list of the trigger sources). When a positive edge occurs on the selected trigger signal, the ADC prescaler is reset and a conversion is started. This provides a method of starting conversions at fixed intervals. If the trigger signal still is set when the conversion completes, a new conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an Interrupt Flag will be set even if the specific interrupt is disabled or the global interrupt enable bit in SREG is cleared. A conversion can thus be triggered without causing an interrupt. However, the Interrupt Flag must be cleared in order to trigger a new conversion at the next interrupt event.

Figure 99. ADC Auto Trigger Logic

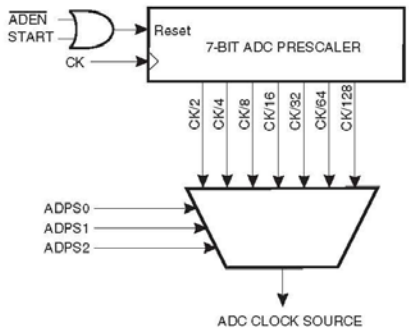


Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

Prescaling and Conversion Timing

Figure 100. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA, the conversion starts at the following rising edge of the ADC clock cycle. See "Differential Gain Channels" on page 209 for details on differential conversion timing.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA is set) takes 25 ADC clock cycles in order to initialize the analog circuitry.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of a first conversion. When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set. In single conversion mode, ADSC is cleared simultaneously. The software may then set ADSC again, and a new conversion will be initiated on the first rising ADC clock edge.

When Auto Triggering is used, the prescaler is reset when the trigger event occurs. This assures a fixed delay from the trigger event to the start of conversion. In this mode, the sample-and-hold takes place 2 ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic. When using Differential mode, along with Auto triggering from a source other than the ADC Conversion Complete, each conversion

will require 25 ADC clocks. This is because the ADC must be disabled and re-enabled after every conversion.

In Free Running mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high. For a summary of conversion times, see Table 81.

Figure 101. ADC Timing Diagram, First Conversion (Single Conversion Mode)

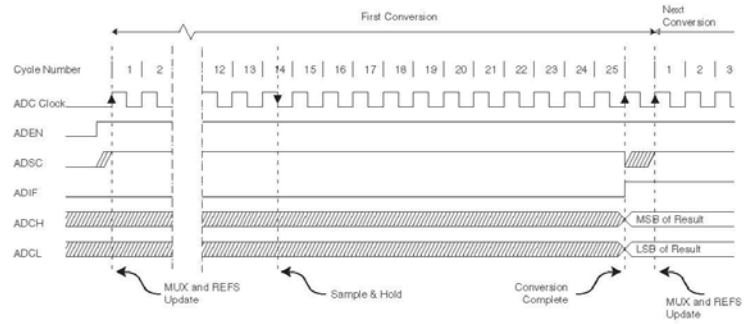


Figure 102. ADC Timing Diagram, Single Conversion

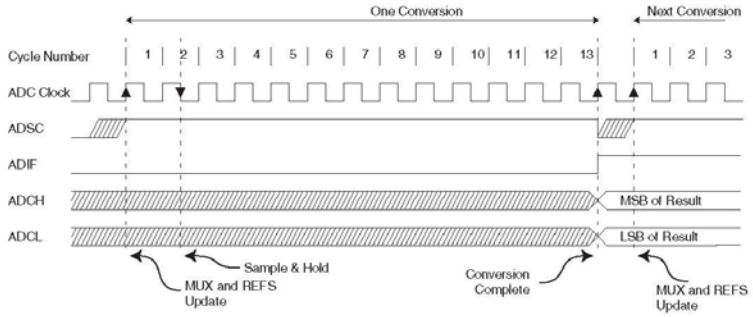


Figure 103. ADC Timing Diagram, Auto Triggered Conversion

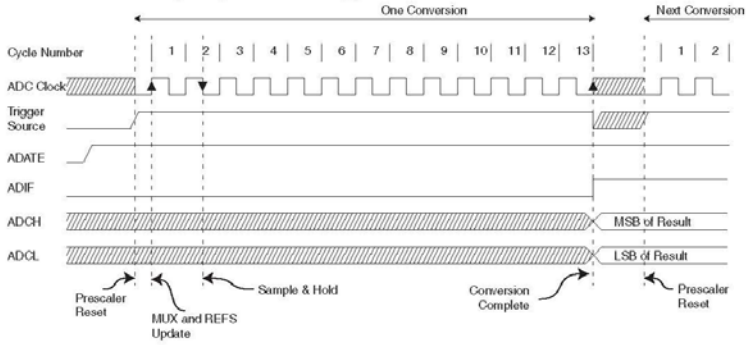


Figure 104. ADC Timing Diagram, Free Running Conversion

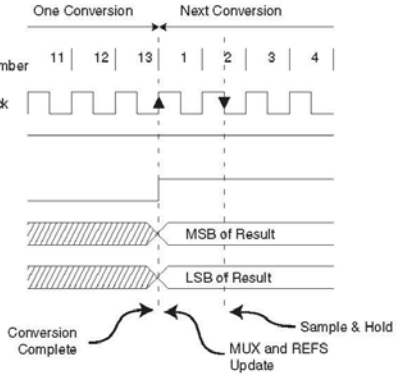


Table 81. ADC Conversion Time

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5
Normal conversions, differential	1.5/2.5	13/14

Differential Gain Channels

When using differential gain channels, certain aspects of the conversion need to be taken into consideration.

Differential conversions are synchronized to the internal clock CK_{ADC2} equal to half the ADC clock. This synchronization is done automatically by the ADC interface in such a way that the sample-and-hold occurs at a specific phase of CK_{ADC2}. A conversion initiated by the user (that is,

all single conversions, and the first free running conversion) when CK_{ADC2} is low will take the same amount of time as a single ended conversion (13 ADC clock cycles from the next prescaled clock cycle). A conversion initiated by the user when CK_{ADC2} is high will take 14 ADC clock cycles due to the synchronization mechanism. In Free Running mode, a new conversion is initiated immediately after the previous conversion completes, and since CK_{ADC2} is high at this time, all automatically started (that is, all but the first) free running conversions will take 14 ADC clock cycles.

The gain stage is optimized for a bandwidth of 4 kHz at all gain settings. Higher frequencies may be subjected to non-linear amplification. An external low-pass filter should be used if the input signal contains higher frequency components than the gain stage bandwidth. Note that the ADC clock frequency is independent of the gain stage bandwidth limitation. For example, the ADC clock period may be 6 µs, allowing a channel to be sampled at 12 kSPS, regardless of the bandwidth of this channel.

If differential gain channels are used and conversions are started by Auto Triggering, the ADC must be switched off between conversions. When Auto Triggering is used, the ADC prescaler is reset before the conversion is started. Since the gain stage is dependent of a stable ADC clock prior to the conversion, this conversion will not be valid. By disabling and then re-enabling the ADC between each conversion (writing ADEN in ADCSRA to "0" then to "1"), only extended conversions are performed. The result from the extended conversions will be valid. See "Prescaling and Conversion Timing" on page 207 for timing details.

Changing Channel or Reference Selection

The MUXn and REFS1:0 bits in the ADMUX Register are single buffered through a temporary register to which the CPU has random access. This ensures that the channels and reference selection only takes place at a safe point during the conversion. The channel and reference selection is continuously updated until a conversion is started. Once the conversion starts, the channel and reference selection is locked to ensure a sufficient sampling time for the ADC. Continuous updating resumes in the last ADC clock cycle before the conversion completes (ADIF in ADCSRA is set). Note that the conversion starts on the following rising ADC clock edge after ADSC is written. The user is thus advised not to write new channel or reference selection values to ADMUX until one ADC clock cycle after ADSC is written.

If Auto Triggering is used, the exact time of the triggering event can be indeterministic. Special care must be taken when updating the ADMUX Register, in order to control which conversion will be affected by the new settings.

If both ADATE and ADEN is written to one, an interrupt event can occur at any time. If the ADMUX Register is changed in this period, the user cannot tell if the next conversion is based on the old or the new settings. ADMUX can be safely updated in the following ways:

1. When ADATE or ADEN is cleared.
2. During conversion, minimum one ADC clock cycle after the trigger event.
3. After a conversion, before the Interrupt Flag used as trigger source is cleared.

When updating ADMUX in one of these conditions, the new settings will affect the next ADC conversion.

Special care should be taken when changing differential channels. Once a differential channel has been selected, the gain stage may take as much as 125 µs to stabilize to the new value. Thus conversions should not be started within the first 125 µs after selecting a new differential channel. Alternatively, conversion results obtained within this period should be discarded.

The same settling time should be observed for the first differential conversion after changing ADC reference (by changing the REFS1:0 bits in ADMUX).

ADC Input Channels

When changing channel selections, the user should observe the following guidelines to ensure that the correct channel is selected:

In Single Conversion mode, always select the channel before starting the conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the conversion to complete before changing the channel selection.

In Free Running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.

When switching to a differential gain channel, the first conversion result may have a poor accuracy due to the required settling time for the automatic offset cancellation circuitry. The user should preferably disregard the first conversion result.

ADC Voltage Reference

The reference voltage for the ADC (V_{REF}) indicates the conversion range for the ADC. Single ended channels that exceed V_{REF} will result in codes close to 0x3FF. V_{REF} can be selected as either AVCC, internal 2.56V reference, or external AREF pin.

AVCC is connected to the ADC through a passive switch. The internal 2.56V reference is generated from the internal bandgap reference (V_{BG}) through an internal amplifier. In either case, the external AREF pin is directly connected to the ADC, and the reference voltage can be made more immune to noise by connecting a capacitor between the AREF pin and ground. V_{REF} can also be measured at the AREF pin with a high impedant voltmeter. Note that V_{REF} is a high impedant source, and only a capacitive load should be connected in a system.

If the user has a fixed voltage source connected to the AREF pin, the user may not use the other reference voltage options in the application, as they will be shorted to the external voltage. If no external voltage is applied to the AREF pin, the user may switch between AVCC and 2.56V as reference selection. The first ADC conversion result after switching reference voltage source may be inaccurate, and the user is advised to discard this result.

If differential channels are used, the selected reference should not be closer to AVCC than indicated in Table 122 on page 297.

ADC Noise Canceler

The ADC features a noise canceler that enables conversion during sleep mode to reduce noise induced from the CPU core and other I/O peripherals. The noise canceler can be used with ADC Noise Reduction and Idle mode. To make use of this feature, the following procedure should be used:

1. Make sure that the ADC is enabled and is not busy converting. Single Conversion Mode must be selected and the ADC conversion complete interrupt must be enabled.
2. Enter ADC Noise Reduction mode (or Idle mode). The ADC will start a conversion once the CPU has been halted.
3. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the CPU and execute the ADC Conversion Complete interrupt routine. If another interrupt wakes up the CPU before the ADC conversion is complete, that interrupt will be executed, and an ADC Conversion Complete interrupt request will be generated when the ADC conversion completes. The CPU will remain in active mode until a new sleep command is executed.

Note that the ADC will not be automatically turned off when entering other sleep modes than Idle mode and ADC Noise Reduction mode. The user is advised to write zero to ADEN before entering such sleep modes to avoid excessive power consumption. If the ADC is enabled in such

sleep modes and the user wants to perform differential conversions, the user is advised to switch the ADC off and on after waking up from sleep to prompt an extended conversion to get a valid result.

Analog Input Circuitry

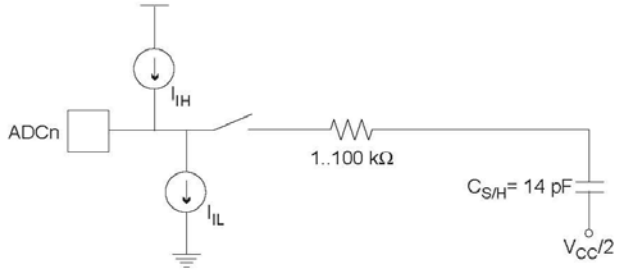
The Analog Input Circuitry for single ended channels is illustrated in Figure 105. An analog source applied to ADCn is subjected to the pin capacitance and input leakage of that pin, regardless of whether that channel is selected as input for the ADC. When the channel is selected, the source must drive the S/H capacitor through the series resistance (combined resistance in the input path).

The ADC is optimized for analog signals with an output impedance of approximately 10 kΩ or less. If such a source is used, the sampling time will be negligible. If a source with higher impedance is used, the sampling time will depend on how long time the source needs to charge the S/H capacitor, with can vary widely. The user is recommended to only use low impedant sources with slowly varying signals, since this minimizes the required charge transfer to the S/H capacitor.

If differential gain channels are used, the input circuitry looks somewhat different, although source impedances of a few hundred kΩ or less is recommended.

Signal components higher than the Nyquist frequency ($f_{ADC}/2$) should not be present for either kind of channels, to avoid distortion from unpredictable signal convolution. The user is advised to remove high frequency components with a low-pass filter before applying the signals as inputs to the ADC.

Figure 105. Analog Input Circuitry

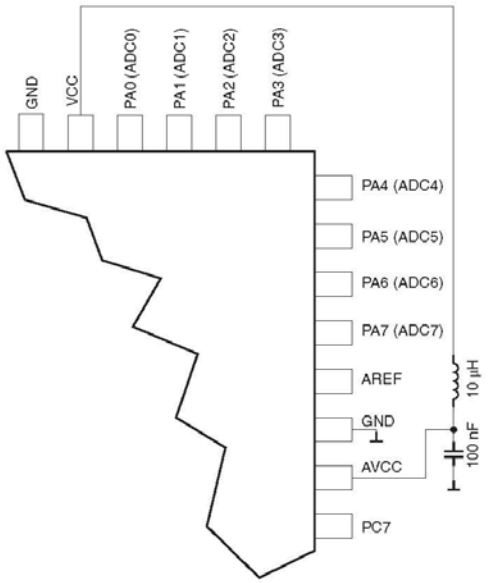


Analog Noise Canceling Techniques

Digital circuitry inside and outside the device generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. Keep analog signal paths as short as possible. Keep them well away from high-speed switching digital tracks.
2. The AVCC pin on the device should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 106.
3. Use the ADC noise canceler function to reduce induced noise from the CPU.
4. If any ADC port pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

Figure 106. ADC Power Connections



Offset Compensation Schemes

The gain stage has a built-in offset cancellation circuitry that nulls the offset of differential measurements as much as possible. The remaining offset in the analog path can be measured directly by selecting the same channel for both differential inputs. This offset residue can be then subtracted in software from the measurement results. Using this kind of software based offset correction, offset on any channel can be reduced below one LSB.

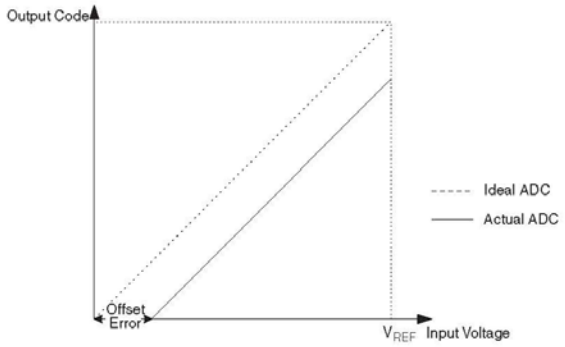
ADC Accuracy Definitions

An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as 2^n-1.

Several parameters describe the deviation from the ideal behavior:

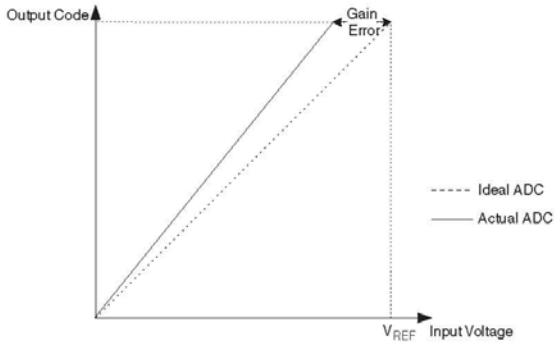
- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

Figure 107. Offset Error



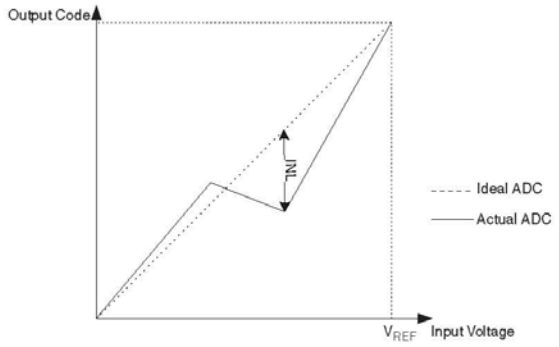
- Gain Error: After adjusting for offset, the Gain Error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB

Figure 108. Gain Error



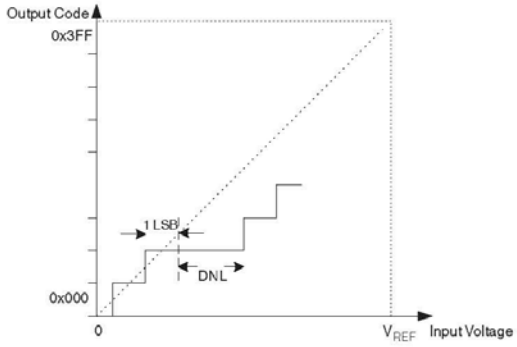
- Integral Non-linearity (INL): After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0 LSB.

Figure 109. Integral Non-linearity (INL)



- Differential Non-linearity (DNL): The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1 LSB). Ideal value: 0 LSB.

Figure 110. Differential Non-linearity (DNL)



- Quantization Error: Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ±0.5 LSB.
- Absolute Accuracy: The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of Offset, Gain Error, Differential Error, Non-linearity, and Quantization Error. Ideal value: ±0.5 LSB.

ADC Conversion Result

After the conversion is complete (ADIF is high), the conversion result can be found in the ADC Result Registers (ADCL, ADCH).

For single ended conversion, the result is

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

where V_{IN} is the voltage on the selected input pin and V_{REF} the selected voltage reference (see Table 83 on page 217 and Table 84 on page 218). 0x000 represents ground, and 0x3FF represents the selected reference voltage minus one LSB.

If differential channels are used, the result is

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot GAIN \cdot 512}{V_{REF}}$$

where V_{POS} is the voltage on the positive input pin, V_{NEG} the voltage on the negative input pin, GAIN the selected gain factor, and V_{REF} the selected voltage reference. The result is presented in two's complement form, from 0x200 (-512d) through 0x1FF (+511d). Note that if the user wants to perform a quick polarity check of the results, it is sufficient to read the MSB of the result (ADC9 in ADCH). If this bit is one, the result is negative, and if this bit is zero, the result is positive. Figure 111 shows the decoding of the differential input range.

Table 82 shows the resulting output codes if the differential input channel pair (ADCn - ADCm) is selected with a gain of GAIN and a reference voltage of V_{REF}.

Figure 111. Differential Measurement Range

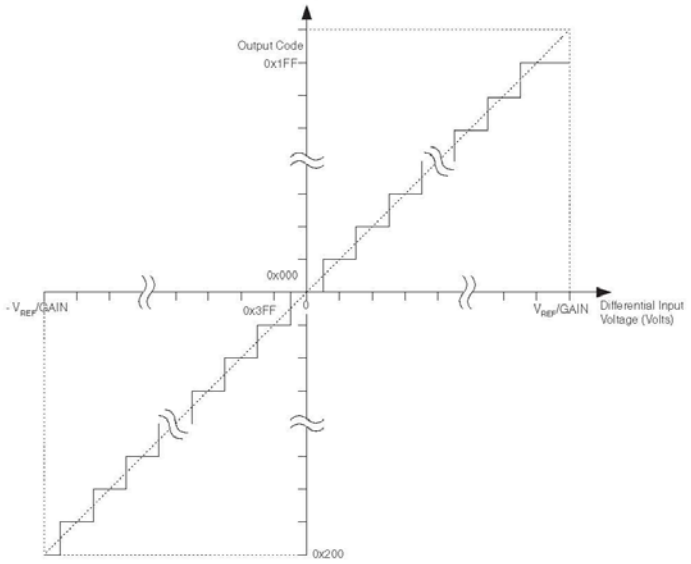


Table 82. Correlation between Input Voltage and Output Codes

V_{ADCn}	Read code	Corresponding Decimal Value
V_{ADCm} + V_{REF}/GAIN	0x1FF	511
V_{ADCm} + 511/512 V_{REF}/GAIN	0x1FF	511
V_{ADCm} + 510/512 V_{REF}/GAIN	0x1FE	510
...	...	...
V_{ADCm} + 1/512 V_{REF}/GAIN	0x001	1
V_{ADCm}	0x000	0
V_{ADCm} - 1/512 V_{REF}/GAIN	0x3FF	-1
...	...	...
V_{ADCm} - 511/512 V_{REF}/GAIN	0x201	-511
V_{ADCm} - V_{REF}/GAIN	0x200	-512

Example:

ADMUX = 0xED (ADC3 - ADC2, 10x gain, 2.56V reference, left adjusted result)

Voltage on ADC3 is 300 mV, voltage on ADC2 is 500 mV.

ADCR = 512 × 10 × (300 - 500) / 2560 = -400 = 0x270

ADCL will thus read 0x00, and ADCH will read 0x9C. Writing zero to ADLAR right adjusts the result: ADCL = 0x70, ADCH = 0x02.

ADC Multiplexer Selection Register – ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – REFS1:0: Reference Selection Bits**

These bits select the voltage reference for the ADC, as shown in Table 83. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 83. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

- **Bit 5 – ADLAR: ADC Left Adjust Result**

The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC Data Register immediately, regardless of any ongoing conver-

sions. For a complete description of this bit, see "The ADC Data Register – ADCL and ADCH" on page 220.

- **Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits**

The value of these bits selects which combination of analog inputs are connected to the ADC. These bits also select the gain for the differential channels. See Table 84 for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

Table 84. Input Channel and Gain Selections

MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010		ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x

Table 84. Input Channel and Gain Selections (Continued)

MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
11101		ADC5	ADC2	1x
11110	1.22V (V_{BG})	N/A		
11111	0 V (GND)			

ADC Control and Status Register A – ADCSRA

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion. In Free Running Mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in SFIOR.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify-Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

- **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

Table 85. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The ADC Data Register – ADCL and ADCH

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers. If differential channels are used, the result is presented in two's complement form.

When ADCL is read, the ADC Data Register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC9:0: ADC Conversion Result**

These bits represent the result from the conversion, as detailed in "ADC Conversion Result" on page 216.

Special FunctionIO Register – SFIOR

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:5 – ADTS2:0: ADC Auto Trigger Source**

If ADATE in ADCSRA is written to one, the value of these bits selects which source will trigger an ADC conversion. If ADATE is cleared, the ADTS2:0 settings will have no effect. A conversion will be triggered by the rising edge of the selected Interrupt Flag. Note that switching from a trigger source that is cleared to a trigger source that is set, will generate a positive edge on the trigger signal. If ADEN in ADCSRA is set, this will start a conversion. Switching to Free Running mode (ADTS[2:0]=0) will not cause a trigger event, even if the ADC Interrupt Flag is set.

Table 86. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

- **Bit 4 – Res: Reserved Bit**

This bit is reserved for future use. To ensure compatibility with future devices, this bit must be written to zero when SFIOR is written.

JTAG Interface and On-chip Debug System

Features

- JTAG (IEEE std. 1149.1 Compliant) Interface
- Boundary-scan Capabilities According to the IEEE std. 1149.1 (JTAG) Standard
- Debugger Access to:
  - All Internal Peripheral Units
  - Internal and External RAM
  - The Internal Register File
  - Program Counter
  - EEPROM and Flash Memories
  - Extensive On-chip Debug Support for Break Conditions, Including
  - AVR *Break* Instruction
  - Break on Change of Program Memory Flow
  - Single Step Break
  - Program Memory Breakpoints on Single Address or Address Range
  - Data Memory Breakpoints on Single Address or Address Range
- Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- On-chip Debugging Supported by AVR Studio®

Overview

The AVR IEEE std. 1149.1 compliant JTAG interface can be used for

- Testing PCBs by using the JTAG Boundary-scan capability
- Programming the non-volatile memories, Fuses and Lock bits
- On-chip Debugging

A brief description is given in the following sections. Detailed descriptions for Programming via the JTAG interface, and using the Boundary-scan Chain can be found in the sections "Programming via the JTAG Interface" on page 278 and "IEEE 1149.1 (JTAG) Boundary-scan" on page 228, respectively. The On-chip Debug support is considered being private JTAG instructions, and distributed within ATMEL and to selected third party vendors only.

Figure 112 shows a block diagram of the JTAG interface and the On-chip Debug system. The TAP Controller is a state machine controlled by the TCK and TMS signals. The TAP Controller selects either the JTAG Instruction Register or one of several Data Registers as the scan chain (Shift Register) between the TDI input and TDO output. The Instruction Register holds JTAG instructions controlling the behavior of a Data Register.

The ID-Register, Bypass Register, and the Boundary-scan Chain are the Data Registers used for board-level testing. The JTAG Programming Interface (actually consisting of several physical and virtual Data Registers) is used for JTAG Serial Programming via the JTAG interface. The Internal Scan Chain and Break Point Scan Chain are used for On-chip Debugging only.

Test Access Port – TAP

The JTAG interface is accessed through four of the AVR's pins. In JTAG terminology, these pins constitute the Test Access Port – TAP. These pins are:

- TMS: Test Mode Select. This pin is used for navigating through the TAP-controller state machine.
- TCK: Test Clock. JTAG operation is synchronous to TCK.
- TDI: Test Data In. Serial input data to be shifted in to the Instruction Register or Data Register (Scan Chains).
- TDO: Test Data Out. Serial output data from Instruction register or Data Register.

The IEEE std. 1149.1 also specifies an optional TAP signal; TRST – Test ReSeT – which is not provided.

When the JTAGEN fuse is unprogrammed, these four TAP pins are normal port pins and the TAP controller is in reset. When programmed and the JTD bit in MCUCSR is cleared, the TAP input signals are internally pulled high and the JTAG is enabled for Boundary-scan and programming. In this case, the TAP output pin (TDO) is left floating in states where the JTAG TAP controller is not shifting data, and must therefore be connected to a pull-up resistor or other hardware having pull-ups (for instance the TDI-input of the next device in the scan chain). The device is shipped with this fuse programmed.

For the On-chip Debug system, in addition to the JTAG interface pins, the $\overline{\text{RESET}}$ pin is monitored by the debugger to be able to detect external reset sources. The debugger can also pull the $\overline{\text{RESET}}$ pin low to reset the whole system, assuming only open collectors on the reset line are used in the application.

Figure 112. Block Diagram

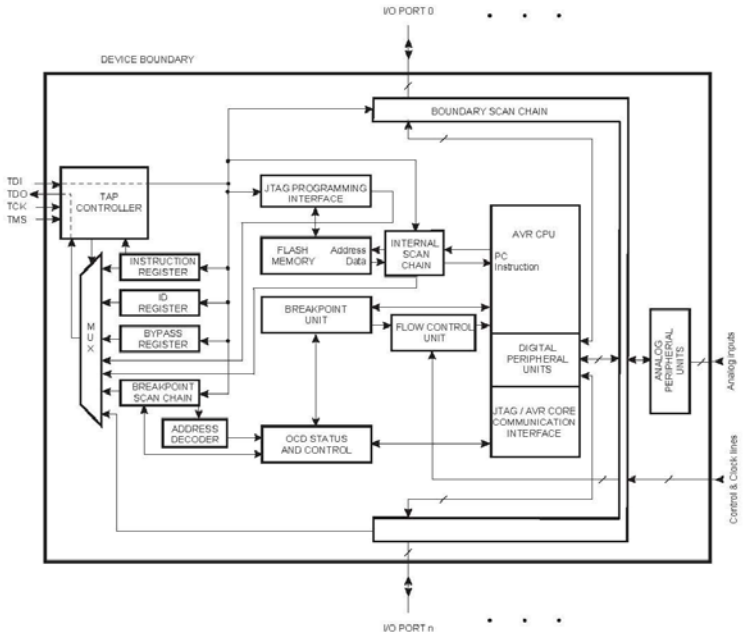
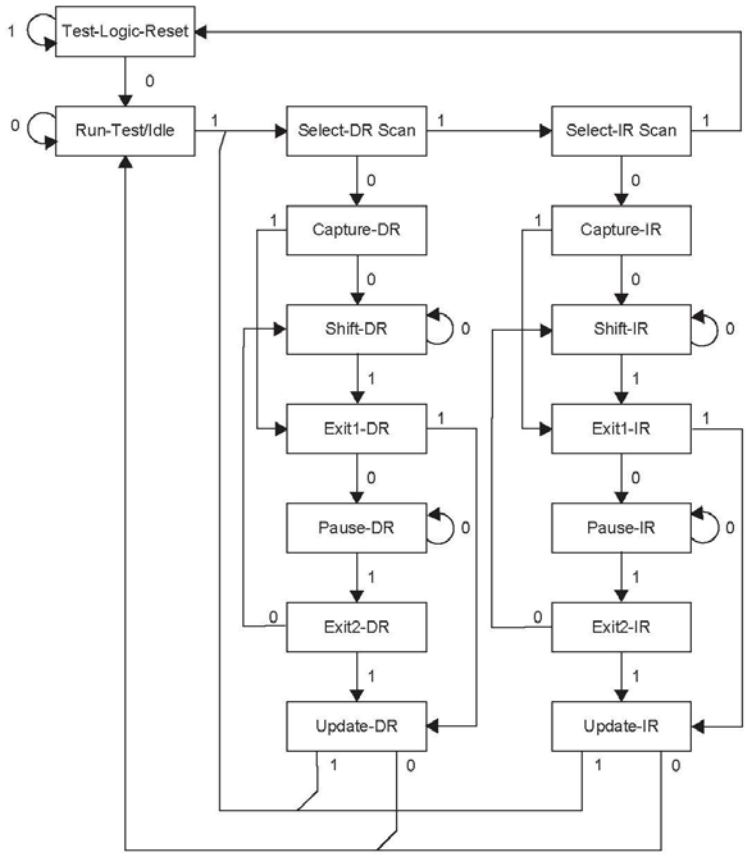


Figure 113. TAP Controller State Diagram



TAP Controller

The TAP controller is a 16-state finite state machine that controls the operation of the Boundary-scan circuitry, JTAG programming circuitry, or On-chip Debug system. The state transitions depicted in Figure 113 depend on the signal present on TMS (shown adjacent to each state transition) at the time of the rising edge at TCK. The initial state after a Power-On Reset is Test-Logic-Reset.

As a definition in this document, the LSB is shifted in and out first for all Shift Registers.

Assuming Run-Test/Idle is the present state, a typical scenario for using the JTAG interface is:

- At the TMS input, apply the sequence 1, 1, 0, 0 at the rising edges of TCK to enter the Shift Instruction Register – Shift-IR state. While in this state, shift the four bits of the JTAG instructions into the JTAG Instruction Register from the TDI input at the rising edge of TCK. The TMS input must be held low during input of the 3 LSBs in order to remain in the Shift-IR state. The MSB of the instruction is shifted in when this state is left by setting TMS high. While the instruction is shifted in from the TDI pin, the captured IR-state 0x01 is shifted out

on the TDO pin. The JTAG Instruction selects a particular Data Register as path between TDI and TDO and controls the circuitry surrounding the selected Data Register.

- Apply the TMS sequence 1, 1, 0 to re-enter the Run-Test/Idle state. The instruction is latched onto the parallel output from the Shift Register path in the Update-IR state. The Exit-IR, Pause-IR, and Exit2-IR states are only used for navigating the state machine.
- At the TMS input, apply the sequence 1, 0, 0 at the rising edges of TCK to enter the Shift Data Register – Shift-DR state. While in this state, upload the selected Data Register (selected by the present JTAG instruction in the JTAG Instruction Register) from the TDI input at the rising edge of TCK. In order to remain in the Shift-DR state, the TMS input must be held low during input of all bits except the MSB. The MSB of the data is shifted in when this state is left by setting TMS high. While the Data Register is shifted in from the TDI pin, the parallel inputs to the Data Register captured in the Capture-DR state is shifted out on the TDO pin.
- Apply the TMS sequence 1, 1, 0 to re-enter the Run-Test/Idle state. If the selected Data Register has a latched parallel-output, the latching takes place in the Update-DR state. The Exit-DR, Pause-DR, and Exit2-DR states are only used for navigating the state machine.

As shown in the state diagram, the Run-Test/Idle state need not be entered between selecting JTAG instruction and using Data Registers, and some JTAG instructions may select certain functions to be performed in the Run-Test/Idle, making it unsuitable as an Idle state.

Note: Independent of the initial state of the TAP Controller, the Test-Logic-Reset state can always be entered by holding TMS high for five TCK clock periods.

For detailed information on the JTAG specification, refer to the literature listed in "Bibliography" on page 227.

Using the Boundary-scan Chain

A complete description of the Boundary-scan capabilities are given in the section "IEEE 1149.1 (JTAG) Boundary-scan" on page 228.

Using the On-chip Debug System

As shown in Figure 112, the hardware support for On-chip Debugging consists mainly of:

- A scan chain on the interface between the internal AVR CPU and the internal peripheral units
- Break Point unit
- Communication interface between the CPU and JTAG system

All read or modify/write operations needed for implementing the Debugger are done by applying AVR instructions via the internal AVR CPU Scan Chain. The CPU sends the result to an I/O memory mapped location which is part of the communication interface between the CPU and the JTAG system.

The Break Point Unit implements Break on Change of Program Flow, Single Step Break, 2 Program Memory Break Points, and 2 combined Break Points. Together, the 4 Break Points can be configured as either:

- 4 single Program Memory Break Points
- 3 Single Program Memory Break Point + 1 single Data Memory Break Point
- 2 single Program Memory Break Points + 2 single Data Memory Break Points
- 2 single Program Memory Break Points + 1 Program Memory Break Point with mask ("range Break Point")
- 2 single Program Memory Break Points + 1 Data Memory Break Point with mask ("range Break Point")

A debugger, like the AVR Studio, may however use one or more of these resources for its internal purpose, leaving less flexibility to the end-user.

A list of the On-chip Debug specific JTAG instructions is given in "On-chip Debug Specific JTAG Instructions" on page 226.

The JTAGEN Fuse must be programmed to enable the JTAG Test Access Port. In addition, the OCDEN Fuse must be programmed and no Lock bits must be set for the On-chip Debug system to work. As a security feature, the On-chip Debug system is disabled when *any* Lock bits are set. Otherwise, the On-chip Debug system would have provided a back-door into a secured device.

The AVR JTAG ICE from Atmel is a powerful development tool for On-chip Debugging of all AVR 8-bit RISC Microcontrollers with IEEE 1149.1 compliant JTAG interface. The JTAG ICE and the AVR Studio user interface give the user complete control of the internal resources of the microcontroller, helping to reduce development time by making debugging easier. The JTAG ICE performs real-time emulation of the microcontroller while it is running in a target system.

Please refer to the Support Tools section on the AVR pages on www.atmel.com for a full description of the AVR JTEG ICE. AVR Studio can be downloaded free from Software section on the same web site.

All necessary execution commands are available in AVR Studio, both on source level and on disassembly level. The user can execute the program, single step through the code either by tracing into or stepping over functions, step out of functions, place the cursor on a statement and execute until the statement is reached, stop the execution, and reset the execution target. In addition, the user can have an unlimited number of code breakpoints (using the BREAK instruction) and up to two data memory breakpoints, alternatively combined as a mask (range) Break Point.

On-chip Debug Specific JTAG Instructions

The On-chip Debug support is considered being private JTAG instructions, and distributed within ATMEL and to selected third party vendors only. Instruction opcodes are listed for reference.

PRIVATE0; $8 Private JTAG instruction for accessing On-chip Debug system.

PRIVATE1; $9 Private JTAG instruction for accessing On-chip Debug system.

PRIVATE2; $A Private JTAG instruction for accessing On-chip Debug system.

PRIVATE3; $B Private JTAG instruction for accessing On-chip Debug system.

On-chip Debug Related Register in I/O Memory

On-chip Debug Register – OCDR

Bit	7	6	5	4	3	2	1	0	
	MSB/IDRD							LSB	OCDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The OCDR Register provides a communication channel from the running program in the microcontroller to the debugger. The CPU can transfer a byte to the debugger by writing to this location. At the same time, an Internal Flag; I/O Debug Register Dirty – IDRD – is set to indicate to the debugger that the register has been written. When the CPU reads the OCDR Register the 7 LSB will be from the OCDR Register, while the MSB is the IDRD bit. The debugger clears the IDRD bit when it has read the information.

In some AVR devices, this register is shared with a standard I/O location. In this case, the OCDR Register can only be accessed if the OCDEN Fuse is programmed, and the debugger enables access to the OCDR Register. In all other cases, the standard I/O location is accessed.

Refer to the debugger documentation for further information on how to use this register.

Using the JTAG Programming Capabilities

Programming of AVR parts via JTAG is performed via the 4-pin JTAG port, TCK, TMS, TDI and TDO. These are the only pins that need to be controlled/observed to perform JTAG programming (in addition to power pins). It is not required to apply 12V externally. The JTAGEN Fuse must be programmed and the JTD bit in the MCUSR Register must be cleared to enable the JTAG Test Access Port.

The JTAG programming capability supports:

- Flash programming and verifying
- EEPROM programming and verifying
- Fuse programming and verifying
- Lock bit programming and verifying

The Lock bit security is exactly as in Parallel Programming mode. If the Lock bits LB1 or LB2 are programmed, the OCDEN Fuse cannot be programmed unless first doing a chip erase. This is a security feature that ensures no back-door exists for reading out the content of a secured device.

The details on programming through the JTAG interface and programming specific JTAG instructions are given in the section "Programming via the JTAG Interface" on page 278.

Bibliography

For more information about general Boundary-scan, the following literature can be consulted:

- IEEE: IEEE Std. 1149.1-1990. IEEE Standard Test Access Port and Boundary-scan Architecture, IEEE, 1993
- Colin Maunder: The Board Designers Guide to Testable Logic Circuits, Addison-Wesley, 1992

IEEE 1149.1 (JTAG) Boundary-scan

Features

- JTAG (IEEE std. 1149.1 Compliant) Interface
- Boundary-scan Capabilities According to the JTAG Standard
- Full Scan of all Port Functions as well as Analog Circuitry having Off-chip Connections
- Supports the Optional IDCODE Instruction
- Additional Public AVR_RESET Instruction to Reset the AVR

System Overview

The Boundary-scan chain has the capability of driving and observing the logic levels on the digital I/O pins, as well as the boundary between digital and analog logic for analog circuitry having Off-chip connections. At system level, all ICs having JTAG capabilities are connected serially by the TDI/TDO signals to form a long Shift Register. An external controller sets up the devices to drive values at their output pins, and observe the input values received from other devices. The controller compares the received data with the expected result. In this way, Boundary-scan provides a mechanism for testing interconnections and integrity of components on Printed Circuits Boards by using the four TAP signals only.

The four IEEE 1149.1 defined mandatory JTAG instructions IDCODE, BYPASS, SAMPLE/PRELOAD, and EXTEST, as well as the AVR specific public JTAG instruction AVR_RESET can be used for testing the Printed Circuit Board. Initial scanning of the Data Register path will show the ID-code of the device, since IDCODE is the default JTAG instruction. It may be desirable to have the AVR device in Reset during Test mode. If not reset, inputs to the device may be determined by the scan operations, and the internal software may be in an undetermined state when exiting the Test mode. Entering reset, the outputs of any Port Pin will instantly enter the high impedance state, making the HIGHZ instruction redundant. If needed, the BYPASS instruction can be issued to make the shortest possible scan chain through the device. The device can be set in the reset state either by pulling the external $\overline{\text{RESET}}$ pin low, or issuing the AVR_RESET instruction with appropriate setting of the Reset Data Register.

The EXTEST instruction is used for sampling external pins and loading output pins with data. The data from the output latch will be driven out on the pins as soon as the EXTEST instruction is loaded into the JTAG IR-Register. Therefore, the SAMPLE/PRELOAD should also be used for setting initial values to the scan ring, to avoid damaging the board when issuing the EXTEST instruction for the first time. SAMPLE/PRELOAD can also be used for taking a snapshot of the external pins during normal operation of the part.

The JTAGEN Fuse must be programmed and the JTD bit in the I/O Register MCUCSR must be cleared to enable the JTAG Test Access Port.

When using the JTAG interface for Boundary-scan, using a JTAG TCK clock frequency higher than the internal chip frequency is possible. The chip clock is not required to run.

Data Registers

The Data Registers relevant for Boundary-scan operations are:

- Bypass Register
- Device Identification Register
- Reset Register
- Boundary-scan Chain

Bypass Register

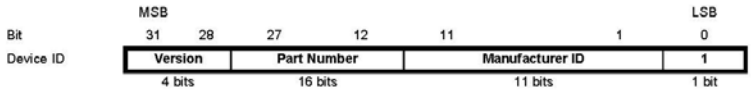
The Bypass Register consists of a single Shift Register stage. When the Bypass Register is selected as path between TDI and TDO, the register is reset to 0 when leaving the Capture-DR

controller state. The Bypass Register can be used to shorten the scan chain on a system when the other devices are to be tested.

Device Identification Register

Figure 114 shows the structure of the Device Identification Register.

Figure 114. The Format of the Device Identification Register



Version

Version is a 4-bit number identifying the revision of the component. The JTAG version number follows the revision of the device. Revision A is 0x0, revision B is 0x1 and so on. However, some revisions deviate from this rule, and the relevant version number is shown in Table 87.

Table 87. JTAG Version Numbers

Version	JTAG Version Number (Hex)
ATmega16 revision G	0x6
ATmega16 revision H	0xE
ATmega16 revision I	0x8
ATmega16 revision J	0x9
ATmega16 revision K	0xA
ATmega16 revision L	0xB

Part Number

The part number is a 16-bit code identifying the component. The JTAG Part Number for ATmega16 is listed in Table 88.

Table 88. AVR JTAG Part Number

Part Number	JTAG Part Number (Hex)
ATmega16	0x9403

Manufacturer ID

The Manufacturer ID is a 11 bit code identifying the manufacturer. The JTAG manufacturer ID for ATMEL is listed in Table 89.

Table 89. Manufacturer ID

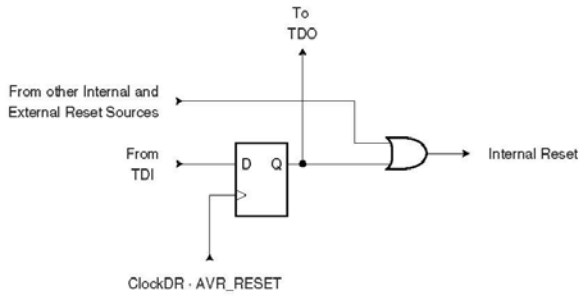
Manufacturer	JTAG Manufacturer ID (Hex)
ATMEL	0x01F

Reset Register

The Reset Register is a Test Data Register used to reset the part. Since the AVR tri-states Port Pins when reset, the Reset Register can also replace the function of the unimplemented optional JTAG instruction HIGHZ.

A high value in the Reset Register corresponds to pulling the External Reset low. The part is reset as long as there is a high value present in the Reset Register. Depending on the Fuse settings for the clock options, the part will remain reset for a Reset Time-Out Period (refer to "Clock Sources" on page 25) after releasing the Reset Register. The output from this Data Register is not latched, so the reset will take place immediately, as shown in Figure 115.

Figure 115. Reset Register



Boundary-scan Chain

The Boundary-scan Chain has the capability of driving and observing the logic levels on the digital I/O pins, as well as the boundary between digital and analog logic for analog circuitry having Off-chip connections.

See "Boundary-scan Chain" on page 232 for a complete description.

Boundary-scan Specific JTAG Instructions

The instruction register is 4-bit wide, supporting up to 16 instructions. Listed below are the JTAG instructions useful for Boundary-scan operation. Note that the optional HIGHZ instruction is not implemented, but all outputs with tri-state capability can be set in high-impedant state by using the AVR_RESET instruction, since the initial state for all port pins is tri-state.

As a definition in this datasheet, the LSB is shifted in and out first for all Shift Registers.

The OPCODE for each instruction is shown behind the instruction name in hex format. The text describes which Data Register is selected as path between TDI and TDO for each instruction.

EXTEST; $0

Mandatory JTAG instruction for selecting the Boundary-scan Chain as Data Register for testing circuitry external to the AVR package. For port-pins, Pull-up Disable, Output Control, Output Data, and Input Data are all accessible in the scan chain. For Analog circuits having Off-chip connections, the interface between the analog and the digital logic is in the scan chain. The contents of the latched outputs of the Boundary-scan chain is driven out as soon as the JTAG IR-register is loaded with the EXTEST instruction.

The active states are:

- Capture-DR: Data on the external pins are sampled into the Boundary-scan Chain.
- Shift-DR: The Internal Scan Chain is shifted by the TCK input.
- Update-DR: Data from the scan chain is applied to output pins.

IDCODE; $1

Optional JTAG instruction selecting the 32-bit ID-register as Data Register. The ID-register consists of a version number, a device number and the manufacturer code chosen by JEDEC. This is the default instruction after power-up.

The active states are:

- Capture-DR: Data in the IDCODE-register is sampled into the Boundary-scan Chain.
- Shift-DR: The IDCODE scan chain is shifted by the TCK input.

SAMPLE_PRELOAD; $2

Mandatory JTAG instruction for pre-loading the output latches and talking a snap-shot of the input/output pins without affecting the system operation. However, the output latches are not connected to the pins. The Boundary-scan Chain is selected as Data Register.

The active states are:

- Capture-DR: Data on the external pins are sampled into the Boundary-scan Chain.
- Shift-DR: The Boundary-scan Chain is shifted by the TCK input.
- Update-DR: Data from the Boundary-scan Chain is applied to the output latches. However, the output latches are not connected to the pins.

AVR_RESET; $C

The AVR specific public JTAG instruction for forcing the AVR device into the Reset mode or releasing the JTAG Reset source. The TAP controller is not reset by this instruction. The one bit Reset Register is selected as Data Register. Note that the reset will be active as long as there is a logic 'one' in the Reset Chain. The output from this chain is not latched.

The active states are:

- Shift-DR: The Reset Register is shifted by the TCK input.

BYPASS; $F

Mandatory JTAG instruction selecting the Bypass Register for Data Register.

The active states are:

- Capture-DR: Loads a logic "0" into the Bypass Register.
- Shift-DR: The Bypass Register cell between TDI and TDO is shifted.

Boundary-scan Related Register in I/O Memory

MCU Control and Status Register – MCUCSR

The MCU Control and Status Register contains control bits for general MCU functions, and provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	See Bit Description					

- **Bit 7 – JTD: JTAG Interface Disable**

When this bit is zero, the JTAG interface is enabled if the JTAGEN Fuse is programmed. If this bit is one, the JTAG interface is disabled. In order to avoid unintentional disabling or enabling of the JTAG interface, a timed sequence must be followed when changing this bit: The application software must write this bit to the desired value twice within four cycles to change its value.

If the JTAG interface is left unconnected to other JTAG circuitry, the JTD bit should be set to one. The reason for this is to avoid static current at the TDO pin in the JTAG interface.

- **Bit 4 – JTRF: JTAG Reset Flag**

This bit is set if a reset is being caused by a logic one in the JTAG Reset Register selected by the JTAG instruction AVR_RESET. This bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

Boundary-scan Chain

The Boundary-scan chain has the capability of driving and observing the logic levels on the digital I/O pins, as well as the boundary between digital and analog logic for analog circuitry having Off-chip connection.

Scanning the Digital Port Pins

Figure 116 shows the Boundary-scan Cell for a bi-directional port pin with pull-up function. The cell consists of a standard Boundary-scan cell for the Pull-up Enable – PUExn – function, and a bi-directional pin cell that combines the three signals Output Control – OCxn, Output Data – ODxn, and Input Data – IDxn, into only a two-stage Shift Register. The port and pin indexes are not used in the following description.

The Boundary-scan logic is not included in the figures in the datasheet. Figure 117 shows a simple digital Port Pin as described in the section "I/O Ports" on page 50. The Boundary-scan details from Figure 116 replaces the dashed box in Figure 117.

When no alternate port function is present, the Input Data – ID – corresponds to the PINxn Register value (but ID has no synchronizer), Output Data corresponds to the PORT Register, Output Control corresponds to the Data Direction – DD Register, and the Pull-up Enable – PUExn – corresponds to logic expression $\overline{PUD} \cdot \overline{DDxn} \cdot PORTxn$.

Digital alternate port functions are connected outside the dotted box in Figure 117 to make the scan chain read the actual pin value. For Analog function, there is a direct connection from the external pin to the analog circuit, and a scan chain is inserted on the interface between the digital logic and the analog circuitry.

Figure 116. Boundary-scan Cell for Bidirectional Port Pin with Pull-up Function.

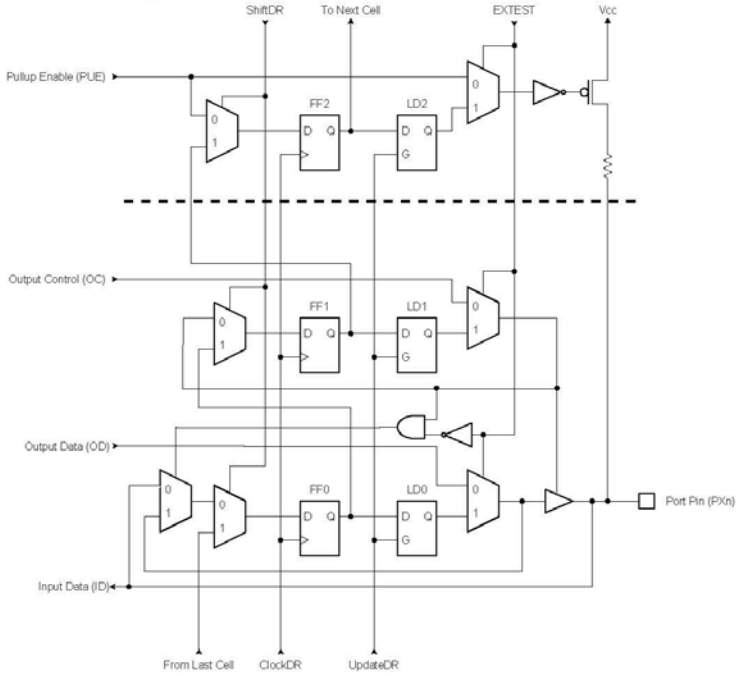
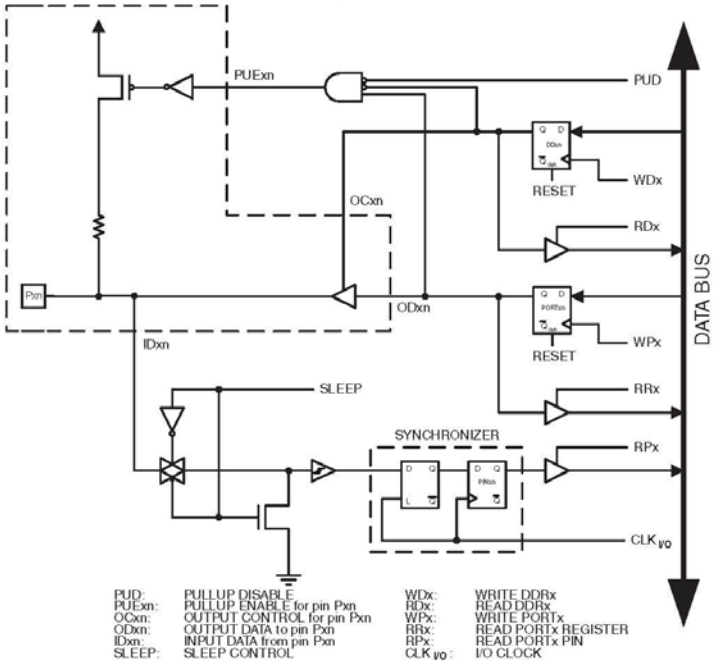


Figure 117. General Port Pin Schematic Diagram[1]



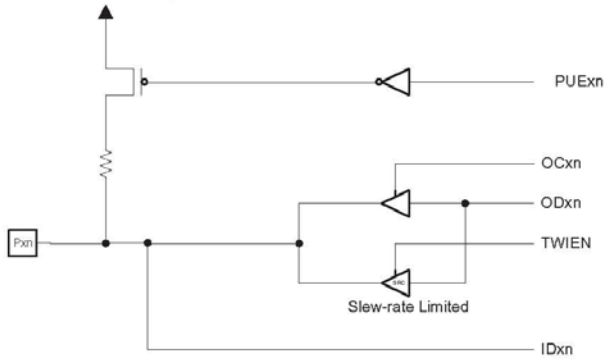
Note: 1. See Boundary-scan description for details.

Boundary-scan and the Two-wire Interface

The 2 Two-wire Interface pins SCL and SDA have one additional control signal in the scan-chain; Two-wire Interface Enable – TWIEN. As shown in Figure 118, the TWIEN signal enables a tri-state buffer with slew-rate control in parallel with the ordinary digital port pins. A general scan cell as shown in Figure 122 is attached to the TWIEN signal.

Notes: 1. A separate scan chain for the 50 ns spike filter on the input is not provided. The ordinary scan support for digital port pins suffice for connectivity tests. The only reason for having TWIEN in the scan path, is to be able to disconnect the slew-rate control buffer when doing boundary-scan.
2. Make sure the OC and TWIEN signals are not asserted simultaneously, as this will lead to drive contention.

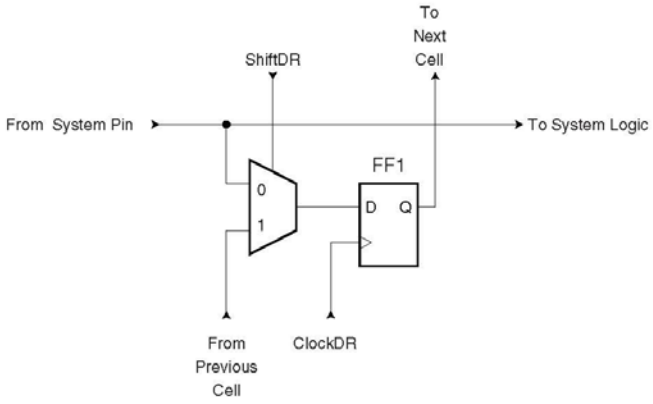
Figure 118. Additional Scan Signal for the Two-wire Interface



Scanning the RESET Pin

The RESET pin accepts 5V active low logic for standard reset operation, and 12V active high logic for High Voltage Parallel Programming. An observe-only cell as shown in Figure 119 is inserted both for the 5V reset signal; RSTT, and the 12V reset signal; RSTHV.

Figure 119. Observe-only Cell



Scanning the Clock Pins

The AVR devices have many clock options selectable by fuses. These are: Internal RC Oscillator, External RC, External Clock, (High Frequency) Crystal Oscillator, Low Frequency Crystal Oscillator, and Ceramic Resonator.

Figure 120 shows how each Oscillator with external connection is supported in the scan chain. The Enable signal is supported with a general boundary-scan cell, while the Oscillator/Clock output is attached to an observe-only cell. In addition to the main clock, the Timer Oscillator is scanned in the same way. The output from the internal RC Oscillator is not scanned, as this Oscillator does not have external connections.

Figure 120. Boundary-scan Cells for Oscillators and Clock Options

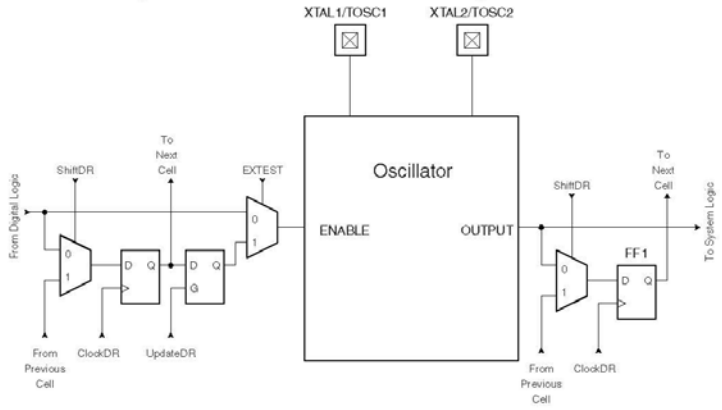


Table 90 summaries the scan registers for the external clock pin XTAL1, Oscillators with XTAL1/XTAL2 connections as well as 32 kHz Timer Oscillator.

Table 90. Scan Signals for the Oscillators[1][2][3]

Enable Signal	Scanned Clock Line	Clock Option	Scanned Clock Line when not Used
EXTCLKEN	EXTCLK (XTAL1)	External Clock	0
OSCON	OSCCK	External Crystal External Ceramic Resonator	0
RCOSCEN	RCCK	External RC	1
OSC32EN	OSC32CK	Low Freq. External Crystal	0
TOSKON	TOSCK	32 kHz Timer Oscillator	0

Notes:
1. Do not enable more than one clock source as main clock at a time.
2. Scanning an Oscillator output gives unpredictable results as there is a frequency drift between the Internal Oscillator and the JTAG TCK clock. If possible, scanning an external clock is preferred.
3. The clock configuration is programmed by fuses. As a fuse is not changed run-time, the clock configuration is considered fixed for a given application. The user is advised to scan the same clock option as to be used in the final system. The enable signals are supported in the scan chain because the system logic can disable clock options in sleep modes, thereby disconnecting the Oscillator pins from the scan path if not provided. The INTCAP Fuses are not supported in the scan-chain, so the boundary scan chain can not make a XTAL Oscillator requiring internal capacitors to run unless the fuse is correctly programmed.

Scanning the Analog Comparator

The relevant Comparator signals regarding Boundary-scan are shown in Figure 121. The Boundary-scan cell from Figure 122 is attached to each of these signals. The signals are described in Table 91.

The Comparator need not be used for pure connectivity testing, since all analog inputs are shared with a digital port pin as well.

Figure 121. Analog Comparator

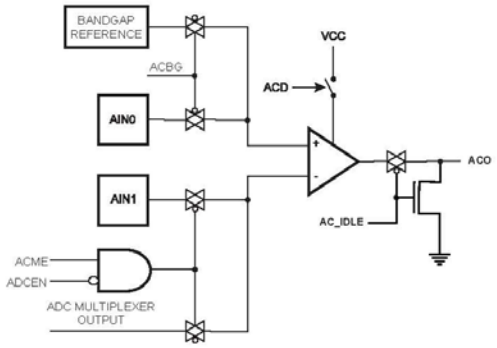


Figure 122. General Boundary-scan Cell used for Signals for Comparator and ADC

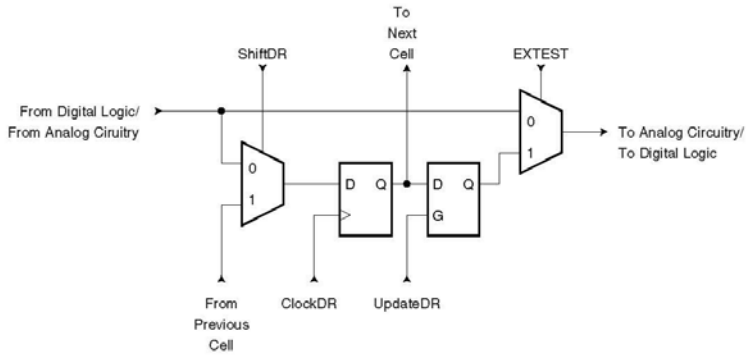


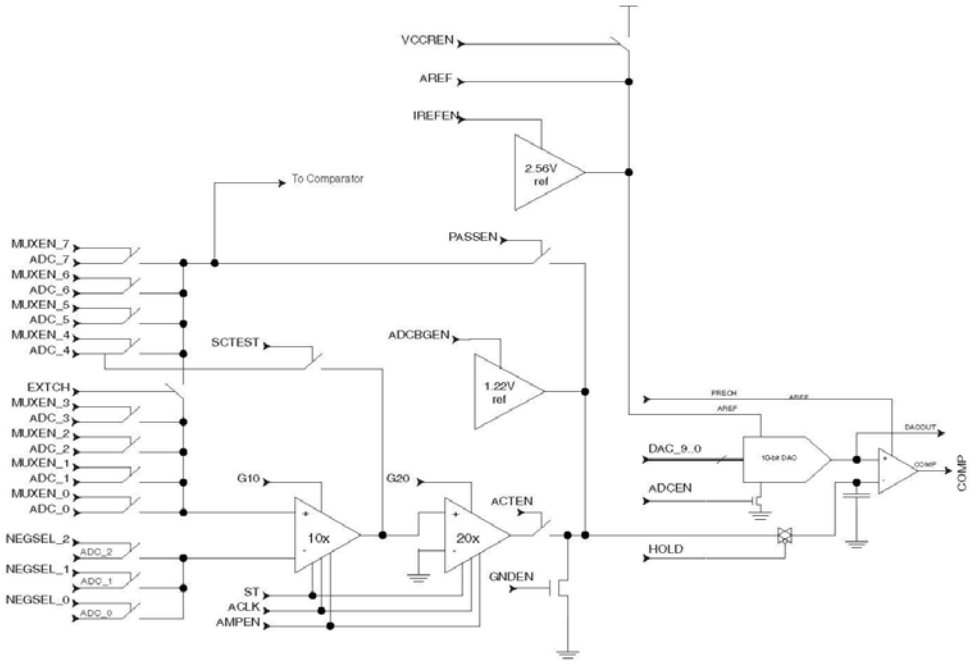
Table 91. Boundary-scan Signals for the Analog Comparator

Signal Name	Direction as Seen from the Comparator	Description	Recommended Input when Not in Use	Output Values when Recommended Inputs are Used
AC_IDLE	Input	Turns off Analog comparator when true	1	Depends upon µC code being executed
ACO	Output	Analog Comparator Output	Will become input to µC code being executed	0
ACME	Input	Uses output signal from ADC mux when true	0	Depends upon µC code being executed
ACBG	Input	Bandgap Reference enable	0	Depends upon µC code being executed

Scanning the ADC

Figure 123 shows a block diagram of the ADC with all relevant control and observe signals. The Boundary-scan cell from Figure 122 is attached to each of these signals. The ADC need not be used for pure connectivity testing, since all analog inputs are shared with a digital port pin as well.

Figure 123. Analog to Digital Converter



The signals are described briefly in Table 92.

Table 92. Boundary-scan Signals for the ADC

Signal Name	Direction as Seen from the ADC	Description	Recommended Input when Not in Use	Output Values when Recommended Inputs are used, and CPU is not Using the ADC
COMP	Output	Comparator Output	0	0
ACLK	Input	Clock signal to gain stages implemented as Switch-cap filters	0	0
ACTEN	Input	Enable path from gain stages to the comparator	0	0
ADCBGEN	Input	Enable Band-gap reference as negative input to comparator	0	0
ADCEN	Input	Power-on signal to the ADC	0	0
AMPEN	Input	Power-on signal to the gain stages	0	0
DAC_9	Input	Bit 9 of digital value to DAC	1	1
DAC_8	Input	Bit 8 of digital value to DAC	0	0
DAC_7	Input	Bit 7 of digital value to DAC	0	0
DAC_6	Input	Bit 6 of digital value to DAC	0	0
DAC_5	Input	Bit 5 of digital value to DAC	0	0
DAC_4	Input	Bit 4 of digital value to DAC	0	0
DAC_3	Input	Bit 3 of digital value to DAC	0	0
DAC_2	Input	Bit 2 of digital value to DAC	0	0
DAC_1	Input	Bit 1 of digital value to DAC	0	0
DAC_0	Input	Bit 0 of digital value to DAC	0	0
EXTCH	Input	Connect ADC channels 0 - 3 to by-pass path around gain stages	1	1
G10	Input	Enable 10x gain	0	0
G20	Input	Enable 20x gain	0	0
GNDEN	Input	Ground the negative input to comparator when true	0	0
HOLD	Input	Sample&Hold signal. Sample analog signal when low. Hold signal when high. If gain stages are used, this signal must go active when ACLK is high.	1	1
IREFEN	Input	Enables Band-gap reference as AREF signal to DAC	0	0
MUXEN_7	Input	Input Mux bit 7	0	0
MUXEN_6	Input	Input Mux bit 6	0	0
MUXEN_5	Input	Input Mux bit 5	0	0
MUXEN_4	Input	Input Mux bit 4	0	0
MUXEN_3	Input	Input Mux bit 3	0	0

Table 92. Boundary-scan Signals for the ADC (Continued)

Signal Name	Direction as Seen from the ADC	Description	Recommended Input when Not in Use	Output Values when Recommended Inputs are used, and CPU is not Using the ADC
MUXEN_2	Input	Input Mux bit 2	0	0
MUXEN_1	Input	Input Mux bit 1	0	0
MUXEN_0	Input	Input Mux bit 0	1	1
NEGSEL_2	Input	Input Mux for negative input for differential signal, bit 2	0	0
NEGSEL_1	Input	Input Mux for negative input for differential signal, bit 1	0	0
NEGSEL_0	Input	Input Mux for negative input for differential signal, bit 0	0	0
PASSEN	Input	Enable pass-gate of gain stages.	1	1
PRECH	Input	Precharge output latch of comparator. (Active low)	1	1
SCTEST	Input	Switch-cap TEST enable. Output from x10 gain stage send out to Port Pin having ADC_4	0	0
ST	Input	Output of gain stages will settle faster if this signal is high first two ACLK periods after AMPEN goes high.	0	0
VCCREN	Input	Selects Vcc as the ACC reference voltage.	0	0

Note: Incorrect setting of the switches in Figure 123 will make signal contention and may damage the part. There are several input choices to the S&H circuitry on the negative input of the output comparator in Figure 123. Make sure only one path is selected from either one ADC pin, Bandgap reference source, or Ground.

If the ADC is not to be used during scan, the recommended input values from Table 92 should be used. The user is recommended **not** to use the Differential Gain stages during scan. Switch-cap based gain stages require fast operation and accurate timing which is difficult to obtain when used in a scan chain. Details concerning operations of the differential gain stage is therefore not provided.

The AVR ADC is based on the analog circuitry shown in Figure 123 with a successive approximation algorithm implemented in the digital logic. When used in Boundary-scan, the problem is usually to ensure that an applied analog voltage is measured within some limits. This can easily be done without running a successive approximation algorithm: apply the lower limit on the digital DAC[9:0] lines, make sure the output from the comparator is low, then apply the upper limit on the digital DAC[9:0] lines, and verify the output from the comparator to be high.

The ADC need not be used for pure connectivity testing, since all analog inputs are shared with a digital port pin as well.

When using the ADC, remember the following:

- The Port Pin for the ADC channel in use must be configured to be an input with pull-up disabled to avoid signal contention.
- In Normal mode, a dummy conversion (consisting of 10 comparisons) is performed when enabling the ADC. The user is advised to wait at least 200 ns after enabling the ADC before

controlling/observing any ADC signal, or perform a dummy conversion before using the first result.

- The DAC values must be stable at the midpoint value 0x200 when having the HOLD signal low (Sample mode).

As an example, consider the task of verifying a 1.5V ±5% input signal at ADC channel 3 when the power supply is 5.0V and AREF is externally connected to V_{CC}.

$$\text{The lower limit is:} \quad \lceil 1024 \cdot 1{,}5V \cdot 0{,}95/5V \rceil = 291 = 0x123$$
$$\text{The upper limit is:} \quad \lceil 1024 \cdot 1{,}5V \cdot 1{,}05/5V \rceil = 323 = 0x143$$

The recommended values from Table 92 are used unless other values are given in the algorithm in Table 93. Only the DAC and Port Pin values of the Scan-chain are shown. The column "Actions" describes what JTAG instruction to be used before filling the Boundary-scan Register with the succeeding columns. The verification should be done on the data scanned out when scanning in the data on the same row in the table.

Table 93. Algorithm for Using the ADC

Step	Actions	ADCEN	DAC	MUXEN	HOLD	PRECH	PA3. Data	PA3. Control	PA3. Pullup_ Enable
1	SAMPLE _PRELO AD	1	0x200	0x08	1	1	0	0	0
2	EXTEST	1	0x200	0x08	0	1	0	0	0
3		1	0x200	0x08	1	1	0	0	0
4		1	0x123	0x08	1	1	0	0	0
5		1	0x123	0x08	1	0	0	0	0
6	Verify the COMP bit scanned out to be 0	1	0x200	0x08	1	1	0	0	0
7		1	0x200	0x08	0	1	0	0	0
8		1	0x200	0x08	1	1	0	0	0
9		1	0x143	0x08	1	1	0	0	0
10		1	0x143	0x08	1	0	0	0	0
11	Verify the COMP bit scanned out to be 1	1	0x200	0x08	1	1	0	0	0

Using this algorithm, the timing constraint on the HOLD signal constrains the TCK clock frequency. As the algorithm keeps HOLD high for five steps, the TCK clock frequency has to be at least five times the number of scan bits divided by the maximum hold time, $t_{hold,max}$.

ATmega16 Boundary-scan Order

Table 94 shows the scan order between TDI and TDO when the Boundary-scan chain is selected as data path. Bit 0 is the LSB; the first bit scanned in, and the first bit scanned out. The scan order follows the pin-out order as far as possible. Therefore, the bits of Port A is scanned in the opposite bit order of the other ports. Exceptions from the rules are the Scan chains for the analog circuits, which constitute the most significant bits of the scan chain regardless of which physical pin they are connected to. In Figure 116, PXn. Data corresponds to FF0, PXn. Control corresponds to FF1, and PXn. Pullup_enable corresponds to FF2. Bit 2, 3, 4, and 5 of Port C is not in the scan chain, since these pins constitute the TAP pins when the JTAG is enabled.

Table 94. ATmega16 Boundary-scan Order

Bit Number	Signal Name	Module
140	AC_IDLE	Comparator
139	ACO	
138	ACME	
137	ACBG	
136	COMP	ADC
135	PRIVATE_SIGNAL1[1]	
134	ACLK	
133	ACTEN	
132	PRIVATE_SIGNAL2[2]	
131	ADCBGEN	
130	ADCEN	
129	AMPEN	
128	DAC_9	
127	DAC_8	
126	DAC_7	
125	DAC_6	
124	DAC_5	
123	DAC_4	
122	DAC_3	
121	DAC_2	
120	DAC_1	
119	DAC_0	
118	EXTCH	
117	G10	
116	G20	
115	GNDEN	
114	HOLD	
113	IREFEN	
112	MUXEN_7	

Table 94. ATmega16 Boundary-scan Order (Continued)

Bit Number	Signal Name	Module
111	MUXEN_6	
110	MUXEN_5	
109	MUXEN_4	
108	MUXEN_3	
107	MUXEN_2	
106	MUXEN_1	
105	MUXEN_0	
104	NEGSEL_2	
103	NEGSEL_1	
102	NEGSEL_0	
101	PASSEN	
100	PRECH	
99	SCTEST	
98	ST	
97	VCCREN	
96	PB0.Data	Port B
95	PB0.Control	
94	PB0.Pullup_Enable	
93	PB1.Data	
92	PB1.Control	
91	PB1.Pullup_Enable	
90	PB2.Data	
89	PB2.Control	
88	PB2.Pullup_Enable	
87	PB3.Data	
86	PB3.Control	
85	PB3.Pullup_Enable	
84	PB4.Data	
83	PB4.Control	
82	PB4.Pullup_Enable	
81	PB5.Data	
80	PB5.Control	
79	PB5.Pullup_Enable	
78	PB6.Data	
77	PB6.Control	
76	PB6.Pullup_Enable	

Table 94. ATmega16 Boundary-scan Order (Continued)

Bit Number	Signal Name	Module
75	PB7.Data	
74	PB7.Control	
73	PB7.Pullup_Enable	
72	RSTT	Reset Logic (Observe-Only)
71	RSTHV	
70	EXTCLKEN	Enable signals for main clock/Oscillators
69	OSCON	
68	RCOSCEN	
67	OSC32EN	
66	EXTCLK (XTAL1)	Clock input and Oscillators for the main clock (Observe-Only)
65	OSCCK	
64	RCCK	
63	OSC32CK	
62	TWIEN	TWI
61	PD0.Data	Port D
60	PD0.Control	
59	PD0.Pullup_Enable	
58	PD1.Data	
57	PD1.Control	
56	PD1.Pullup_Enable	
55	PD2.Data	
54	PD2.Control	
53	PD2.Pullup_Enable	
52	PD3.Data	
51	PD3.Control	
50	PD3.Pullup_Enable	
49	PD4.Data	
48	PD4.Control	
47	PD4.Pullup_Enable	
46	PD5.Data	
45	PD5.Control	
44	PD5.Pullup_Enable	
43	PD6.Data	
42	PD6.Control	
41	PD6.Pullup_Enable	
40	PD7.Data	

Table 94. ATmega16 Boundary-scan Order (Continued)

Bit Number	Signal Name	Module
39	PD7.Control	
38	PD7.Pullup_Enable	
37	PC0.Data	Port C
36	PC0.Control	
35	PC0.Pullup_Enable	
34	PC1.Data	
33	PC1.Control	
32	PC1.Pullup_Enable	
31	PC6.Data	
30	PC6.Control	
29	PC6.Pullup_Enable	
28	PC7.Data	
27	PC7.Control	
26	PC7.Pullup_Enable	
25	TOSC	32 kHz Timer Oscillator
24	TOSCON	
23	PA7.Data	Port A
22	PA7.Control	
21	PA7.Pullup_Enable	
20	PA6.Data	
19	PA6.Control	
18	PA6.Pullup_Enable	
17	PA5.Data	
16	PA5.Control	
15	PA5.Pullup_Enable	
14	PA4.Data	
13	PA4.Control	
12	PA4.Pullup_Enable	
11	PA3.Data	
10	PA3.Control	
9	PA3.Pullup_Enable	
8	PA2.Data	
7	PA2.Control	
6	PA2.Pullup_Enable	
5	PA1.Data	

Table 94. ATmega16 Boundary-scan Order (Continued)

Bit Number	Signal Name	Module
4	PA1.Control	
3	PA1.Pullup_Enable	
2	PA0.Data	
1	PA0.Control	
0	PA0.Pullup_Enable	

Notes: 1. PRIVATE_SIGNAL1 should always be scanned in as zero.
2. PRIVATE:SIGNAL2 should always be scanned in as zero.

Boundary-scan Description Language Files

Boundary-scan Description Language (BSDL) files describe Boundary-scan capable devices in a standard format used by automated test-generation software. The order and function of bits in the Boundary-scan Data Register are included in this description. A BSDL file for ATmega16 is available.

Boot Loader Support – Read-While-Write Self-Programming

The Boot Loader Support provides a real Read-While-Write Self-Programming mechanism for downloading and uploading program code by the MCU itself. This feature allows flexible application software updates controlled by the MCU using a Flash-resident Boot Loader program. The Boot Loader program can use any available data interface and associated protocol to read code and write (program) that code into the Flash memory, or read the code from the Program memory. The program code within the Boot Loader section has the capability to write into the entire Flash, including the Boot Loader memory. The Boot Loader can thus even modify itself, and it can also erase itself from the code if the feature is not needed anymore. The size of the Boot Loader memory is configurable with Fuses and the Boot Loader has two separate sets of Boot Lock bits which can be set independently. This gives the user a unique flexibility to select different levels of protection.

Features

- **Read-While-Write Self-Programming**
- **Flexible Boot Memory size**
- **High Security (Separate Boot Lock Bits for a Flexible Protection)**
- **Separate Fuse to Select Reset Vector**
- **Optimized Page[1] Size**
- **Code Efficient Algorithm**
- **Efficient Read-Modify-Write Support**

Note: 1. A page is a section in the flash consisting of several bytes (see Table 107 on page 262) used during programming. The page organization does not affect normal operation.

Application and Boot Loader Flash Sections

The Flash memory is organized in two main sections, the Application section and the Boot Loader section (see Figure 125). The size of the different sections is configured by the BOOTSZ Fuses as shown in Table 100 on page 257 and Figure 125. These two sections can have different level of protection since they have different sets of Lock bits.

Application Section

The Application section is the section of the Flash that is used for storing the application code. The protection level for the application section can be selected by the Application Boot Lock bits (Boot Lock bits 0), see Table 96 on page 249. The Application section can never store any Boot Loader code since the SPM instruction is disabled when executed from the Application section.

BLS – Boot Loader Section

While the Application section is used for storing the application code, the The Boot Loader software must be located in the BLS since the SPM instruction can initiate a programming when executing from the BLS only. The SPM instruction can access the entire Flash, including the BLS itself. The protection level for the Boot Loader section can be selected by the Boot Loader Lock bits (Boot Lock bits 1), see Table 97 on page 249.

Read-While-Write and no Read-While-Write Flash Sections

Whether the CPU supports Read-While-Write or if the CPU is halted during a Boot Loader software update is dependent on which address that is being programmed. In addition to the two sections that are configurable by the BOOTSZ Fuses as described above, the Flash is also divided into two fixed sections, the Read-While-Write (RWW) section and the No Read-While-Write (NRWW) section. The limit between the RWW- and NRWW sections is given in Table 101 on page 257 and Figure 125 on page 248. The main difference between the two sections is:

- When erasing or writing a page located inside the RWW section, the NRWW section can be read during the operation.
- When erasing or writing a page located inside the NRWW section, the CPU is halted during the entire operation.

Note that the user software can never read any code that is located inside the RWW section during a Boot Loader software operation. The syntax “Read-While-Write section” refers to which section that is being programmed (erased or written), not which section that actually is being read during a Boot Loader software update.

RWW – Read-While-Write Section

If a Boot Loader software update is programming a page inside the RWW section, it is possible to read code from the Flash, but only code that is located in the NRWW section. During an on-going programming, the software must ensure that the RWW section never is being read. If the user software is trying to read code that is located inside the RWW section (that is, by a call/jmp/lpm or an interrupt) during programming, the software might end up in an unknown state. To avoid this, the interrupts should either be disabled or moved to the Boot Loader section. The Boot Loader section is always located in the NRWW section. The RWW Section Busy bit (RWWSB) in the Store Program Memory Control Register (SPMCR) will be read as logical one as long as the RWW section is blocked for reading. After a programming is completed, the RWWSB must be cleared by software before reading code located in the RWW section. See "Store Program Memory Control Register – SPMCR" on page 250. for details on how to clear RWWSB.

NRWW – No Read-While-Write Section

The code located in the NRWW section can be read when the Boot Loader software is updating a page in the RWW section. When the Boot Loader code updates the NRWW section, the CPU is halted during the entire page erase or page write operation.

Table 95. Read-While-Write Features

Which Section does the Z-pointer Address during the Programming?	Which Section can be Read during Programming?	Is the CPU Halted?	Read-While-Write Supported?
RWW section	NRWW section	No	Yes
NRWW section	None	Yes	No

Figure 124. Read-While-Write vs. No Read-While-Write

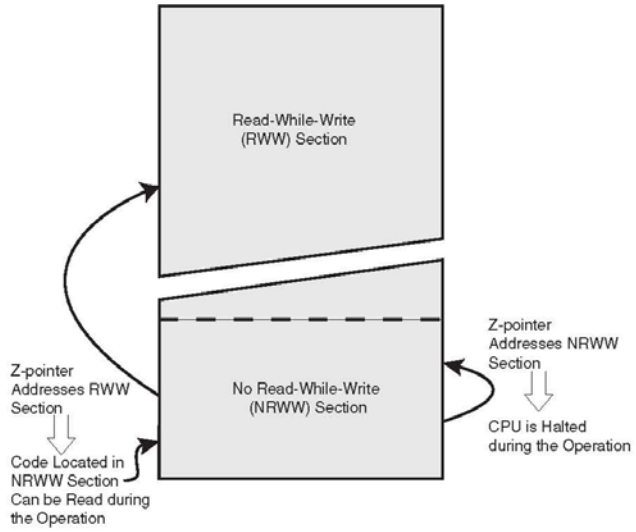
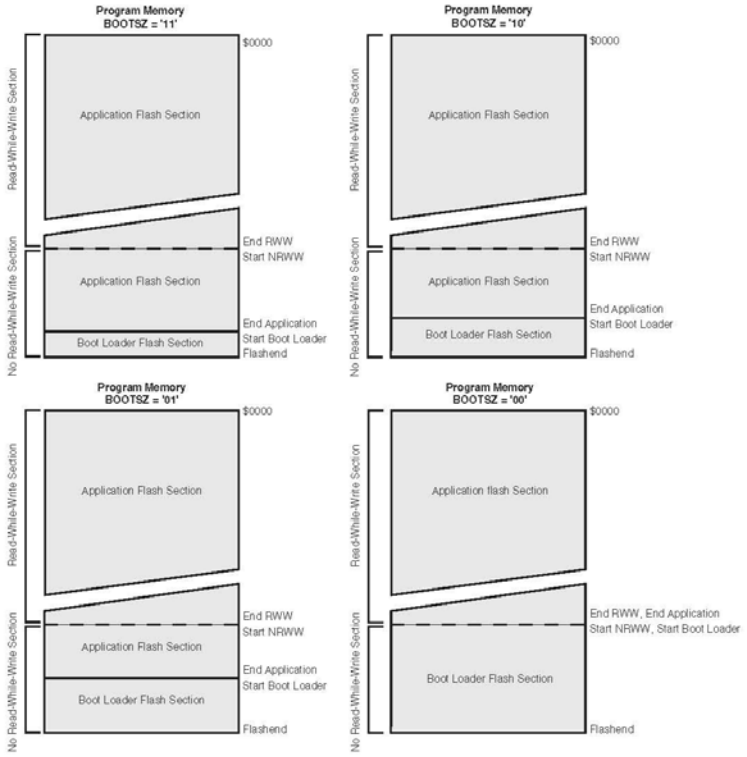


Figure 125. Memory Sections[1]



Note: 1. The parameters in the figure above are given in Table 100 on page 257.

Boot Loader Lock Bits

If no Boot Loader capability is needed, the entire Flash is available for application code. The Boot Loader has two separate sets of Boot Lock bits which can be set independently. This gives the user a unique flexibility to select different levels of protection.

The user can select:

- To protect the entire Flash from a software update by the MCU
- To protect only the Boot Loader Flash section from a software update by the MCU
- To protect only the Application Flash section from a software update by the MCU
- Allow software update in the entire Flash

See Table 96 and Table 97 for further details. The Boot Lock bits can be set in software and in Serial or Parallel Programming mode, but they can be cleared by a Chip Erase command only. The general Write Lock (Lock Bit mode 2) does not control the programming of the Flash memory by SPM instruction. Similarly, the general Read/Write Lock (Lock Bit mode 3) does not control reading nor writing by LPM/SPM, if it is attempted.

Table 96. Boot Lock Bit0 Protection Modes (Application Section)[1]

BLB0 Mode	BLB02	BLB01	Protection
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If interrupt vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If interrupt vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.

Note: 1. "1" means unprogrammed, "0" means programmed

Table 97. Boot Lock Bit1 Protection Modes (Boot Loader Section)[1]

BLB1 mode	BLB12	BLB11	Protection
1	1	1	No restrictions for SPM or LPM accessing the Boot Loader section.
2	1	0	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If interrupt vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If interrupt vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Note: 1. "1" means unprogrammed, "0" means programmed

Entering the Boot Loader Program

Entering the Boot Loader takes place by a jump or call from the application program. This may be initiated by a trigger such as a command received via USART, or SPI interface. Alternatively, the Boot Reset Fuse can be programmed so that the Reset Vector is pointing to the Boot Flash start address after a reset. In this case, the Boot Loader is started after a reset. After the application code is loaded, the program can start executing the application code. Note that the fuses cannot be changed by the MCU itself. This means that once the Boot Reset Fuse is programmed, the Reset Vector will always point to the Boot Loader Reset and the fuse can only be changed through the serial or parallel programming interface.

Table 98. Boot Reset Fuse[1]

BOOTRST	Reset Address
1	Reset Vector = Application reset (address $0000)
0	Reset Vector = Boot Loader reset (see Table 100 on page 257)

Note: 1. "1" means unprogrammed, "0" means programmed

Store Program Memory Control Register – SPMCR

The Store Program Memory Control Register contains the control bits needed to control the Boot Loader operations.

Bit	7	6	5	4	3	2	1	0	
	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	SPMCR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPMIE: SPM Interrupt Enable**

When the SPMIE bit is written to one, and the I-bit in the Status Register is set (one), the SPM ready interrupt will be enabled. The SPM ready Interrupt will be executed as long as the SPMEN bit in the SPMCR Register is cleared.

- **Bit 6 – RWWSB: Read-While-Write Section Busy**

When a self-programming (Page Erase or Page Write) operation to the RWW section is initiated, the RWWSB will be set (one) by hardware. When the RWWSB bit is set, the RWW section cannot be accessed. The RWWSB bit will be cleared if the RWWSRE bit is written to one after a Self-Programming operation is completed. Alternatively the RWWSB bit will automatically be cleared if a page load operation is initiated.

- **Bit 5 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega16 and always read as zero.

- **Bit 4 – RWWSRE: Read-While-Write Section Read Enable**

When programming (Page Erase or Page Write) to the RWW section, the RWW section is blocked for reading (the RWWSB will be set by hardware). To re-enable the RWW section, the user software must wait until the programming is completed (SPMEN will be cleared). Then, if the RWWSRE bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles re-enables the RWW section. The RWW section cannot be re-enabled while the Flash is busy with a page erase or a page write (SPMEN is set). If the RWWSRE bit is written while the Flash is being loaded, the Flash load operation will abort and the data loaded will be lost.

- **Bit 3 – BLBSET: Boot Lock Bit Set**

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles sets Boot Lock bits, according to the data in R0. The data in R1 and the address in the Z-pointer are ignored. The BLBSET bit will automatically be cleared upon completion of the Lock bit set, or if no SPM instruction is executed within four clock cycles.

An LPM instruction within three cycles after BLBSET and SPMEN are set in the SPMCR Register, will read either the Lock bits or the Fuse bits (depending on Z0 in the Z-pointer) into the destination register. See "Reading the Fuse and Lock Bits from Software" on page 254 for details.

- **Bit 2 – PGWRT: Page Write**

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles executes Page Write, with the data stored in the temporary buffer. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGWRT bit will auto-clear upon completion of a page write, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire page write operation if the NRWW section is addressed.

- **Bit 1 – PGERS: Page Erase**

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles executes Page Erase. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGERS bit will auto-clear upon completion of a page erase, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire page write operation if the NRWW section is addressed.

- **Bit 0 – SPMEN: Store Program Memory Enable**

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either RWWSRE, BLBSET, PGWRT' or PGERS, the following SPM instruction will have a special meaning, see description above. If only SPMEN is written, the following SPM instruction will store the value in R1:R0 in the temporary page buffer addressed by the Z-pointer. The LSB of the Z-pointer is ignored. The SPMEN bit will auto-clear upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During page erase and page write, the SPMEN bit remains high until the operation is completed.

Writing any other combination than "10001", "01001", "00101", "00011" or "00001" in the lower five bits will have no effect.

Addressing the Flash during Self-Programming

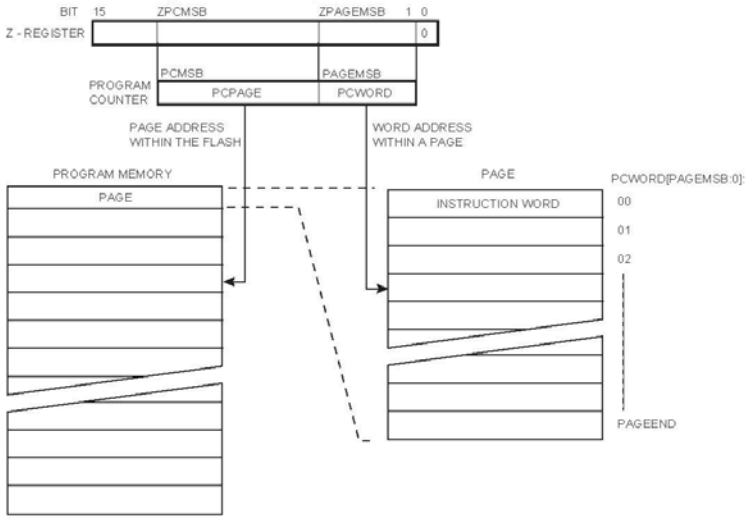
The Z-pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	Z15	Z14	Z13	Z12	Z11	Z10	Z9	Z8
ZL (R30)	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
	7	6	5	4	3	2	1	0

Since the Flash is organized in pages (see Table 107 on page 262), the Program Counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in Figure 126. Note that the Page Erase and Page Write operations are addressed independently. Therefore it is of major importance that the Boot Loader software addresses the same page in both the Page Erase and Page Write operation. Once a programming operation is initiated, the address is latched and the Z-pointer can be used for other operations.

The only SPM operation that does not use the Z-pointer is Setting the Boot Loader Lock bits. The content of the Z-pointer is ignored and will have no effect on the operation. The LPM instruction does also use the Z pointer to store the address. Since this instruction addresses the Flash byte by byte, also the LSB (bit Z0) of the Z-pointer is used.

Figure 126. Addressing the Flash during SPM[1]



Notes: 1. The different variables used in Figure 126 are listed in Table 102 on page 258.
2. PCPAGE and PCWORD are listed in Table 107 on page 262.

Self-Programming the Flash

The program memory is updated in a page by page fashion. Before programming a page with the data stored in the temporary page buffer, the page must be erased. The temporary page buffer is filled one word at a time using SPM and the buffer can be filled either before the page erase command or between a page erase and a page write operation:

Alternative 1, fill the buffer before a Page Erase

- Fill temporary page buffer
- Perform a Page Erase
- Perform a Page Write

Alternative 2, fill the buffer after Page Erase

- Perform a Page Erase
- Fill temporary page buffer
- Perform a Page Write

If only a part of the page needs to be changed, the rest of the page must be stored (for example in the temporary page buffer) before the erase, and then be rewritten. When using alternative 1, the Boot Loader provides an effective Read-Modify-Write feature which allows the user software to first read the page, do the necessary changes, and then write back the modified data. If alternative 2 is used, it is not possible to read the old data while loading since the page is already erased. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the page erase and page write operation is addressing the same page. See "Simple Assembly Code Example for a Boot Loader" on page 256 for an assembly code example.

Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write "X0000011" to SPMCR and execute SPM within four clock cycles after writing SPMCR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE in the Z-register. Other bits in the Z-pointer must be written zero during this operation.

- Page Erase to the RWW section: The NRWW section can be read during the page erase.
- Page Erase to the NRWW section: The CPU is halted during the operation.

Filling the Temporary Buffer (Page Loading)

To write an instruction word, set up the address in the Z-pointer and data in R1:R0, write "00000001" to SPMCR and execute SPM within four clock cycles after writing SPMCR. The content of PCWORD in the Z-register is used to address the data in the temporary buffer. The temporary buffer will auto-erase after a page write operation or by writing the RWWSRE bit in SPMCR. It is also erased after a system reset. Note that it is not possible to write more than one time to each address without erasing the temporary buffer.

Note: If the EEPROM is written in the middle of an SPM Page Load operation, all data loaded will be lost.

Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write "X0000101" to SPMCR and execute SPM within four clock cycles after writing SPMCR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE. Other bits in the Z-pointer must be written zero during this operation.

- Page Write to the RWW section: The NRWW section can be read during the Page Write.
- Page Write to the NRWW section: The CPU is halted during the operation.

Using the SPM Interrupt

If the SPM interrupt is enabled, the SPM interrupt will generate a constant interrupt when the SPMEN bit in SPMCR is cleared. This means that the interrupt can be used instead of polling the SPMCR Register in software. When using the SPM interrupt, the Interrupt Vectors should be moved to the BLS section to avoid that an interrupt is accessing the RWW section when it is blocked for reading. How to move the interrupts is described in "Interrupts" on page 45.

Consideration while Updating BLS

Special care must be taken if the user allows the Boot Loader section to be updated by leaving Boot Lock bit11 unprogrammed. An accidental write to the Boot Loader itself can corrupt the entire Boot Loader, and further software updates might be impossible. If it is not necessary to change the Boot Loader software itself, it is recommended to program the Boot Lock bit11 to protect the Boot Loader software from any internal software changes.

Prevent Reading the RWW Section during Self-Programming

During Self-Programming (either Page Erase or Page Write), the RWW section is always blocked for reading. The user software itself must prevent that this section is addressed during the Self-Programming operation. The RWWSB in the SPMCR will be set as long as the RWW section is busy. During self-programming the Interrupt Vector table should be moved to the BLS as described in "Interrupts" on page 45, or the interrupts must be disabled. Before addressing the RWW section after the programming is completed, the user software must clear the RWWSB by writing the RWWSRE. See "Simple Assembly Code Example for a Boot Loader" on page 256 for an example.

Setting the Boot Loader Lock Bits by SPM

To set the Boot Loader Lock bits, write the desired data to R0, write "X0001001" to SPMCR and execute SPM within four clock cycles after writing SPMCR. The only accessible Lock bits are the Boot Lock bits that may prevent the Application and Boot Loader section from any software update by the MCU.

Bit	7	6	5	4	3	2	1	0
R0	1	1	BLB12	BLB11	BLB02	BLB01	1	1

See Table 96 and Table 97 for how the different settings of the Boot Loader bits affect the Flash access.

If bits 5..2 in R0 are cleared (zero), the corresponding Boot Lock bit will be programmed if an SPM instruction is executed within four cycles after BLBSET and SPMEN are set in SPMCR. The Z-pointer is don't care during this operation, but for future compatibility it is recommended to load the Z-pointer with $0001 (same as used for reading the Lock bits). For future compatibility It is also recommended to set bits 7, 6, 1, and 0 in R0 to "1" when writing the Lock bits. When programming the Lock bits the entire Flash can be read during the operation.

EEPROM Write Prevents Writing to SPMCR

Note that an EEPROM write operation will block all software programming to Flash. Reading the Fuses and Lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEWE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCR Register.

Reading the Fuse and Lock Bits from Software

It is possible to read both the Fuse and Lock bits from software. To read the Lock bits, load the Z-pointer with $0001 and set the BLBSET and SPMEN bits in SPMCR. When an LPM instruction is executed within three CPU cycles after the BLBSET and SPMEN bits are set in SPMCR, the value of the Lock bits will be loaded in the destination register. The BLBSET and SPMEN bits will auto-clear upon completion of reading the Lock bits or if no LPM instruction is executed within three CPU cycles or no SPM instruction is executed within four CPU cycles. When BLBSET and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Bit	7	6	5	4	3	2	1	0
Rd	–	–	BLB12	BLB11	BLB02	BLB01	LB2	LB1

The algorithm for reading the Fuse Low bits is similar to the one described above for reading the Lock bits. To read the Fuse Low bits, load the Z-pointer with $0000 and set the BLBSET and SPMEN bits in SPMCR. When an LPM instruction is executed within three cycles after the BLBSET and SPMEN bits are set in the SPMCR, the value of the Fuse Low bits (FLB) will be loaded in the destination register as shown below. Refer to Table 106 on page 261 for a detailed description and mapping of the Fuse Low bits.

Bit	7	6	5	4	3	2	1	0
Rd	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

Similarly, when reading the Fuse High bits, load $0003 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SPMEN bits are set in the SPMCR, the value of the Fuse High bits (FHB) will be loaded in the destination register as shown below. Refer to Table 105 on page 260 for detailed description and mapping of the Fuse High bits.

Bit	7	6	5	4	3	2	1	0
Rd	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

Fuse and Lock bits that are programmed, will be read as zero. Fuse and Lock bits that are unprogrammed, will be read as one.

Preventing Flash Corruption

During periods of low V_{CC}, the Flash program can be corrupted because the supply voltage is too low for the CPU and the Flash to operate properly. These issues are the same as for board level systems using the Flash, and the same design solutions should be applied.

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the Flash requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. If there is no need for a Boot Loader update in the system, program the Boot Loader Lock bits to prevent any Boot Loader software updates.
2. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} Reset Protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
3. Keep the AVR core in Power-down Sleep mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCR Register and thus the Flash from unintentional writes.

Programming Time for Flash when using SPM

The Calibrated RC Oscillator is used to time Flash accesses. Table 99 shows the typical programming time for Flash accesses from the CPU.

Table 99. SPM Programming Time.

Symbol	Min Programming Time	Max Programming Time
Flash write (Page Erase, Page Write, and write Lock bits by SPM)	3.7 ms	4.5 ms

Simple Assembly Code Example for a Boot Loader

```
  ;-the routine writes one page of data from RAM to Flash
  ; the first data location in RAM is pointed to by the Y pointer
  ; the first data location in Flash is pointed to by the Z pointer
  ;-error handling is not included
  ;-the routine must be placed inside the boot space
  ; (at least the Do_spm sub routine). Only code inside NRWW section can
  ; be read during self-programming (page erase and page write).
  ;-registers used: r0, r1, temp1 (r16), temp2 (r17), looplo (r24),
  ; loophi (r25), spmcrval (r20)
  ; storing and restoring of registers is not included in the routine
  ; register usage can be optimized at the expense of code size
  ;-It is assumed that either the interrupt table is moved to the Boot
  ; loader section or that the interrupts are disabled.
.equ  PAGESIZEB = PAGESIZE*2        ; PAGESIZEB is page size in BYTES, not
                                    ; words
.org SMALLBOOTSTART
Write_page:
  ; page erase
  ldi   spmcrval, (1<<PGERS) | (1<<SPMEN)
  call  Do_spm

  ; re-enable the RWW section
  ldi   spmcrval, (1<<RWWSRE) | (1<<SPMEN)
  call  Do_spm

  ; transfer data from RAM to Flash page buffer
  ldi   looplo, low(PAGESIZEB)      ;init loop variable
  ldi   loophi, high(PAGESIZEB)     ;not required for PAGESIZEB<=256
Wrloop:
  ld    r0, Y+
  ld    r1, Y+
  ldi   spmcrval, (1<<SPMEN)
  call  Do_spm
  adiw  ZH:ZL, 2
  sbiw  loophi:looplo, 2            ;use subi for PAGESIZEB<=256
  brne  Wrloop

  ; execute page write
  subi  ZL, low(PAGESIZEB)          ;restore pointer
  sbci  ZH, high(PAGESIZEB)         ;not required for PAGESIZEB<=256
  ldi   spmcrval, (1<<PGWRT) | (1<<SPMEN)
  call  Do_spm

  ; re-enable the RWW section
  ldi   spmcrval, (1<<RWWSRE) | (1<<SPMEN)
  call  Do_spm

  ; read back and check, optional
  ldi   looplo, low(PAGESIZEB)      ;init loop variable
  ldi   loophi, high(PAGESIZEB)     ;not required for PAGESIZEB<=256
  subi  YL, low(PAGESIZEB)          ;restore pointer
  sbci  YH, high(PAGESIZEB)
Rdloop:
  lpm   r0, Z+
  ld    r1, Y+
  cpse  r0, r1
  jmp   Error
  sbiw  loophi:looplo, 1            ;use subi for PAGESIZEB<=256
  brne  Rdloop

  ; return to RWW section
  ; verify that RWW section is safe to read
Return:
  in    temp1, SPMCR
```

```
  sbrs  temp1, RWWSB              ; If RWWSB is set, the RWW section is not
                                  ; ready yet
  ret
  ; re-enable the RWW section
  ldi   spmcrval, (1<<RWWSRE) | (1<<SPMEN)
  call  Do_spm
  rjmp  Return

Do_spm:
  ; check for previous SPM complete
Wait_spm:
  in    temp1, SPMCR
  sbrc  temp1, SPMEN
  rjmp  Wait_spm
  ; input: spmcrval determines SPM action
  ; disable interrupts if enabled, store status
  in    temp2, SREG
  cli
  ; check that no EEPROM write access is present
Wait_ee:
  sbic  EECR, EEWE
  rjmp  Wait_ee
  ; SPM timed sequence
  out   SPMCR, spmcrval
  spm
  ; restore SREG (to enable interrupts if originally enabled)
  out   SREG, temp2
  ret
```

ATmega16 Boot Loader Parameters

In Table 100 through Table 102, the parameters used in the description of the self programming are given.

Table 100. Boot Size Configuration[1]

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application section	Boot Reset Address (start Boot Loader Section)
1	1	128 words	2	$0000 - $1F7F	$1F80 - $1FFF	$1F7F	$1F80
1	0	256 words	4	$0000 - $1EFF	$1F00 - $1FFF	$1EFF	$1F00
0	1	512 words	8	$0000 - $1DFF	$1E00 - $1FFF	$1DFF	$1E00
0	0	1024 words	16	$0000 - $1BFF	$1C00 - $1FFF	$1BFF	$1C00

Note: 1. The different BOOTSZ Fuse configurations are shown in Figure 125

Table 101. Read-While-Write Limit[1]

Section	Pages	Address
Read-While-Write section (RWW)	112	$0000 - $1BFF
No Read-While-Write section (NRWW)	16	$1C00 - $1FFF

Note: 1. For details about these two section, see "NRWW – No Read-While-Write Section" on page 247 and "RWW – Read-While-Write Section" on page 247

Table 102. Explanation of Different Variables used in Figure 126 and the Mapping to the Z-pointer

Variable		Corresponding Z-value[1]	Description
PCMSB	12		Most significant bit in the Program Counter. (The Program Counter is 13 bits PC[12:0])
PAGEMSB	5		Most significant bit which is used to address the words within one page (64 words in a page requires 6 bits PC [5:0]).
ZPCMSB		Z13	Bit in Z-register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z6	Bit in Z-register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.
PCPAGE	PC[12:6]	Z13:Z7	Program Counter page address: Page select, for Page Erase and Page Write
PCWORD	PC[5:0]	Z6:Z1	Program Counter word address: Word select, for filling temporary buffer (must be zero during page write operation)

Note: 1. Z15:Z14: always ignored
Z0: should be zero for all SPM commands, byte select for the LPM instruction.
See "Addressing the Flash during Self-Programming" on page 251 for details about the use of Z-pointer during Self-Programming.

Memory Programming

Program And Data Memory Lock Bits

The ATmega16 provides six Lock bits which can be left unprogrammed ("1") or can be programmed ("0") to obtain the additional features listed in Table 104. The Lock bits can only be erased to "1" with the Chip Erase command.

Table 103. Lock Bit Byte[1]

Lock Bit Byte	Bit No.	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12	5	Boot Lock bit	1 (unprogrammed)
BLB11	4	Boot Lock bit	1 (unprogrammed)
BLB02	3	Boot Lock bit	1 (unprogrammed)
BLB01	2	Boot Lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Note: 1. "1" means unprogrammed, "0" means programmed

Table 104. Lock Bit Protection Modes

Memory Lock Bits[2]			Protection Type
LB Mode	LB2	LB1	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in Parallel and SPI/JTAG Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and SPI/JTAG Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]
BLB0 Mode	**BLB02**	**BLB01**	
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If interrupt vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If interrupt vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.

Table 104. Lock Bit Protection Modes (Continued)

Memory Lock Bits[2]			Protection Type
BLB1 Mode	BLB12	BLB11	
1	1	1	No restrictions for SPM or LPM accessing the Boot Loader section.
2	1	0	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If interrupt vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If interrupt vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Notes: 1. Program the Fuse bits before programming the Lock bits.
2. "1" means unprogrammed, "0" means programmed

Fuse Bits

The ATmega16 has two fuse bytes. Table 105 and Table 106 describe briefly the functionality of all the fuses and how they are mapped into the fuse bytes. Note that the fuses are read as logical zero, "0", if they are programmed.

Table 105. Fuse High Byte

Fuse High Byte	Bit No.	Description	Default Value
OCDEN[4]	7	Enable OCD	1 (unprogrammed, OCD disabled)
JTAGEN[5]	6	Enable JTAG	0 (programmed, JTAG enabled)
SPIEN[1]	5	Enable SPI Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
CKOPT[2]	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BOOTSZ1	2	Select Boot Size (see Table 100 for details)	0 (programmed)[3]
BOOTSZ0	1	Select Boot Size (see Table 100 for details)	0 (programmed)[3]
BOOTRST	0	Select reset vector	1 (unprogrammed)

Notes: 1. The SPIEN Fuse is not accessible in SPI Serial Programming mode.
2. The CKOPT Fuse functionality depends on the setting of the CKSEL bits. See See "Clock Sources" on page 25. for details.
3. The default value of BOOTSZ1..0 results in maximum Boot Size. See Table 100 on page 257.
4. Never ship a product with the OCDEN Fuse programmed regardless of the setting of Lock bits and the JTAGEN Fuse. A programmed OCDEN Fuse enables some parts of the clock system to be running in all sleep modes. This may increase the power consumption.
5. If the JTAG interface is left unconnected, the JTAGEN fuse should if possible be disabled. This to avoid static current at the TDO pin in the JTAG interface.

Table 106. Fuse Low Byte

Fuse Low Byte	Bit No.	Description	Default Value
BODLEVEL	7	Brown-out Detector trigger level	1 (unprogrammed)
BODEN	6	Brown-out Detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed)(1)
SUT0	4	Select start-up time	0 (programmed)(1)
CKSEL3	3	Select Clock source	0 (programmed)(2)
CKSEL2	2	Select Clock source	0 (programmed)(2)
CKSEL1	1	Select Clock source	0 (programmed)(2)
CKSEL0	0	Select Clock source	1 (unprogrammed)(2)

Notes: 1. The default value of SUT1..0 results in maximum start-up time. SeeTable 10 on page 29 for details.
2. The default setting of CKSEL3..0 results in internal RC Oscillator @ 1 MHz. See Table 2 on page 25 for details.

The status of the Fuse bits is not affected by Chip Erase. Note that the Fuse bits are locked if Lock bit1 (LB1) is programmed. Program the Fuse bits before programming the Lock bits.

Latching of Fuses

The Fuse values are latched when the device enters programming mode and changes of the Fuse values will have no effect until the part leaves Programming mode. This does not apply to the EESAVE Fuse which will take effect once it is programmed. The fuses are also latched on Power-up in Normal mode.

Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial and parallel mode, also when the device is locked. The three bytes reside in a separate address space.

For the ATmega16 the signature bytes are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $94 (indicates 16 Kbytes Flash memory)
3. $002: $03 (indicates ATmega16 device when $001 is $94)

Calibration Byte

The ATmega16 stores four different calibration values for the internal RC Oscillator. These bytes resides in the signature row High Byte of the addresses 0x0000, 0x0001, 0x0002, and 0x0003 for 1 MHz, 2 MHz, 4 MHz, and 8 Mhz respectively. During Reset, the 1 MHz value is automatically loaded into the OSCCAL Register. If other frequencies are used, the calibration value has to be loaded manually, see "Oscillator Calibration Register – OSCCAL" on page 30 for details.

Page Size

Table 107. No. of Words in a Page and no. of Pages in the Flash

Flash Size	Page Size	PCWORD	No. of Pages	PCPAGE	PCMSB
8K words (16 Kbytes)	64 words	PC[5:0]	128	PC[12:6]	12

Table 108. No. of Words in a Page and no. of Pages in the EEPROM

EEPROM Size	Page Size	PCWORD	No. of Pages	PCPAGE	EEAMSB
512 bytes	4 bytes	EEA[1:0]	128	EEA[8:2]	8

Parallel Programming Parameters, Pin Mapping, and Commands

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory, Memory Lock bits, and Fuse bits in the ATmega16. Pulses are assumed to be at least 250 ns unless otherwise noted.

Signal Names

In this section, some pins of the ATmega16 are referenced by signal names describing their functionality during parallel programming, see Figure 127 and Table 109. Pins not described in the following table are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding is shown in Table 111.

When pulsing $\overline{WR}$ or $\overline{OE}$, the command loaded determines the action executed. The different Commands are shown in Table 112.

Figure 127. Parallel Programming

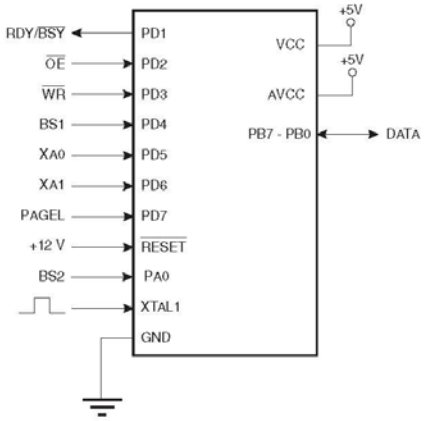


Table 109. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/$\overline{BSY}$	PD1	O	0: Device is busy programming, 1: Device is ready for new command
$\overline{OE}$	PD2	I	Output Enable (Active low)
$\overline{WR}$	PD3	I	Write Pulse (Active low)
BS1	PD4	I	Byte Select 1 ("0" selects Low byte, "1" selects High byte)
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
PAGEL	PD7	I	Program Memory and EEPROM data Page Load
BS2	PA0	I	Byte Select 2 ("0" selects Low byte, "1" selects 2'nd High byte)
DATA	PB7-0	I/O	Bidirectional Data bus (Output when $\overline{OE}$ is low)

Table 110. Pin Values used to Enter Programming Mode

Pin	Symbol	Value
PAGEL	Prog_enable[3]	0
XA1	Prog_enable[2]	0
XA0	Prog_enable[1]	0
BS1	Prog_enable[0]	0

Table 111. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or low address byte determined by BS1)
0	1	Load Data (High or Low data byte for Flash determined by BS1)
1	0	Load Command
1	1	No Action, Idle

Table 112. Command Byte Bit Coding

Command Byte	Command Executed
1000 0000	Chip Erase
0100 0000	Write Fuse Bits
0010 0000	Write Lock Bits
0001 0000	Write Flash
0001 0001	Write EEPROM

Table 112. Command Byte Bit Coding

Command Byte	Command Executed
0000 1000	Read Signature Bytes and Calibration byte
0000 0100	Read Fuse and Lock bits
0000 0010	Read Flash
0000 0011	Read EEPROM

Parallel Programming

Enter Programming Mode

The following algorithm puts the device in Parallel Programming mode:

1. Apply 4.5V - 5.5V between V_{CC} and GND, and wait at least 100 µs.
2. Set $\overline{\text{RESET}}$ to "0" and toggle XTAL1 at least 6 times
3. Set the Prog_enable pins listed in Table 110 on page 263 to "0000" and wait at least 100 ns.
4. Apply 11.5V - 12.5V to $\overline{\text{RESET}}$. Any activity on Prog_enable pins within 100 ns after +12V has been applied to $\overline{\text{RESET}}$, will cause the device to fail entering Programming mode.

Note, if External Crystal or External RC configuration is selected, it may not be possible to apply qualified XTAL1 pulses. In such cases, the following algorithm should be followed:

1. Set Prog_enable pins listed in Table 110 on page 263 to "0000".
2. Apply 4.5V - 5.5V between V_{CC} and GND simultaneously as 11.5V - 12.5V is applied to $\overline{\text{RESET}}$.
3. Wait 100 µs.
4. Re-program the fuses to ensure that External Clock is selected as clock source (CKSEL3:0 = 0b0000) If Lock bits are programmed, a Chip Erase command must be executed before changing the fuses.
5. Exit Programming mode by power the device down or by bringing $\overline{\text{RESET}}$ pin to 0b0.
6. Entering Programming mode with the original algorithm, as described above.

Considerations for Efficient Programming

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command needs only be loaded once when writing or reading multiple memory locations.
- Skip writing the data value $FF, that is the contents of the entire EEPROM (unless the EESAVE Fuse is programmed) and Flash after a Chip Erase.
- Address High byte needs only be loaded before programming or reading a new 256 word window in Flash or 256 byte EEPROM. This consideration also applies to Signature bytes reading.

Chip Erase

The Chip Erase will erase the Flash and EEPROM[1] memories plus Lock bits. The Lock bits are not reset until the program memory has been completely erased. The Fuse bits are not changed. A Chip Erase must be performed before the Flash and/or the EEPROM are reprogrammed.

Note: 1. The EEPRPOM memory is preserved during chip erase if the EESAVE Fuse is programmed.

Load Command "Chip Erase"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS1 to "0".
3. Set DATA to "1000 0000". This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give $\overline{\text{WR}}$ a negative pulse. This starts the Chip Erase. RDY/$\overline{\text{BSY}}$ goes low.
6. Wait until RDY/$\overline{\text{BSY}}$ goes high before loading a new command.

Programming the Flash

The Flash is organized in pages, see Table 107 on page 262. When programming the Flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory:

A. Load Command "Write Flash"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS1 to "0".
3. Set DATA to "0001 0000". This is the command for Write Flash.
4. Give XTAL1 a positive pulse. This loads the command.

B. Load Address Low byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS1 to "0". This selects low address.
3. Set DATA = Address Low byte ($00 - $FF).
4. Give XTAL1 a positive pulse. This loads the address Low byte.

C. Load Data Low Byte

1. Set XA1, XA0 to "01". This enables data loading.
2. Set DATA = Data Low byte ($00 - $FF).
3. Give XTAL1 a positive pulse. This loads the data byte.

D. Load Data High Byte

1. Set BS1 to "1". This selects high data byte.
2. Set XA1, XA0 to "01". This enables data loading.
3. Set DATA = Data High byte ($00 - $FF).
4. Give XTAL1 a positive pulse. This loads the data byte.

E. Latch Data

1. Set BS1 to "1". This selects high data byte.
2. Give PAGEL a positive pulse. This latches the data bytes. (See Figure 129 for signal waveforms)

F. Repeat B through E until the entire buffer is filled or until all data within the page is loaded.

While the lower bits in the address are mapped to words within the page, the higher bits address the pages within the FLASH. This is illustrated in Figure 128 on page 267. Note that if less than 8 bits are required to address words in the page (pagesize < 256), the most significant bit(s) in the address Low byte are used to address the page when performing a page write.

G. Load Address High byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS1 to "1". This selects high address.
3. Set DATA = Address High byte ($00 - $FF).
4. Give XTAL1 a positive pulse. This loads the address High byte.

H. Program Page

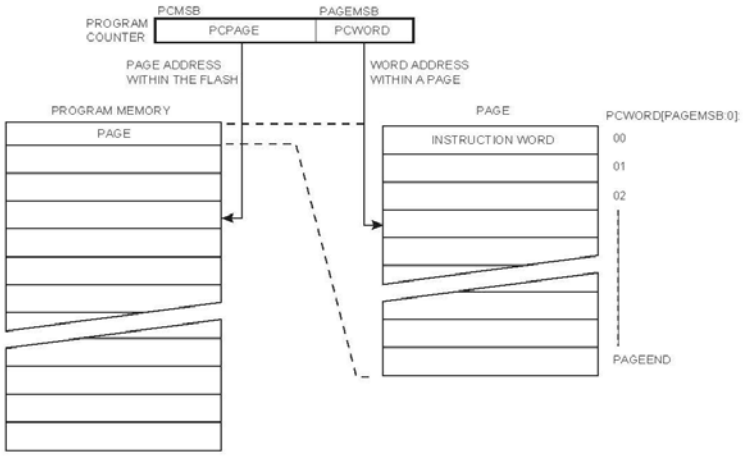
1. Set BS1 = "0"
2. Give $\overline{WR}$ a negative pulse. This starts programming of the entire page of data. RDY/$\overline{BSY}$ goes low.
3. Wait until RDY/$\overline{BSY}$ goes high. (See Figure 129 for signal waveforms)

I. Repeat B through H until the entire Flash is programmed or until all data has been programmed.

J. End Page Programming

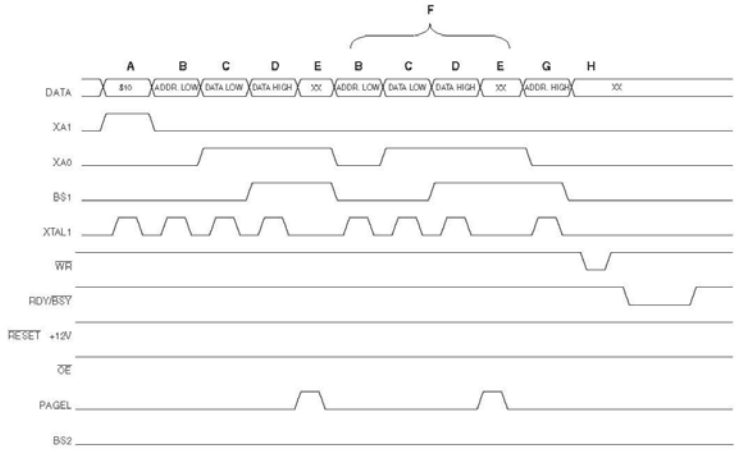
1. 1. Set XA1, XA0 to "10". This enables command loading.
2. Set DATA to "0000 0000". This is the command for No Operation.
3. Give XTAL1 a positive pulse. This loads the command, and the internal write signals are reset.

Figure 128. Addressing the Flash which is Organized in Pages



Note: 1. PCPAGE and PCWORD are listed in Table 107 on page 262.

Figure 129. Programming the Flash Waveforms[1]



Note: 1. "XX" is don't care. The letters refer to the programming description above.

Programming the EEPROM

The EEPROM is organized in pages, see Table 108 on page 262. When programming the EEPROM, the program data is latched into a page buffer. This allows one page of data to be programmed simultaneously. The programming algorithm for the EEPROM data memory is as follows (refer to "Programming the Flash" on page 266 for details on Command, Address and Data loading):

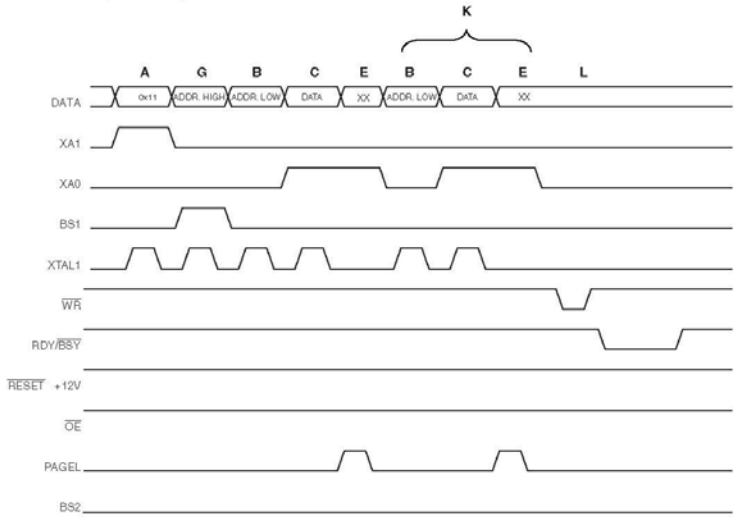
1. A: Load Command "0001 0001".
2. G: Load Address High Byte ($00 - $FF)
3. B: Load Address Low Byte ($00 - $FF)
4. C: Load Data ($00 - $FF)
5. E: Latch data (give PAGEL a positive pulse)

K: Repeat 3 through 5 until the entire buffer is filled

L: Program EEPROM page

1. Set BS1 to "0".
2. Give $\overline{WR}$ a negative pulse. This starts programming of the EEPROM page. RDY/$\overline{BSY}$ goes low.
3. Wait until to RDY/$\overline{BSY}$ goes high before programming the next page. (See Figure 130 for signal waveforms)

Figure 130. Programming the EEPROM Waveforms



Reading the Flash

The algorithm for reading the Flash memory is as follows (refer to "Programming the Flash" on page 266 for details on Command and Address loading):

1. A: Load Command "0000 0010".
2. G: Load Address High Byte ($00 - $FF)
3. B: Load Address Low Byte ($00 - $FF)
4. Set $\overline{OE}$ to "0", and BS1 to "0". The Flash word Low byte can now be read at DATA.
5. Set BS1 to "1". The Flash word High byte can now be read at DATA.
6. Set $\overline{OE}$ to "1".

Reading the EEPROM

The algorithm for reading the EEPROM memory is as follows (refer to "Programming the Flash" on page 266 for details on Command and Address loading):

1. A: Load Command "0000 0011".
2. G: Load Address High Byte ($00 - $FF)
3. B: Load Address Low Byte ($00 - $FF)
4. Set $\overline{OE}$ to "0", and BS1 to "0". The EEPROM Data byte can now be read at DATA.
5. Set $\overline{OE}$ to "1".

Programming the Fuse Low Bits

The algorithm for programming the Fuse Low bits is as follows (refer to "Programming the Flash" on page 266 for details on Command and Data loading):

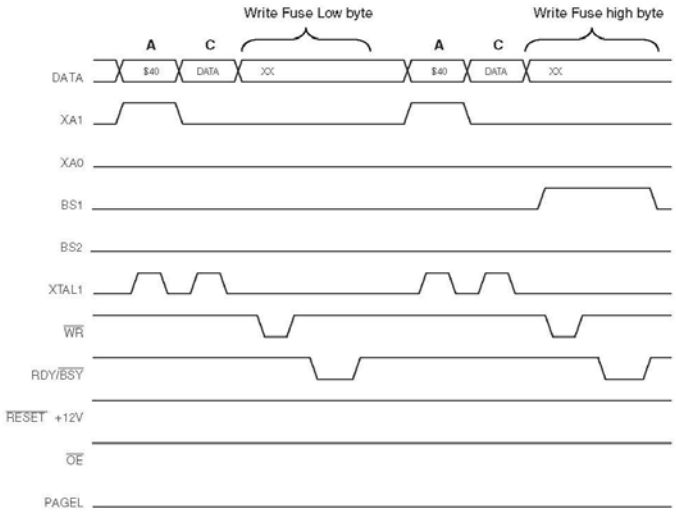
1. A: Load Command "0100 0000".
2. C: Load Data Low Byte. Bit n = "0" programs and bit n = "1" erases the Fuse bit.
3. Set BS1 to "0" and BS2 to "0".
4. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.

Programming the Fuse High Bits

The algorithm for programming the Fuse high bits is as follows (refer to "Programming the Flash" on page 266 for details on Command and Data loading):

1. A: Load Command "0100 0000".
2. C: Load Data Low Byte. Bit n = "0" programs and bit n = "1" erases the Fuse bit.
3. Set BS1 to "1" and BS2 to "0". This selects high data byte.
4. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.
5. Set BS1 to "0". This selects low data byte.

Figure 131. Programming the Fuses



Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (refer to "Programming the Flash" on page 266 for details on Command and Data loading):

1. A: Load Command "0010 0000".
2. C: Load Data Low Byte. Bit n = "0" programs the Lock bit.
3. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.

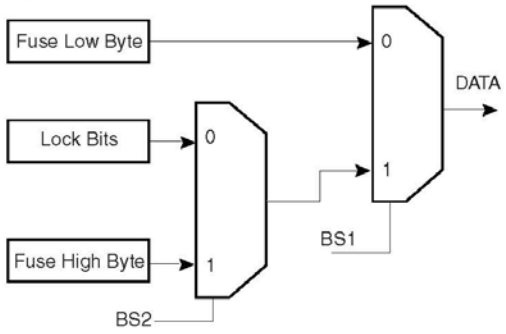
The Lock bits can only be cleared by executing Chip Erase.

Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (refer to "Programming the Flash" on page 266 for details on Command loading):

1. A: Load Command "0000 0100".
2. Set $\overline{OE}$ to "0", BS2 to "0" and BS1 to "0". The status of the Fuse Low bits can now be read at DATA ("0" means programmed).
3. Set $\overline{OE}$ to "0", BS2 to "1" and BS1 to "1". The status of the Fuse High bits can now be read at DATA ("0" means programmed).
4. Set $\overline{OE}$ to "0", BS2 to "0" and BS1 to "1". The status of the Lock bits can now be read at DATA ("0" means programmed).
5. Set $\overline{OE}$ to "1".

Figure 132. Mapping between BS1, BS2 and the Fuse- and Lock Bits during Read



Reading the Signature Bytes

The algorithm for reading the Signature bytes is as follows (refer to "Programming the Flash" on page 266 for details on Command and Address loading):

1. A: Load Command "0000 1000".
2. B: Load Address Low Byte ($00 - $02).
3. Set $\overline{OE}$ to "0", and BS1 to "0". The selected Signature byte can now be read at DATA.
4. Set $\overline{OE}$ to "1".

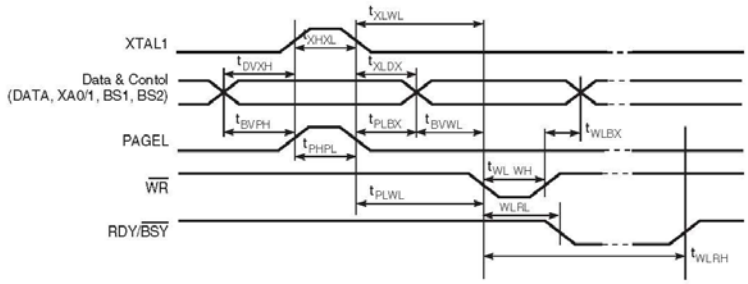
Reading the Calibration Byte

The algorithm for reading the Calibration byte is as follows (refer to "Programming the Flash" on page 266 for details on Command and Address loading):

1. A: Load Command "0000 1000".
2. B: Load Address Low Byte, $00.
3. Set $\overline{OE}$ to "0", and BS1 to "1". The Calibration byte can now be read at DATA.
4. Set $\overline{OE}$ to "1".

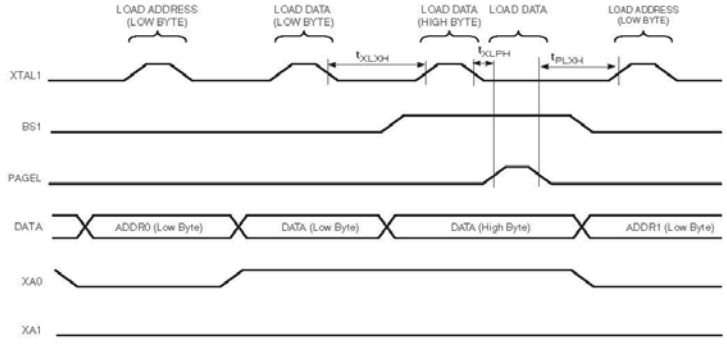
Parallel Programming Characteristics

Figure 133. Parallel Programming Timing, Including some General Timing Requirements



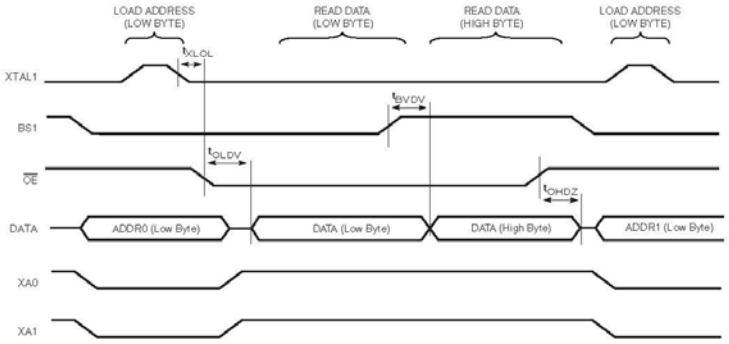
2466T–AVR–07/10

Figure 134. Parallel Programming Timing, Loading Sequence with Timing Requirements[1]



Note: 1. The timing requirements shown in Figure 133 (that is, t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to loading operation.

Figure 135. Parallel Programming Timing, Reading Sequence (within the Same Page) with Timing Requirements[1]



Note: 1. The timing requirements shown in Figure 133 (that is, t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to reading operation.

Table 113. Parallel Programming Characteristics, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250	µA

Table 113. Parallel Programming Characteristics, V_{CC} = 5V ±10% (Continued)

Symbol	Parameter	Min	Typ	Max	Units
t_{DVXH}	Data and Control Valid before XTAL1 High	67			ns
t_{XLXH}	XTAL1 Low to XTAL1 High	200			
t_{XHXL}	XTAL1 Pulse Width High	150			
t_{XLDX}	Data and Control Hold after XTAL1 Low	67			
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	0			
t_{XLPH}	XTAL1 Low to PAGEL high	0			
t_{PLXH}	PAGEL low to XTAL1 high	150			
t_{BVPH}	BS1 Valid before PAGEL High	67			
t_{PHPL}	PAGEL Pulse Width High	150			
t_{PLBX}	BS1 Hold after PAGEL Low	67			
t_{WLBX}	BS2/1 Hold after $\overline{WR}$ Low	67			
t_{PLWL}	PAGEL Low to $\overline{WR}$ Low	67			
t_{BVWL}	BS1 Valid to $\overline{WR}$ Low	67			
t_{WLWH}	$\overline{WR}$ Pulse Width Low	150			
t_{WLRL}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ Low	0		1	µs
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High[1]	3.7		4.5	ms
t_{WLRH_CE}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Chip Erase[2]	7.5		9	
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	0			ns
t_{BVDV}	BS1 Valid to DATA valid	0		250	
t_{OLDV}	$\overline{OE}$ Low to DATA Valid			250	
t_{OHDZ}	$\overline{OE}$ High to DATA Tri-stated			250	

Notes: 1. t_{WLRH} is valid for the Write Flash, Write EEPROM, Write Fuse bits and Write Lock bits commands.
2. t_{WLRH_CE} is valid for the Chip Erase command.

Serial Downloading

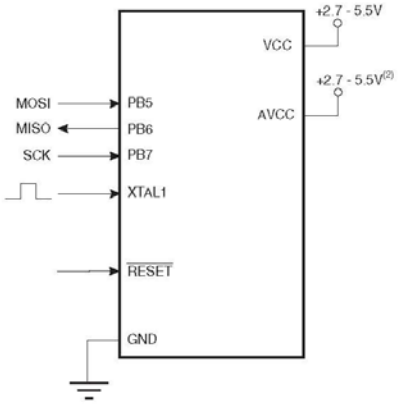
Both the Flash and EEPROM memory arrays can be programmed using the serial SPI bus while $\overline{RESET}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input), and MISO (output). After $\overline{RESET}$ is set low, the Programming Enable instruction needs to be executed first before program/erase operations can be executed. NOTE, in Table 114 on page 273, the pin mapping for SPI programming is listed. Not all parts use the SPI pins dedicated for the internal SPI interface.

SPI Serial Programming Pin Mapping

Table 114. Pin Mapping SPI Serial Programming

Symbol	Pins	I/O	Description
MOSI	PB5	I	Serial Data in
MISO	PB6	O	Serial Data out
SCK	PB7	I	Serial Clock

Figure 136. SPI Serial Programming and Verify[1]



Notes: 1. If the device is clocked by the Internal Oscillator, it is no need to connect a clock source to the XTAL1 pin.
2. V_{CC} -0.3V < AVCC < V_{CC} +0.3V, however, AVCC should always be within 2.7V - 5.5V

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Program and EEPROM arrays into $FF.

Depending on CKSEL Fuses, a valid clock must be present. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 CPU clock cycles for f_{ck} < 12 MHz, 3 CPU clock cycles for f_{ck} ≥ 12 MHz

High: > 2 CPU clock cycles for f_{ck} < 12 MHz, 3 CPU clock cycles for f_{ck} ≥ 12 MHz

SPI Serial Programming Algorithm

When writing serial data to the ATmega16, data is clocked on the rising edge of SCK.

When reading data from the ATmega16, data is clocked on the falling edge of SCK. See Figure 138 for timing details.

To program and verify the ATmega16 in the SPI Serial Programming mode, the following sequence is recommended (See four byte instruction formats in Figure 116 on page 276):

1. Power-up sequence:
   Apply power between V_{CC} and GND while $\overline{\text{RESET}}$ and SCK are set to "0". In some systems, the programmer can not guarantee that SCK is held low during power-up. In this case, $\overline{\text{RESET}}$ must be given a positive pulse of at least two CPU clock cycles duration after SCK has been set to "0".
2. Wait for at least 20 ms and enable SPI Serial Programming by sending the Programming Enable serial instruction to pin MOSI.
3. The SPI Serial Programming instructions will not work if the communication is out of synchronization. When in sync. the second byte ($53), will echo back when issuing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the $53 did not echo back, give $\overline{\text{RESET}}$ a positive pulse and issue a new Programming Enable command.

4. The Flash is programmed one page at a time. The page size is found in Table 107 on page 262. The memory page is loaded one byte at a time by supplying the 6 LSB of the address and data together with the Load Program Memory Page instruction. To ensure correct loading of the page, the data Low byte must be loaded before data High byte is applied for a given address. The Program Memory Page is stored by loading the Write Program Memory Page instruction with the 7 MSB of the address. If polling is not used, the user must wait at least t_{WD_FLASH} before issuing the next page. (See Table 115). Accessing the SPI Serial Programming interface before the Flash write operation completes can result in incorrect programming.
5. The EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. If polling is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte. (See Table 115). In a chip erased device, no $FFs in the data file(s) need to be programmed.
6. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO.
7. At the end of the programming session, $\overline{\text{RESET}}$ can be set high to commence normal operation.
8. Power-off sequence (if needed):
   Set $\overline{\text{RESET}}$ to "1".
   Turn V_{CC} power off.

Data Polling Flash

When a page is being programmed into the Flash, reading an address location within the page being programmed will give the value $FF. At the time the device is ready for a new page, the programmed value will read correctly. This is used to determine when the next page can be written. Note that the entire page is written simultaneously and any address within the page can be used for polling. Data polling of the Flash will not work for the value $FF, so when programming this value, the user will have to wait for at least t_{WD_FLASH} before programming the next page. As a chip erased device contains $FF in all locations, programming of addresses that are meant to contain $FF, can be skipped. See Table 115 for t_{WD_FLASH} value

Data Polling EEPROM

When a new byte has been written and is being programmed into EEPROM, reading the address location being programmed will give the value $FF. At the time the device is ready for a new byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the value $FF, but the user should have the following in mind: As a chip erased device contains $FF in all locations, programming of addresses that are meant to contain $FF, can be skipped. This does not apply if the EEPROM is re-programmed without chip erasing the device. In this case, data polling cannot be used for the value $FF, and the user will have to wait at least t_{WD_EEPROM} before programming the next byte. See Table 115 for t_{WD_EEPROM} value.

Table 115. Minimum Wait Delay before Writing the Next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FUSE}	4.5 ms
t_{WD_FLASH}	4.5 ms
t_{WD_EEPROM}	9.0 ms
t_{WD_ERASE}	9.0 ms

Serial Programming Instruction set

Table 116 on page 276 and Figure 137 on page 277 describes the Instruction set.

Table 116. Serial Programming Instruction Set (Hexadecimal values)

Instruction[1]/Operation	Instruction Format			
	Byte 1	Byte 2	Byte 3	Byte 4
Programming Enable	$AC	$53	$00	$00
Chip Erase (Program Memory/EEPROM)	$AC	$80	$00	$00
Poll RDY/$\overline{BSY}$	$F0	$00	$00	data byte out
Load Instructions				
Load Extended Address byte[1]	$4D	$00	Extended adr	$00
Load Program Memory Page, High byte	$48	adr MSB	adr LSB	high data byte in
Load Program Memory Page, Low byte	$40	adr MSB	adr LSB	low data byte in
Load EEPROM Memory Page (page access)[1]	$C1	$00	adr LSB	data byte in
Read Instructions				
Read Program Memory, High byte	$28	adr MSB	adr LSB	high data byte out
Read Program Memory, Low byte	$20	adr MSB	adr LSB	low data byte out
Read EEPROM Memory	$A0	adr MSB	adr LSB	data byte out
Read Lock bits	$58	$00	$00	data byte out
Read Signature Byte	$30	$00	0000 000aa	data byte out
Read Fuse bits	$50	$00	$00	data byte out
Read Fuse High bits	$58	$08	$00	data byte out
Read Extended Fuse Bits	$50	$08	$00	data byte out
Read Calibration Byte	$38	$00	$0b00 000bb	data byte out
Write Instructions				
Write Program Memory Page	$4C	000a aaaa	aa00 0000	$00
Write EEPROM Memory	$C0	adr MSB	adr LSB	data byte in
Write EEPROM Memory Page (page access)[1]	$C2	adr MSB	adr LSB	$00
Write Lock bits	$AC	$E0	$00	data byte in
Write Fuse bits	$AC	$A0	$00	data byte in
Write Fuse High bits	$AC	$A8	$00	data byte in
Write Extended Fuse Bits	$AC	$A4	$00	data byte in

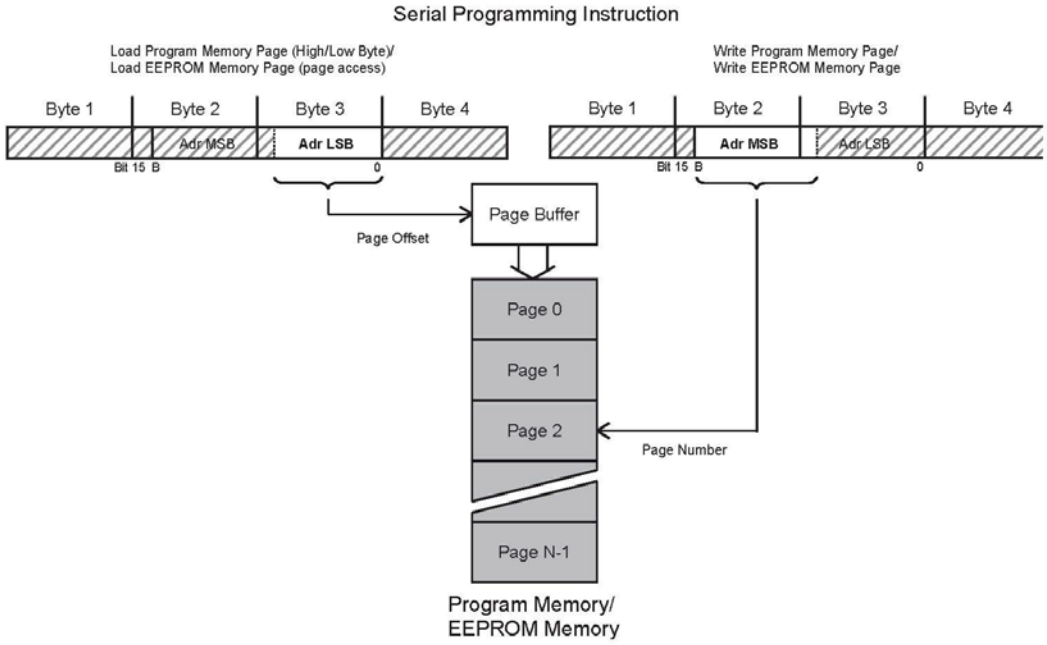
Notes:
1. Not all instructions are applicable for all parts.
2. a = address
3. Bits are programmed '0', unprogrammed '1'.
4. To ensure future compatibility, unused Fuses and Lock bits should be unprogrammed ('1') .
5. Refer to the correspondig section for Fuse and Lock bits, Calibration and Signature bytes and Page size.
6. See htt://www.atmel.com/avr for Application Notes regarding programming and programmers.

If the LSB in RDY/BSY data byte out is '1', a programming operation is still pending. Wait until this bit returns '0' before the next instruction is carried out.

Within the same page, the low data byte must be loaded prior to the high data byte.

After data is loaded to the page buffer, program the EEPROM page, see Figure 137 on page 277.

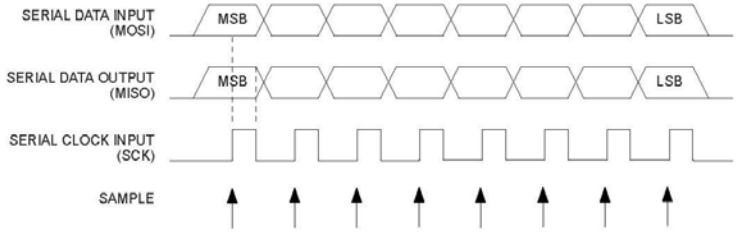
Figure 137. Serial Programming Instruction example



SPI Serial Programming Characteristics

For characteristics of the SPI module, see "SPI Timing Characteristics" on page 295.

Figure 138. SPI Serial Programming Waveforms



Programming via the JTAG Interface

Programming through the JTAG interface requires control of the four JTAG specific pins: TCK, TMS, TDI and TDO. Control of the reset and clock pins is not required.

To be able to use the JTAG interface, the JTAGEN Fuse must be programmed. The device is default shipped with the fuse programmed. In addition, the JTD bit in MCUCSR must be cleared. Alternatively, if the JTD bit is set, the External Reset can be forced low. Then, the JTD bit will be cleared after two chip clocks, and the JTAG pins are available for programming. This provides a means of using the JTAG pins as normal port pins in running mode while still allowing In-System Programming via the JTAG interface. Note that this technique can not be used when using the JTAG pins for Boundary-scan or On-chip Debug. In these cases the JTAG pins must be dedicated for this purpose.

As a definition in this datasheet, the LSB is shifted in and out first of all Shift Registers.

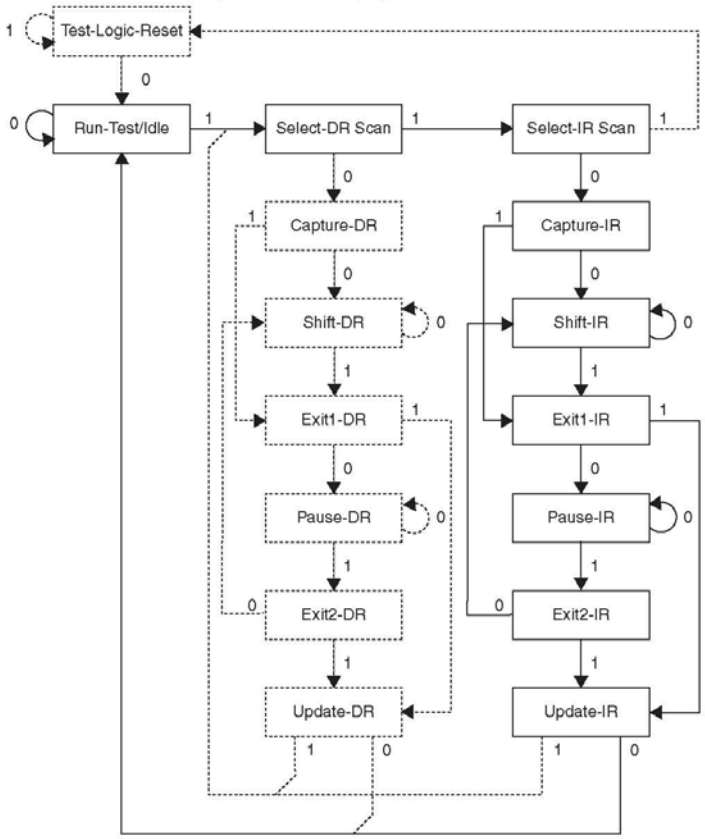
Programming Specific JTAG Instructions

The instruction register is 4-bit wide, supporting up to 16 instructions. The JTAG instructions useful for Programming are listed below.

The OPCODE for each instruction is shown behind the instruction name in hex format. The text describes which Data Register is selected as path between TDI and TDO for each instruction.

The Run-Test/Idle state of the TAP controller is used to generate internal clocks. It can also be used as an idle state between JTAG sequences. The state machine sequence for changing the instruction word is shown in Figure 139.

Figure 139. State Machine Sequence for Changing the Instruction Word



AVR_RESET ($C)

The AVR specific public JTAG instruction for setting the AVR device in the Reset mode or taking the device out from the Reset Mode. The TAP controller is not reset by this instruction. The one bit Reset Register is selected as Data Register. Note that the Reset will be active as long as there is a logic "one" in the Reset Chain. The output from this chain is not latched.

The active states are:

- Shift-DR: The Reset Register is shifted by the TCK input.

PROG_ENABLE ($4)

The AVR specific public JTAG instruction for enabling programming via the JTAG port. The 16-bit Programming Enable Register is selected as Data Register. The active states are the following:

- Shift-DR: The programming enable signature is shifted into the Data Register.
- Update-DR: The programming enable signature is compared to the correct value, and Programming mode is entered if the signature is valid.

PROG_COMMANDS ($5)

The AVR specific public JTAG instruction for entering programming commands via the JTAG port. The 15-bit Programming Command Register is selected as Data Register. The active states are the following:

- Capture-DR: The result of the previous command is loaded into the Data Register.
- Shift-DR: The Data Register is shifted by the TCK input, shifting out the result of the previous command and shifting in the new command.
- Update-DR: The programming command is applied to the Flash inputs
- Run-Test/Idle: One clock cycle is generated, executing the applied command (not always required, see Table 117 below).

PROG_PAGELOAD ($6)

The AVR specific public JTAG instruction to directly load the Flash data page via the JTAG port. The 1024 bit Virtual Flash Page Load Register is selected as Data Register. This is a virtual scan chain with length equal to the number of bits in one Flash page. Internally the Shift Register is 8-bit. Unlike most JTAG instructions, the Update-DR state is not used to transfer data from the Shift Register. The data are automatically transferred to the Flash page buffer byte by byte in the Shift-DR state by an internal state machine. This is the only active state:

- Shift-DR: Flash page data are shifted in from TDI by the TCK input, and automatically loaded into the Flash page one byte at a time.

Note: The JTAG instruction PROG_PAGELOAD can only be used if the AVR device is the first device in JTAG scan chain. If the AVR cannot be the first device in the scan chain, the byte-wise programming algorithm must be used.

PROG_PAGEREAD ($7)

The AVR specific public JTAG instruction to read one full Flash data page via the JTAG port. The 1032 bit Virtual Flash Page Read Register is selected as Data Register. This is a virtual scan chain with length equal to the number of bits in one Flash page plus 8. Internally the Shift Register is 8-bit. Unlike most JTAG instructions, the Capture-DR state is not used to transfer data to the Shift Register. The data are automatically transferred from the Flash page buffer byte by byte in the Shift-DR state by an internal state machine. This is the only active state:

- Shift-DR: Flash data are automatically read one byte at a time and shifted out on TDO by the TCK input. The TDI input is ignored.

Note: The JTAG instruction PROG_PAGEREAD can only be used if the AVR device is the first device in JTAG scan chain. If the AVR cannot be the first device in the scan chain, the byte-wise programming algorithm must be used.

Data Registers

The Data Registers are selected by the JTAG Instruction Registers described in section "Programming Specific JTAG Instructions" on page 278. The Data Registers relevant for programming operations are:

- Reset Register
- Programming Enable Register
- Programming Command Register
- Virtual Flash Page Load Register
- Virtual Flash Page Read Register

Reset Register

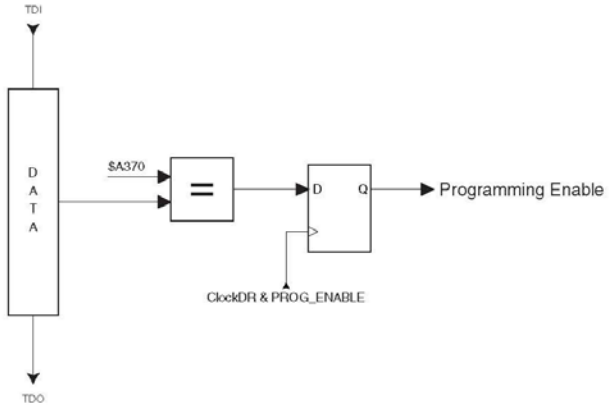
The Reset Register is a Test Data Register used to reset the part during programming. It is required to reset the part before entering programming mode.

A high value in the Reset Register corresponds to pulling the external Reset low. The part is reset as long as there is a high value present in the Reset Register. Depending on the Fuse settings for the clock options, the part will remain reset for a Reset Time-out Period (refer to "Clock Sources" on page 25) after releasing the Reset Register. The output from this Data Register is not latched, so the reset will take place immediately, as shown in Figure 115 on page 230.

Programming Enable Register

The Programming Enable Register is a 16-bit register. The contents of this register is compared to the programming enable signature, binary code 1010_0011_0111_0000. When the contents of the register is equal to the programming enable signature, programming via the JTAG port is enabled. The register is reset to 0 on Power-on Reset, and should always be reset when leaving Programming mode.

Figure 140. Programming Enable Register



Programming Command Register

The Programming Command Register is a 15-bit register. This register is used to serially shift in programming commands, and to serially shift out the result of the previous command, if any. The JTAG Programming Instruction Set is shown in Table 117. The state sequence when shifting in the programming commands is illustrated in Figure 142.

Figure 141. Programming Command Register

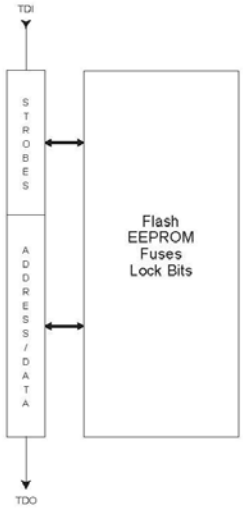


Table 117. JTAG Programming Instruction Set

a = address high bits, **b** = address low bits, **H** = 0 – Low byte, 1 – High Byte, **o** = data out, **i** = data in, x = don't care

Instruction	TDI sequence	TDO sequence	Notes
1a. Chip erase	0100011_10000000 0110001_10000000 0110011_10000000 0110011_10000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	
1b. Poll for chip erase complete	0110011_10000000	xxxxx**o**x_xxxxxxxx	(2)
2a. Enter Flash Write	0100011_00010000	xxxxxxx_xxxxxxxx	
2b. Load Address High Byte	0000111_**aaaaaaaa**	xxxxxxx_xxxxxxxx	(9)
2c. Load Address Low Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	
2d. Load Data Low Byte	0010011_**iiiiiiii**	xxxxxxx_xxxxxxxx	
2e. Load Data High Byte	0010111_**iiiiiiii**	xxxxxxx_xxxxxxxx	
2f. Latch Data	0110111_00000000 1110111_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
2g. Write Flash Page	0110111_00000000 0110101_00000000 0110111_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
2h. Poll for Page Write complete	0110111_00000000	xxxxx**o**x_xxxxxxxx	(2)
3a. Enter Flash Read	0100011_00000010	xxxxxxx_xxxxxxxx	
3b. Load Address High Byte	0000111_**aaaaaaaa**	xxxxxxx_xxxxxxxx	(9)
3c. Load Address Low Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	
3d. Read Data Low and High Byte	0110010_00000000 0110110_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo** xxxxxxx_**oooooooo**	 Low byte High byte
4a. Enter EEPROM Write	0100011_00010001	xxxxxxx_xxxxxxxx	
4b. Load Address High Byte	0000111_**aaaaaaaa**	xxxxxxx_xxxxxxxx	(9)
4c. Load Address Low Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	
4d. Load Data Byte	0010011_**iiiiiiii**	xxxxxxx_xxxxxxxx	
4e. Latch Data	0110111_00000000 1110111_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
4f. Write EEPROM Page	0110011_00000000 0110001_00000000 0110011_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
4g. Poll for Page Write complete	0110011_00000000	xxxxx**o**x_xxxxxxxx	(2)
5a. Enter EEPROM Read	0100011_00000011	xxxxxxx_xxxxxxxx	
5b. Load Address High Byte	0000111_**aaaaaaaa**	xxxxxxx_xxxxxxxx	(9)
5c. Load Address Low Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	

Table 117. JTAG Programming Instruction Set (Continued)
a = address high bits, **b** = address low bits, **H** = 0 – Low byte, 1 – High Byte, **o** = data out, **i** = data in, x = don't care

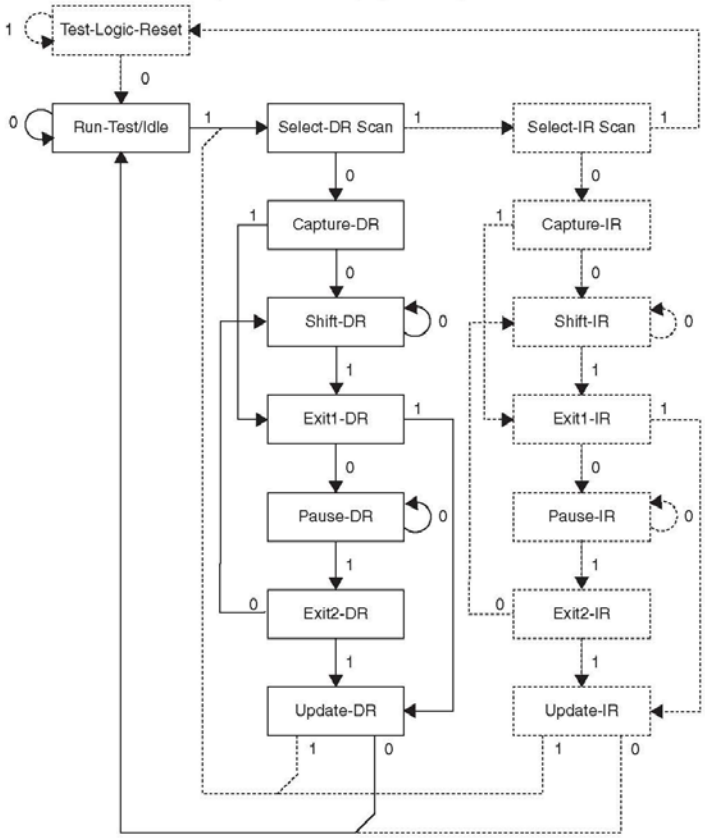
Instruction	TDI sequence	TDO sequence	Notes
5d. Read Data Byte	0110011_**bbbbbbbb** 0110010_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo**	
6a. Enter Fuse Write	0100011_01000000	xxxxxxx_xxxxxxxx	
6b. Load Data Low Byte[6]	0010011_**iiiiiiii**	xxxxxxx_xxxxxxxx	(3)
6c. Write Fuse High byte	0110111_00000000 0110101_00000000 0110111_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
6d. Poll for Fuse Write complete	0110111_00000000	xxxxx**o**x_xxxxxxxx	(2)
6e. Load Data Low Byte[7]	0010011_**iiiiiiii**	xxxxxxx_xxxxxxxx	(3)
6f. Write Fuse Low byte	0110011_00000000 0110001_00000000 0110011_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
6g. Poll for Fuse Write complete	0110011_00000000	xxxxx**o**x_xxxxxxxx	(2)
7a. Enter Lock Bit Write	0100011_00100000	xxxxxxx_xxxxxxxx	
7b. Load Data Byte[8]	0010011_11**iiiiii**	xxxxxxx_xxxxxxxx	(4)
7c. Write Lock Bits	0110011_00000000 0110001_00000000 0110011_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
7d. Poll for Lock Bit Write complete	0110011_00000000	xxxxx**o**x_xxxxxxxx	(2)
8a. Enter Fuse/Lock Bit Read	0100011_00000100	xxxxxxx_xxxxxxxx	
8b. Read Fuse High Byte[6]	0111110_00000000 0111111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo**	
8c. Read Fuse Low Byte[7]	0110010_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo**	
8d. Read Lock Bits[8]	0110110_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xx**oooooo**	(5)
8e. Read Fuses and Lock Bits	0111110_00000000 0110010_00000000 0110110_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo** xxxxxxx_**oooooooo** xxxxxxx_**oooooooo**	(5) Fuse High Byte Fuse Low Byte Lock bits
9a. Enter Signature Byte Read	0100011_00001000	xxxxxxx_xxxxxxxx	
9b. Load Address Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	
9c. Read Signature Byte	0110010_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo**	
10a. Enter Calibration Byte Read	0100011_00001000	xxxxxxx_xxxxxxxx	

Table 117. JTAG Programming Instruction Set (Continued)
a = address high bits, **b** = address low bits, **H** = 0 – Low byte, 1 – High Byte, **o** = data out, **i** = data in, x = don't care

Instruction	TDI sequence	TDO sequence	Notes
10b. Load Address Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	
10c. Read Calibration Byte	0110110_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo**	
11a. Load No Operation Command	0100011_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	

Notes:
1. This command sequence is not required if the seven MSB are correctly set by the previous command sequence (which is normally the case).
2. Repeat until **o** = "1".
3. Set bits to "0" to program the corresponding fuse, "1" to unprogram the fuse.
4. Set bits to "0" to program the corresponding lock bit, "1" to leave the lock bit unchanged.
5. "0" = programmed, "1" = unprogrammed.
6. The bit mapping for Fuses High byte is listed in Table 105 on page 260
7. The bit mapping for Fuses Low byte is listed in Table 106 on page 261
8. The bit mapping for Lock bits byte is listed in Table 103 on page 259
9. Address bits exceeding PCMSB and EEAMSB (Table 107 and Table 108) are don't care

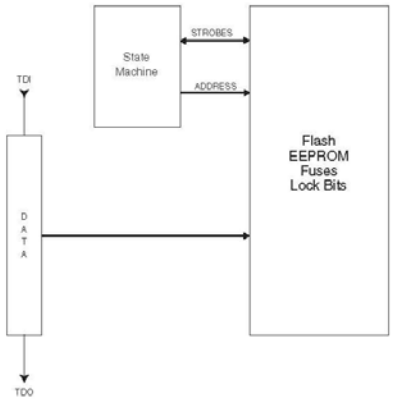
Figure 142. State Machine Sequence for Changing/Reading the Data Word



Virtual Flash Page Load Register

The Virtual Flash Page Load Register is a virtual scan chain with length equal to the number of bits in one Flash page. Internally the Shift Register is 8-bit, and the data are automatically transferred to the Flash page buffer byte by byte. Shift in all instruction words in the page, starting with the LSB of the first instruction in the page and ending with the MSB of the last instruction in the page. This provides an efficient way to load the entire Flash page buffer before executing Page Write.

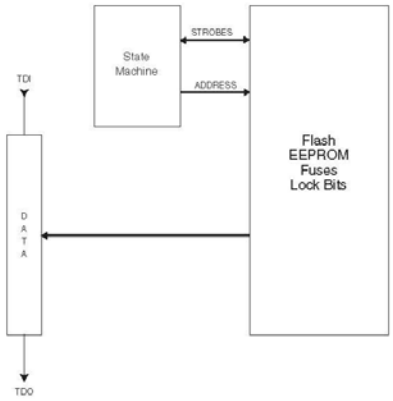
Figure 143. Virtual Flash Page Load Register



Virtual Flash Page Read Register

The Virtual Flash Page Read Register is a virtual scan chain with length equal to the number of bits in one Flash page plus 8. Internally the Shift Register is 8-bit, and the data are automatically transferred from the Flash data page byte by byte. The first 8 cycles are used to transfer the first byte to the internal Shift Register, and the bits that are shifted out during these 8 cycles should be ignored. Following this initialization, data are shifted out starting with the LSB of the first instruction in the page and ending with the MSB of the last instruction in the page. This provides an efficient way to read one full Flash page to verify programming.

Figure 144. Virtual Flash Page Read Register



Programming Algorithm

All references below of type "1a", "1b", and so on, refer to Table 117.

Entering Programming Mode

1. Enter JTAG instruction AVR_RESET and shift 1 in the Reset Register.
2. Enter instruction PROG_ENABLE and shift 1010_0011_0111_0000 in the Programming Enable Register.

Leaving Programming Mode

1. Enter JTAG instruction PROG_COMMANDS.
2. Disable all programming instructions by using no operation instruction 11a.
3. Enter instruction PROG_ENABLE and shift 0000_0000_0000_0000 in the programming Enable Register.
4. Enter JTAG instruction AVR_RESET and shift 0 in the Reset Register.

Performing Chip Erase

1. Enter JTAG instruction PROG_COMMANDS.
2. Start chip erase using programming instruction 1a.
3. Poll for Chip Erase complete using programming instruction 1b, or wait for t_{WLRH_CE} (refer to Table 113 on page 272).

Programming the Flash

Before programming the Flash a Chip Erase must be performed. See "Performing Chip Erase" on page 288.

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Flash write using programming instruction 2a.
3. Load address High byte using programming instruction 2b.
4. Load address Low byte using programming instruction 2c.
5. Load data using programming instructions 2d, 2e and 2f.
6. Repeat steps 4 and 5 for all instruction words in the page.
7. Write the page using programming instruction 2g.
8. Poll for Flash write complete using programming instruction 2h, or wait for t_{WLRH} (refer to Table 113 on page 272).
9. Repeat steps 3 to 7 until all data have been programmed.

A more efficient data transfer can be achieved using the PROG_PAGELOAD instruction:

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Flash write using programming instruction 2a.
3. Load the page address using programming instructions 2b and 2c. PCWORD (refer to Table 107 on page 262) is used to address within one page and must be written as 0.
4. Enter JTAG instruction PROG_PAGELOAD.
5. Load the entire page by shifting in all instruction words in the page, starting with the LSB of the first instruction in the page and ending with the MSB of the last instruction in the page.
6. Enter JTAG instruction PROG_COMMANDS.
7. Write the page using programming instruction 2g.
8. Poll for Flash write complete using programming instruction 2h, or wait for t_{WLRH} (refer to Table 113 on page 272).
9. Repeat steps 3 to 8 until all data have been programmed.

Reading the Flash

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Flash read using programming instruction 3a.
3. Load address using programming instructions 3b and 3c.
4. Read data using programming instruction 3d.
5. Repeat steps 3 and 4 until all data have been read.

A more efficient data transfer can be achieved using the PROG_PAGEREAD instruction:

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Flash read using programming instruction 3a.
3. Load the page address using programming instructions 3b and 3c. PCWORD (refer to Table 107 on page 262) is used to address within one page and must be written as 0.
4. Enter JTAG instruction PROG_PAGEREAD.
5. Read the entire page by shifting out all instruction words in the page, starting with the LSB of the first instruction in the page and ending with the MSB of the last instruction in the page. Remember that the first 8 bits shifted out should be ignored.
6. Enter JTAG instruction PROG_COMMANDS.
7. Repeat steps 3 to 6 until all data have been read.

Programming the EEPROM

Before programming the EEPROM a Chip Erase must be performed. See "Performing Chip Erase" on page 288.

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable EEPROM write using programming instruction 4a.
3. Load address High byte using programming instruction 4b.
4. Load address Low byte using programming instruction 4c.
5. Load data using programming instructions 4d and 4e.
6. Repeat steps 4 and 5 for all data bytes in the page.
7. Write the data using programming instruction 4f.
8. Poll for EEPROM write complete using programming instruction 4g, or wait for t_{WLRH} (refer to Table 113 on page 272).
9. Repeat steps 3 to 8 until all data have been programmed.

Note that the PROG_PAGELOAD instruction can not be used when programming the EEPROM

Reading the EEPROM

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable EEPROM read using programming instruction 5a.
3. Load address using programming instructions 5b and 5c.
4. Read data using programming instruction 5d.
5. Repeat steps 3 and 4 until all data have been read.

Note that the PROG_PAGEREAD instruction can not be used when reading the EEPROM

Programming the Fuses

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Fuse write using programming instruction 6a.
3. Load data High byte using programming instructions 6b. A bit value of "0" will program the corresponding fuse, a "1" will unprogram the fuse.
4. Write Fuse High byte using programming instruction 6c.
5. Poll for Fuse write complete using programming instruction 6d, or wait for t_{WLRH} (refer to Table 113 on page 272).
6. Load data Low byte using programming instructions 6e. A "0" will program the fuse, a "1" will unprogram the fuse.
7. Write Fuse Low byte using programming instruction 6f.
8. Poll for Fuse write complete using programming instruction 6g, or wait for t_{WLRH} (refer to Table 113 on page 272).

Programming the Lock Bits

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Lock bit write using programming instruction 7a.
3. Load data using programming instructions 7b. A bit value of "0" will program the corresponding Lock bit, a "1" will leave the Lock bit unchanged.
4. Write Lock bits using programming instruction 7c.
5. Poll for Lock bit write complete using programming instruction 7d, or wait for t_{WLRH} (refer to Table 113 on page 272).

Reading the Fuses and Lock Bits

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Fuse/Lock bit read using programming instruction 8a.
3. To read all Fuses and Lock bits, use programming instruction 8e.
   To only read Fuse High byte, use programming instruction 8b.
   To only read Fuse Low byte, use programming instruction 8c.
   To only read Lock bits, use programming instruction 8d.

Reading the Signature Bytes

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Signature byte read using programming instruction 9a.
3. Load address $00 using programming instruction 9b.
4. Read first signature byte using programming instruction 9c.
5. Repeat steps 3 and 4 with address $01 and address $02 to read the second and third signature bytes, respectively.

Reading the Calibration Byte

1. Enter JTAG instruction PROG_COMMANDS.
2. Enable Calibration byte read using programming instruction 10a.
3. Load address $00 using programming instruction 10b.
4. Read the calibration byte using programming instruction 10c.

Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA PDIP and 400.0 mA TQFP/MLF

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.7V to 5.5V (Unless Otherwise Noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage except XTAL1 and $\overline{\text{RESET}}$ pins	V_{CC} = 2.7 - 5.5	-0.5		0.2 V_{CC}[1]	V
V_{IH}	Input High Voltage except XTAL1 and $\overline{\text{RESET}}$ pins	V_{CC} = 2.7 - 5.5	0.6 V_{CC}[2]		V_{CC} +0.5	
V_{IH1}	Input High Voltage XTAL1 pin	V_{CC} = 2.7 - 5.5	0.7 V_{CC}[2]		V_{CC} +0.5	
V_{IL1}	Input Low Voltage XTAL1 pin	V_{CC} = 2.7 - 5.5	-0.5		0.1 V_{CC}[1]	
V_{IH2}	Input High Voltage $\overline{\text{RESET}}$ pin	V_{CC} = 2.7 - 5.5	0.9 V_{CC}[2]		V_{CC} +0.5	
V_{IL2}	Input Low Voltage $\overline{\text{RESET}}$ pin	V_{CC} = 2.7 - 5.5	-0.5		0.2 V_{CC}	
V_{OL}	Output Low Voltage[3] (Ports A,B,C,D)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.7 0.5	V V
V_{OH}	Output High Voltage[4] (Ports A,B,C,D)	I_{OH} = -20 mA, V_{CC} = 5V I_{OH} = -10 mA, V_{CC} = 3V	4.2 2.2			V V
I_{IL}	Input Leakage Current I/O Pin	Vcc = 5.5V, pin low (absolute value)			1	µA
I_{IH}	Input Leakage Current I/O Pin	Vcc = 5.5V, pin high (absolute value)			1	
R_{RST}	Reset Pull-up Resistor		30		60	kΩ
R_{pu}	I/O Pin Pull-up Resistor		20		50	

T_A = -40°C to 85°C, V_{CC} = 2.7V to 5.5V (Unless Otherwise Noted) (Continued)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{CC}	Power Supply Current	Active 1 MHz, V_{CC} = 3V (ATmega16L)		1.1		mA
		Active 4 MHz, V_{CC} = 3V (ATmega16L)		3.8	5	
		Active 8 MHz, V_{CC} = 5V (ATmega16)		12	15	
		Idle 1 MHz, V_{CC} = 3V (ATmega16L)		0.35		
		Idle 4 MHz, V_{CC} = 3V (ATmega16L)		1.2	2	
		Idle 8 MHz, V_{CC} = 5V (ATmega16)		5.5	7	
	Power-down Mode(5)	WDT enabled, V_{CC} = 3V		<8	15	µA
		WDT disabled, V_{CC} = 3V		< 1	4	
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V V_{in} = V_{CC}/2			40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = V_{CC}/2	-50		50	nA
t_{ACPD}	Analog Comparator Propagation Delay	V_{CC} = 2.7V V_{CC} = 4.0V		750 500		ns

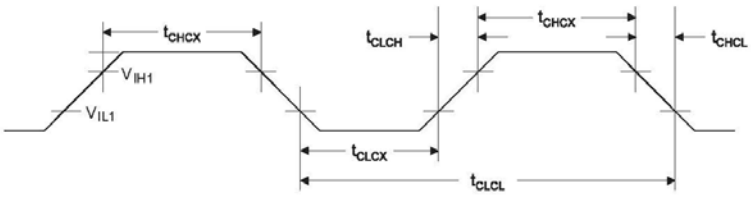
Notes:
1. "Max" means the highest value where the pin is guaranteed to be read as low
2. "Min" means the lowest value where the pin is guaranteed to be read as high
3. Although each I/O port can sink more than the test conditions (20 mA at Vcc = 5V, 10 mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   PDIP Package:
   1] The sum of all IOL, for all ports, should not exceed 200 mA.
   2] The sum of all IOL, for port A0 - A7, should not exceed 100 mA.
   3] The sum of all IOL, for ports B0 - B7,C0 - C7, D0 - D7 and XTAL2, should not exceed 100 mA.
   TQFP and QFN/MLF Package:
   1] The sum of all IOL, for all ports, should not exceed 400 mA.
   2] The sum of all IOL, for ports A0 - A7, should not exceed 100 mA.
   3] The sum of all IOL, for ports B0 - B4, should not exceed 100 mA.
   4] The sum of all IOL, for ports B3 - B7, XTAL2, D0 - D2, should not exceed 100 mA.
   5] The sum of all IOL, for ports D3 - D7, should not exceed 100 mA.
   6] The sum of all IOL, for ports C0 - C7, should not exceed 100 mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
4. Although each I/O port can source more than the test conditions (20 mA at Vcc = 5V, 10 mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   PDIP Package:
   1] The sum of all IOH, for all ports, should not exceed 200 mA.
   2] The sum of all IOH, for port A0 - A7, should not exceed 100 mA.
   3] The sum of all IOH, for ports B0 - B7,C0 - C7, D0 - D7 and XTAL2, should not exceed 100 mA.
   TQFP and QFN/MLF Package:
   1] The sum of all IOH, for all ports, should not exceed 400 mA.
   2] The sum of all IOH, for ports A0 - A7, should not exceed 100 mA.
   3] The sum of all IOH, for ports B0 - B4, should not exceed 100 mA.
   4] The sum of all IOH, for ports B3 - B7, XTAL2, D0 - D2, should not exceed 100 mA.

5] The sum of all IOH, for ports D3 - D7, should not exceed 100 mA.
6] The sum of all IOH, for ports C0 - C7, should not exceed 100 mA.If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

5. Minimum V_{CC} for Power-down is 2.5V.

External Clock Drive Waveforms

Figure 145. External Clock Drive Waveforms



External Clock Drive

Table 118. External Clock Drive[1]

Symbol	Parameter	V_{CC} = 2.7V to 5.5V		V_{CC} = 4.5V to 5.5V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	8	0	16	MHz
t_{CLCL}	Clock Period	125		62.5		ns
t_{CHCX}	High Time	50		25		ns
t_{CLCX}	Low Time	50		25		ns
t_{CLCH}	Rise Time		1.6		0.5	µs
t_{CHCL}	Fall Time		1.6		0.5	µs
Δt_{CLCL}	Change in period from one clock cycle to the next		2		2	%

Note: 1. Refer to "External Clock" on page 31 for details.

Table 119. External RC Oscillator, Typical Frequencies (V_{CC} = 5)

R [kΩ][1]	C [pF]	f[2]
33	22	650 kHz
10	22	2.0 MHz

Notes: 1. R should be in the range 3 kΩ - 100 kΩ, and C should be at least 20 pF.
2. The frequency will vary with package type and board layout.

Two-wire Serial Interface Characteristics

Table 120 describes the requirements for devices connected to the Two-wire Serial Bus. The ATmega16 Two-wire Serial Interface meets or exceeds these requirements under the noted conditions.

Timing symbols refer to Figure 146.

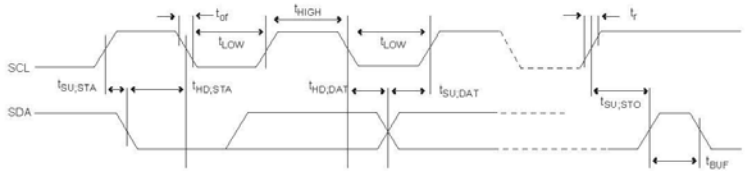
Table 120. Two-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.3 V_{CC}	V
V_{IH}	Input High-voltage		0.7 V_{CC}	V_{CC} + 0.5	
V_{hys} (1)	Hysteresis of Schmitt Trigger Inputs		0.05 V_{CC} (2)	–	
V_{OL} (1)	Output Low-voltage	3 mA sink current	0	0.4	
t_r (1)	Rise Time for both SDA and SCL		20 + 0.1C_b (3)(2)	300	ns
t_{of} (1)	Output Fall Time from V_{IHmin} to V_{ILmax}	10 pF < C_b < 400 pF (3)	20 + 0.1C_b (3)(2)	250	
t_{SP} (1)	Spikes Suppressed by Input Filter		0	50 (2)	
I_i	Input Current each I/O Pin	0.1V_{CC} < V_i < 0.9V_{CC}	-10	10	µA
C_i (1)	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	f_{CK} (4) > max(16f_{SCL}, 250kHz) (5)	0	400	kHz
Rp	Value of Pull-up resistor	f_{SCL} ≤ 100 kHz	$\frac{V_{CC}-0.4V}{3\ mA}$	$\frac{1000ns}{C_b}$	Ω
		f_{SCL} > 100 kHz	$\frac{V_{CC}-0.4V}{3\ mA}$	$\frac{300ns}{C_b}$	
$t_{HD;STA}$	Hold Time (repeated) START Condition	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	
t_{LOW}	Low Period of the SCL Clock	f_{SCL} ≤ 100 kHz	4.7	–	
		f_{SCL} > 100 kHz	1.3	–	
t_{HIGH}	High period of the SCL clock	f_{SCL} ≤ 100 kHz	4.0	–	
		f_{SCL} > 100 kHz	0.6	–	
$t_{SU;STA}$	Set-up time for a repeated START condition	f_{SCL} ≤ 100 kHz	4.7	–	
		f_{SCL} > 100 kHz	0.6	–	
$t_{HD;DAT}$	Data hold time	f_{SCL} ≤ 100 kHz	0	3.45	
		f_{SCL} > 100 kHz	0	0.9	
$t_{SU;DAT}$	Data setup time	f_{SCL} ≤ 100 kHz	250	–	ns
		f_{SCL} > 100 kHz	100	–	
$t_{SU;STO}$	Setup time for STOP condition	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	
t_{BUF}	Bus free time between a STOP and START condition	f_{SCL} ≤ 100 kHz	4.7	–	
		f_{SCL} > 100 kHz	1.3	–	

Notes:
1. In ATmega16, this parameter is characterized and not 100% tested.
2. Required only for f_{SCL} > 100 kHz.
3. C_b = capacitance of one bus line in pF.
4. f_{CK} = CPU clock frequency

5. This requirement applies to all ATmega16 Two-wire Serial Interface operation. Other devices connected to the Two-wire Serial Bus need only obey the general f_{SCL} requirement.

Figure 146. Two-wire Serial Bus Timing



SPI Timing Characteristics

See Figure 147 and Figure 148 for details.

Table 121. SPI Timing Parameters

	Description	Mode	Min	Typ	Max	
1	SCK period	Master		See Table 58		
2	SCK high/low	Master		50% duty cycle		
3	Rise/Fall time	Master		3.6		
4	Setup	Master		10		
5	Hold	Master		10		
6	Out to SCK	Master		0.5 • t_{SCK}		ns
7	SCK to out	Master		10		
8	SCK to out high	Master		10		
9	$\overline{SS}$ low to out	Slave		15		
10	SCK period	Slave	4 • t_{SCK}			
11	SCK high/low	Slave	2 • t_{SCK}			
12	Rise/Fall time	Slave			1.6	µs
13	Setup	Slave	10			
14	Hold	Slave	10			
15	SCK to out	Slave		15		
16	SCK to $\overline{SS}$ high	Slave	20			ns
17	$\overline{SS}$ high to tri-state	Slave		10		
18	$\overline{SS}$ low to SCK	Slave	2 • t_{SCK}			

Figure 147. SPI Interface Timing Requirements (Master Mode)

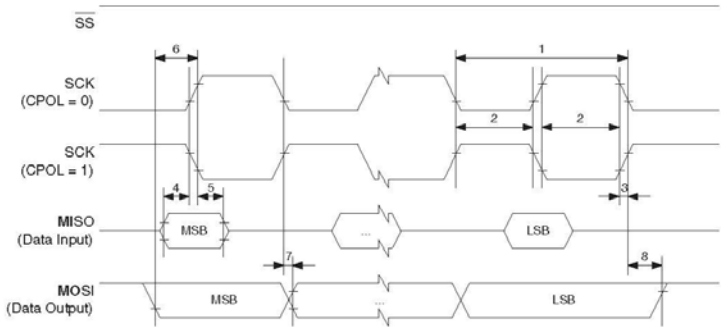
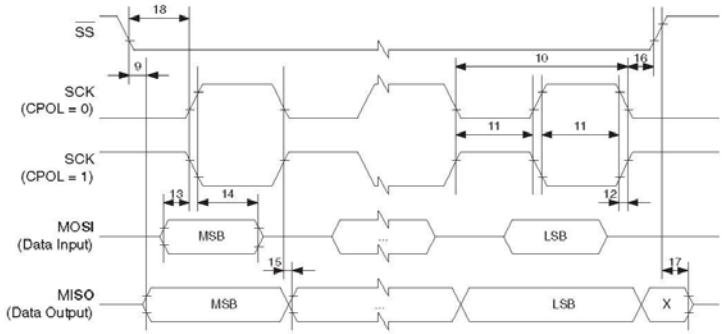


Figure 148. SPI Interface Timing Requirements (Slave Mode)



ADC Characteristics

Table 122. ADC Characteristics

Symbol	Parameter	Condition	Min[1]	Typ[1]	Max[1]	Units
	Resolution	Single Ended Conversion		10		Bits
		Differential Conversion Gain = 1x or 10x		8		
		Differential Conversion Gain = 200x		7		
	Absolute Accuracy (Including INL, DNL, Quantization Error, Gain, and Offset Error).	Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz		1.5	2.5	LSB
		Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 1 MHz		3	4	
		Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz Noise Reduction mode		1.5		
		Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 1 MHz Noise Reduction mode		3		
	Integral Non-linearity (INL)	Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz		1		
	Differential Non-linearity (DNL)	Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz		0.5		
	Gain Error	Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz		1		
	Offset Error	Single Ended Conversion V_{REF} = 4V, V_{CC} = 4V ADC clock = 200 kHz				
	Conversion Time	Free Running Conversion	13		260	µs
	Clock Frequency		50		1000	kHz
AVCC	Analog Supply Voltage		V_{CC} - 0.3[2]		V_{CC} + 0.3[3]	V
V_{REF}	Reference Voltage	Single Ended Conversion	2.0		AVCC	
		Differential Conversion	2.0		AVCC - 0.2	
V_{IN}	Input voltage	Single ended channels	GND		V_{REF}	
		Differential channels	0		V_{REF}	
	Input bandwidth	Single ended channels		38.5		kHz
		Differential channels		4		

Table 122. ADC Characteristics (Continued)

Symbol	Parameter	Condition	Min[1]	Typ[1]	Max[1]	Units
V_{INT}	Internal Voltage Reference		2.3	2.6	2.9	V
R_{REF}	Reference Input Resistance			32		kΩ
R_{AIN}	Analog Input Resistance			100		MΩ

Notes: 1. Values are guidelines only.
2. Minimum for AVCC is 2.7V.
3. Maximum for AVCC is 5.5V.

ATmega16 Typical Characteristics

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A sine wave generator with rail-to-rail output is used as clock source.

The power consumption in Power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as C_L*V_{CC}*f where C_L = load capacitance, V_{CC} = operating voltage and f = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.

Active Supply Current

Figure 149. Active Supply Current vs. Frequency (0.1 MHz - 1.0 MHz

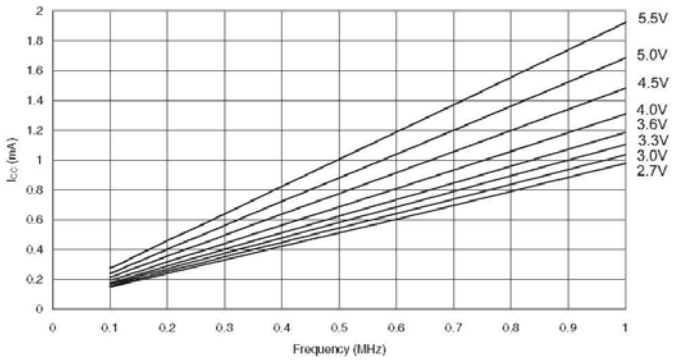


Figure 150. Active Supply Current vs. Frequency (1 MHz - 20 MHz)

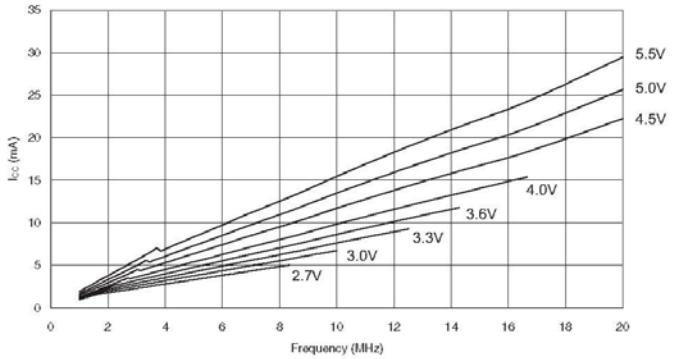


Figure 151. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)

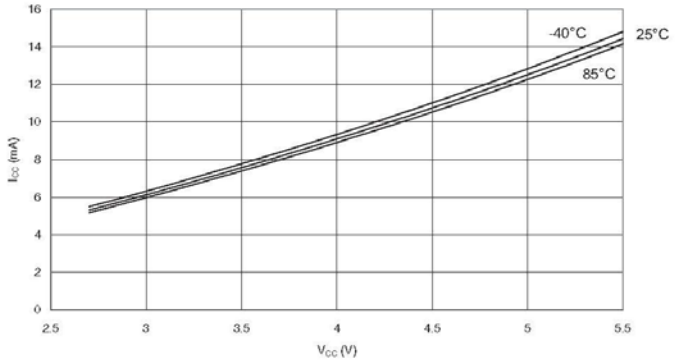


Figure 152. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 4 MHz)

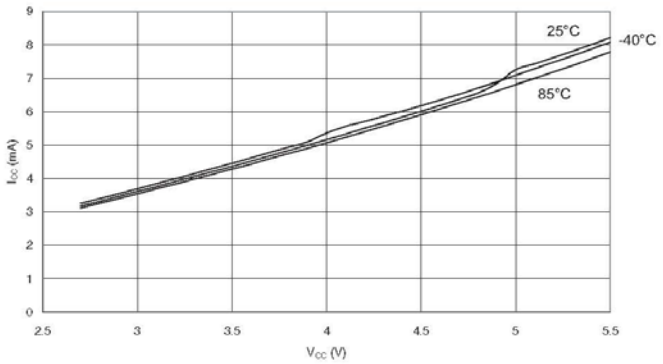


Figure 153. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 2 MHz)

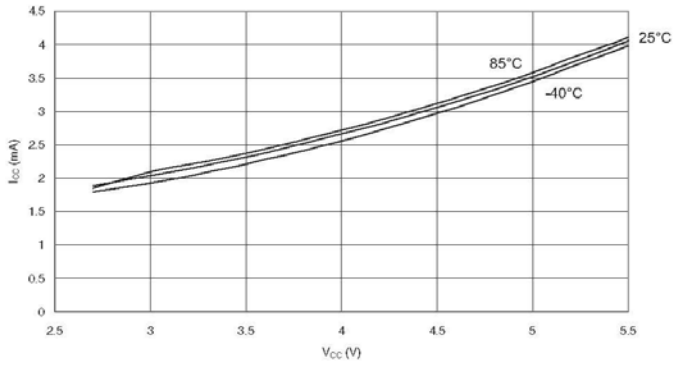


Figure 154. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

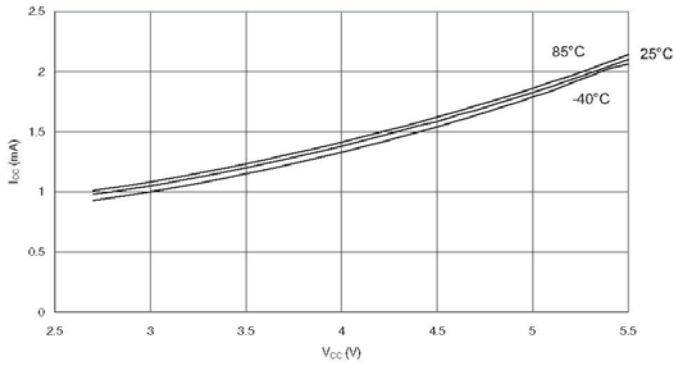
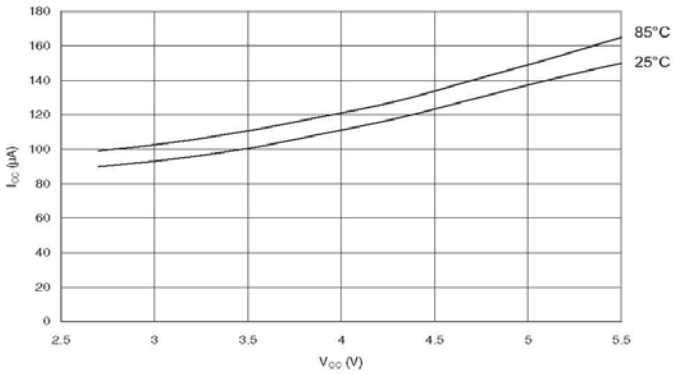


Figure 155. Active Supply Current vs. V_{CC} (32 kHz External Oscillator)



Idle Supply Current

Figure 156. Idle Supply Current vs. Frequency (0.1 MHz - 1.0 MHz)

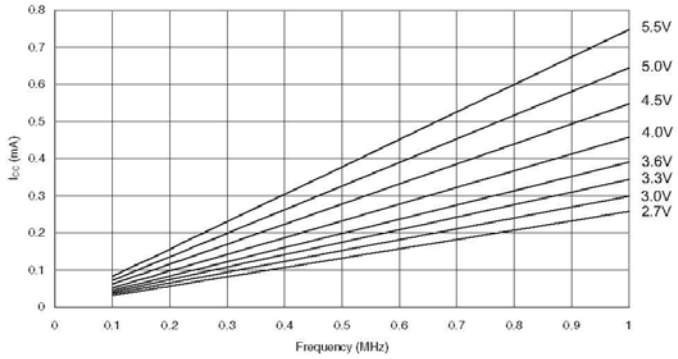


Figure 157. Idle Supply Current vs. Frequency (1 MHz - 20 MHz)

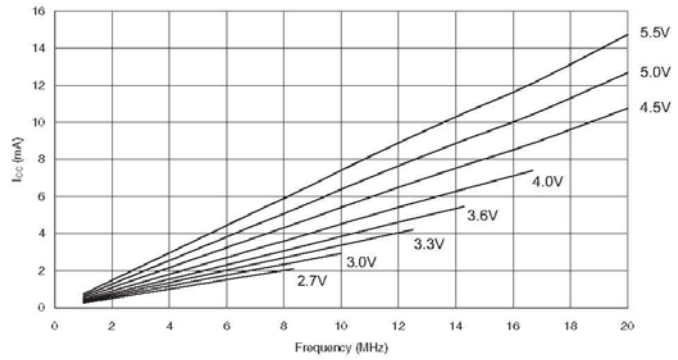


Figure 158. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)

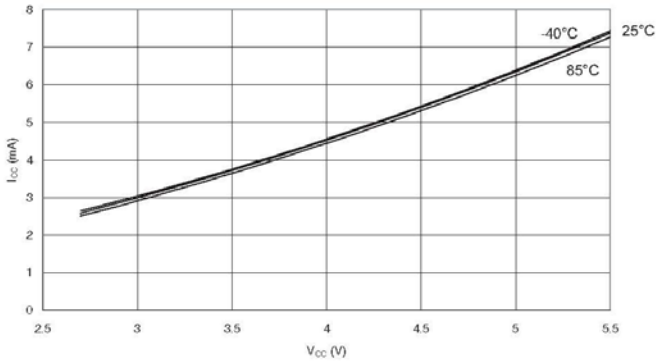


Figure 159. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 4 MHz)

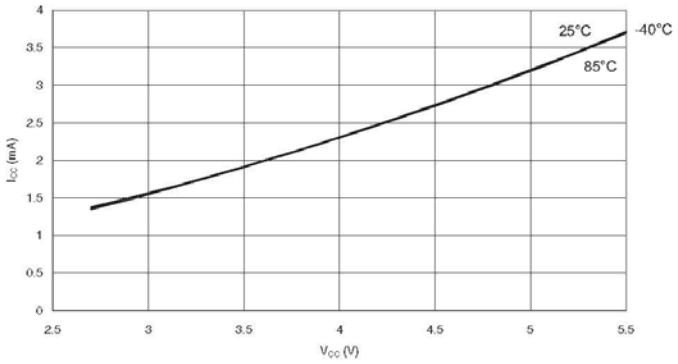


Figure 160. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 2 MHz)

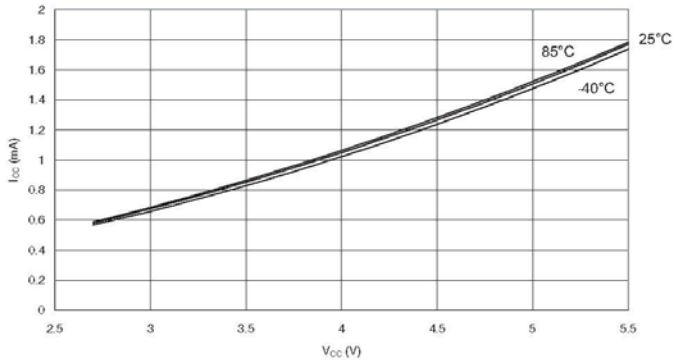


Figure 161. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

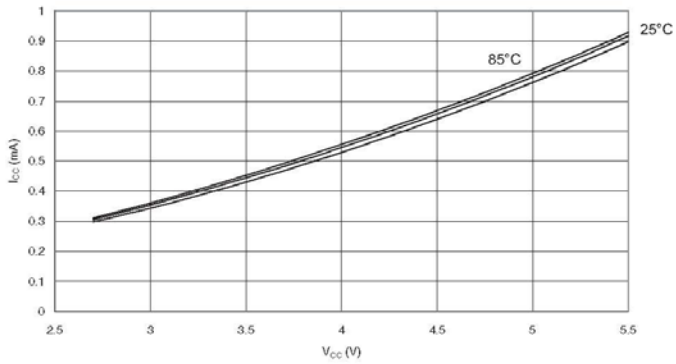
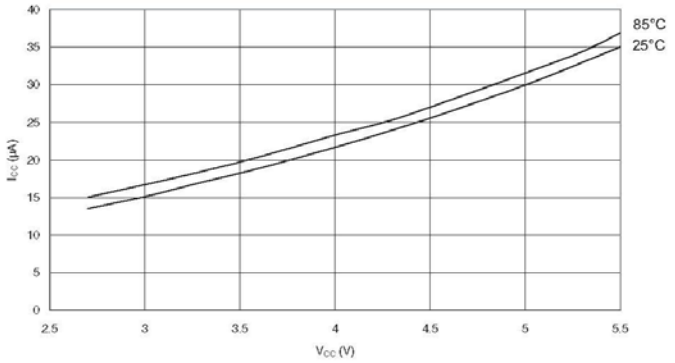


Figure 162. Idle Supply Current vs. V_{CC} (32 kHz External Oscillator)



Power-Down Supply Current

Figure 163. Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

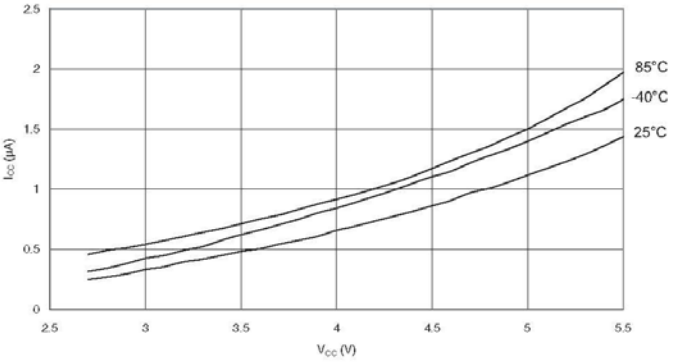
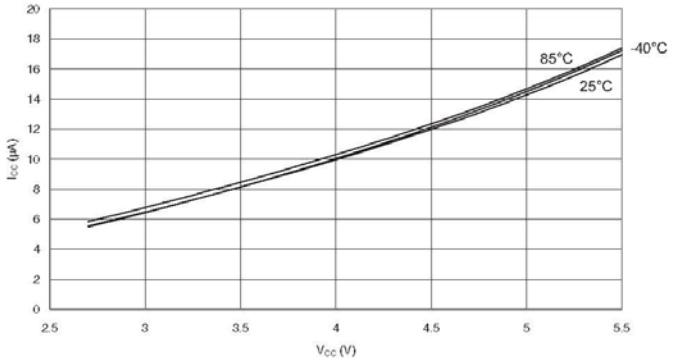
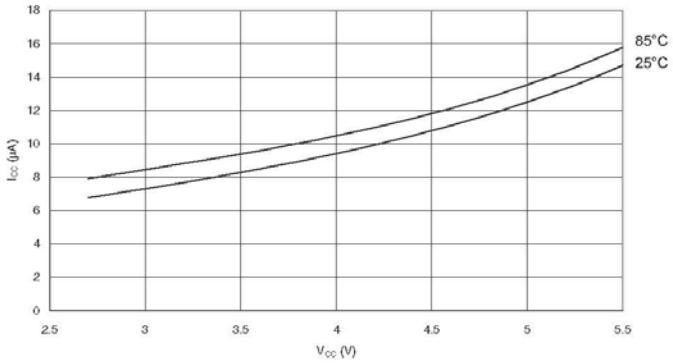


Figure 164. Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



Power-Save Supply Current

Figure 165. Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled)



Standby Supply Current

Figure 166. Standby Supply Current vs. V_{CC} (455 kHz Resonator, Watchdog Timer Disabled)

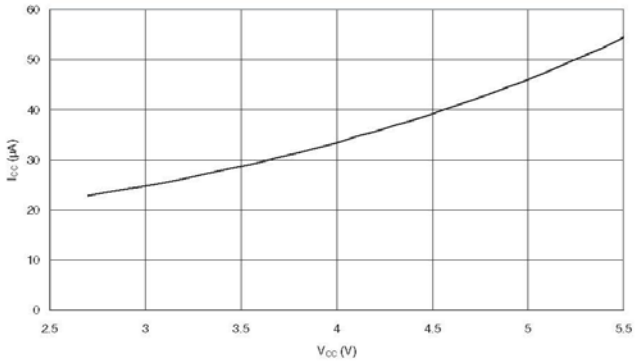


Figure 167. Standby Supply Current vs. V_{CC} (1 MHz Resonator, Watchdog Timer Disabled)

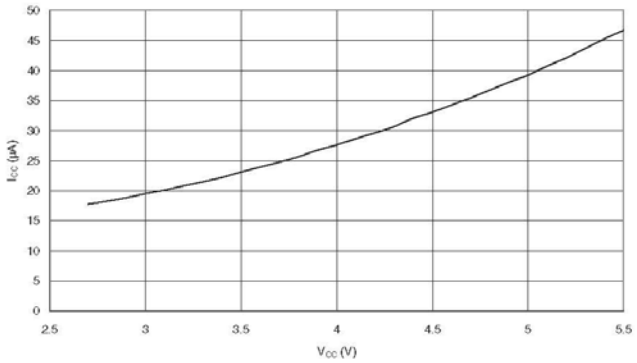


Figure 168. Standby Supply Current vs. V_{CC} (2 MHz Resonator, Watchdog Timer Disabled)

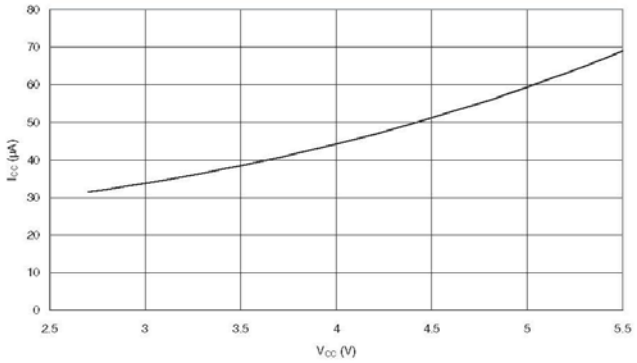


Figure 169. Standby Supply Current vs. V_{CC} (2 MHz Xtal, Watchdog Timer Disabled)

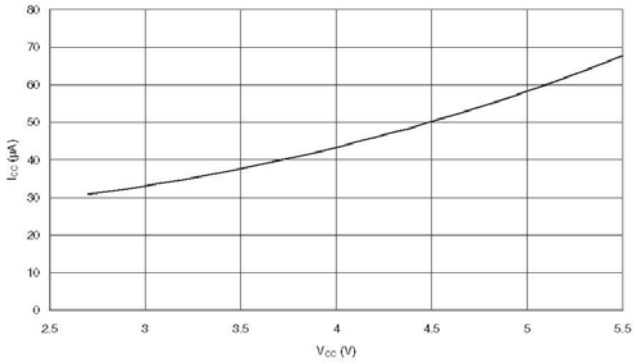


Figure 170. Standby Supply Current vs. V_{CC} (4 MHz Resonator, Watchdog Timer Disabled)

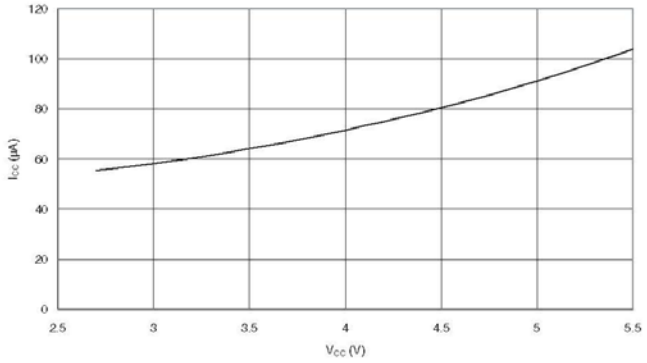


Figure 171. Standby Supply Current vs. V_{CC} (4 MHz Xtal, Watchdog Timer Disabled)

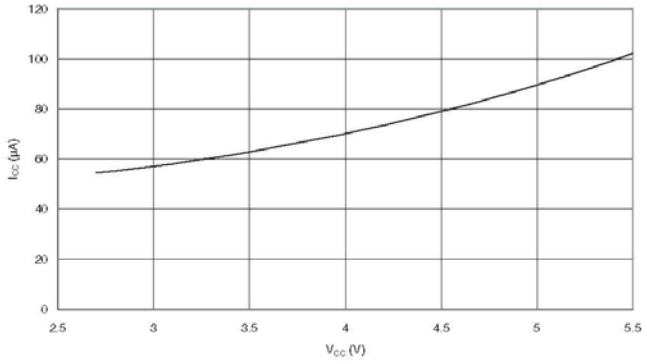


Figure 172. Standby Supply Current vs. V_{CC} (6 MHz Resonator, Watchdog Timer Disabled)

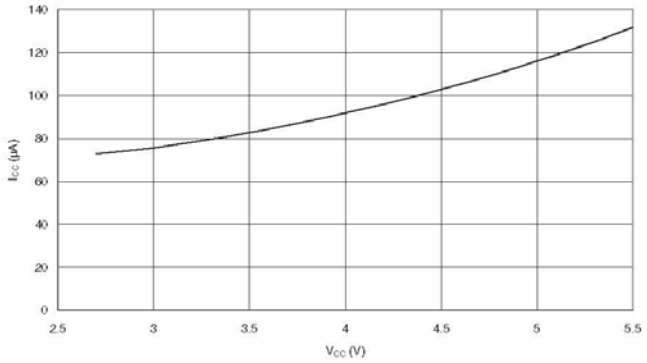
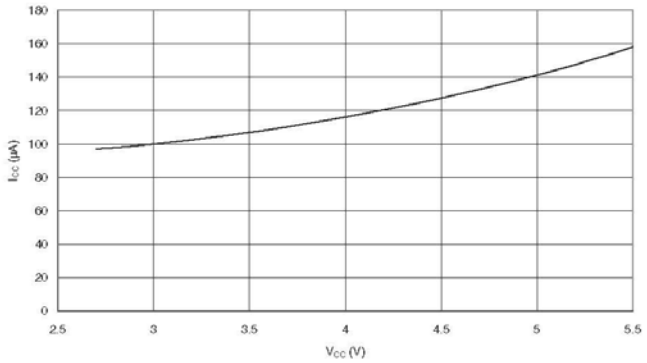


Figure 173. Standby Supply Current vs. V_{CC} (6 MHz Xtal, Watchdog Timer Disabled)



Pin Pullup

Figure 174. I/O Pin Pull-Up Resistor Current vs. Input Voltage (V_{CC} = 5V)

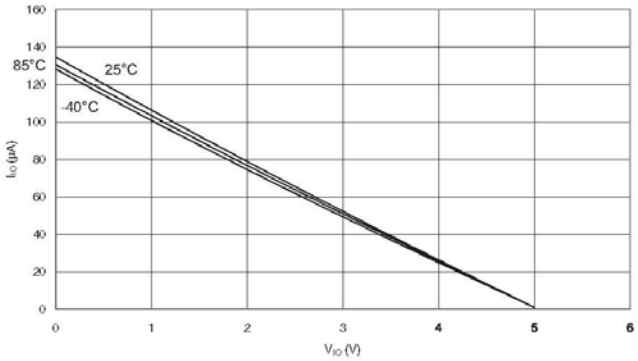


Figure 175. I/O Pin Pull-Up Resistor Current vs. Input Voltage (V_{CC} = 2.7V)

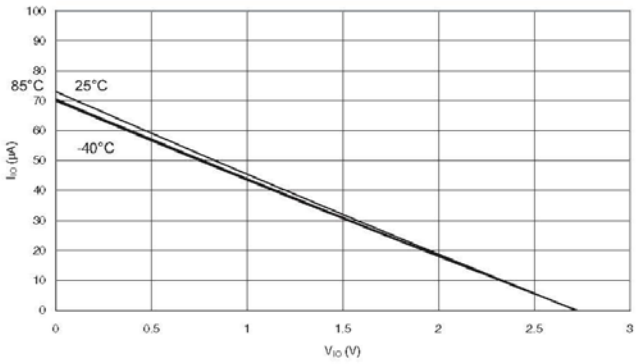


Figure 176. Reset Pull-Up Resistor Current vs. Reset Pin Voltage (V_{CC} = 5V)

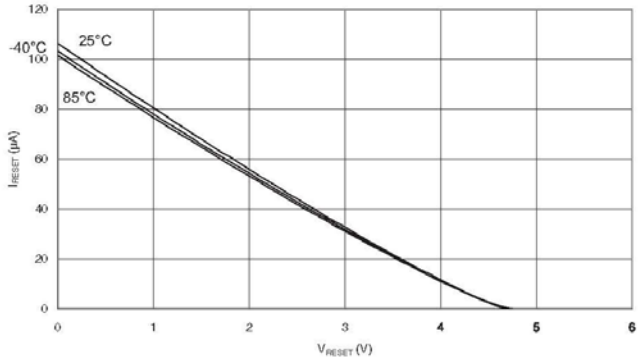
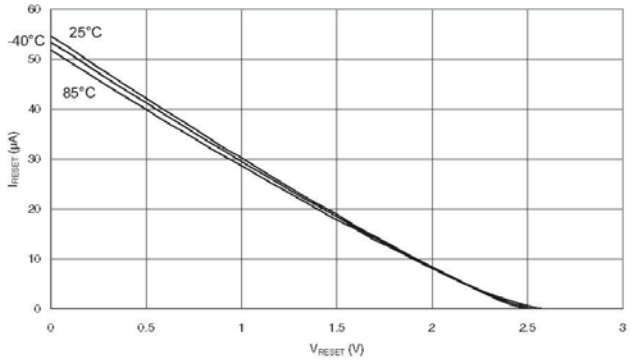


Figure 177. Reset Pull-Up Resistor Current vs. Reset Pin Voltage (V_{CC} = 2.7V)



Pin Driver Strength

Figure 178. I/O Pin Source Current vs. Output Voltage (V_{CC} = 5V)

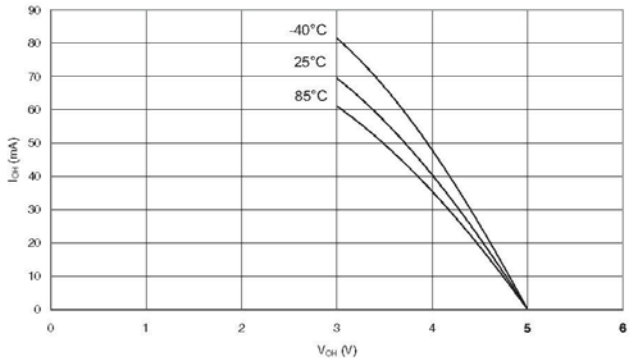


Figure 179. I/O Pin Source Current vs. Output Voltage (V_{CC} = 2.7V)

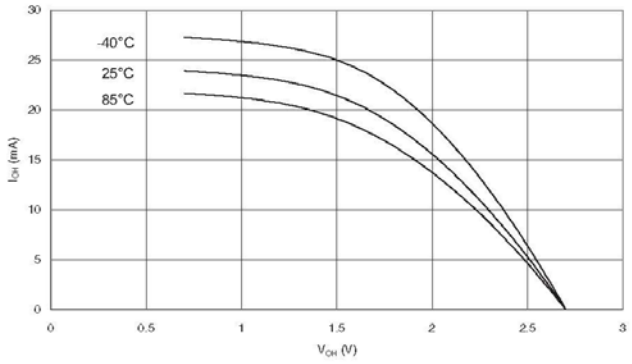


Figure 180. I/O Pin Sink Current vs. Output Voltage (V_{CC} = 5V)

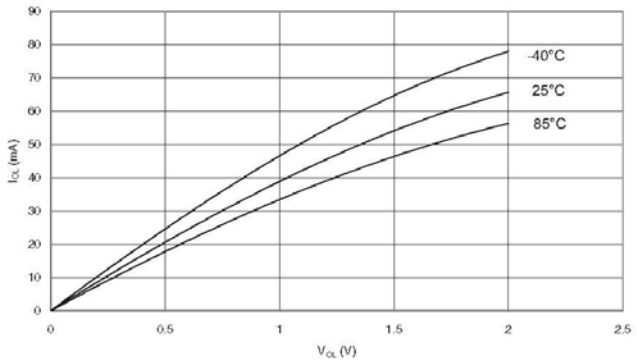
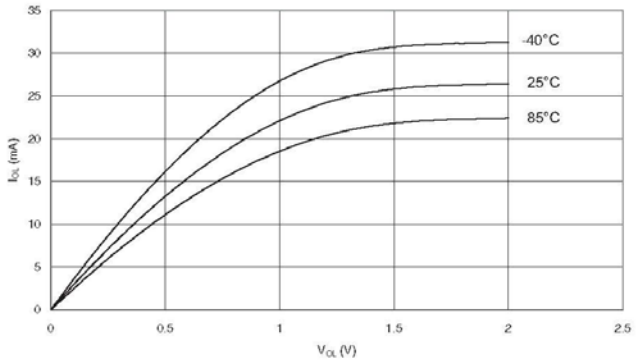


Figure 181. I/O Pin Sink Current vs. Output Voltage (V_{CC} = 2.7V)



Pin Thresholds And Hysteresis

Figure 182. I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin Read As '1')

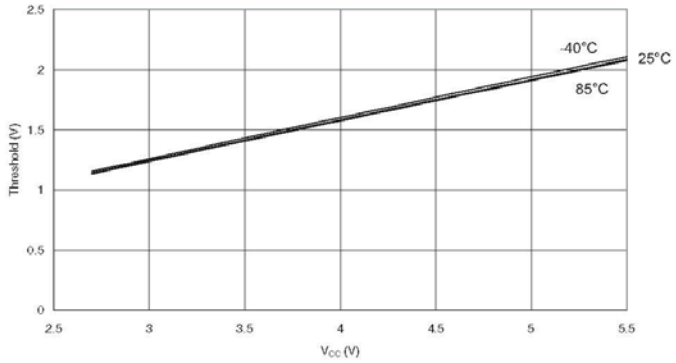


Figure 183. I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin Read As '0')

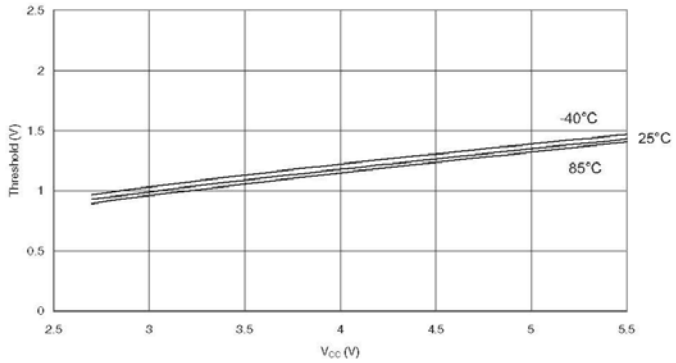


Figure 184. I/O Pin Input Hysteresis vs. V_{CC}

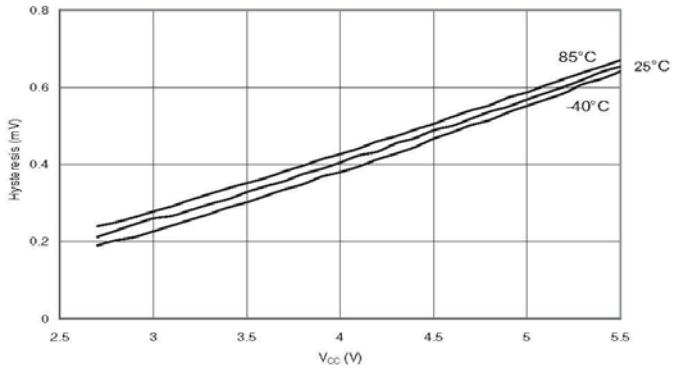


Figure 185. Reset Input Threshold Voltage vs. V_{CC} (V_{IH}, Reset Pin Read As '1')

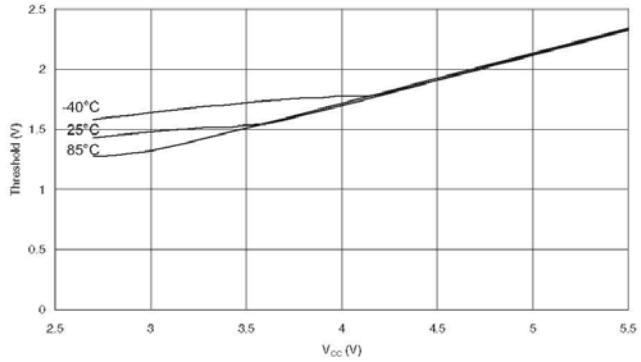


Figure 186. Reset Input Threshold Voltage vs. V_{CC} (V_{IL}, Reset Pin Read As '0')

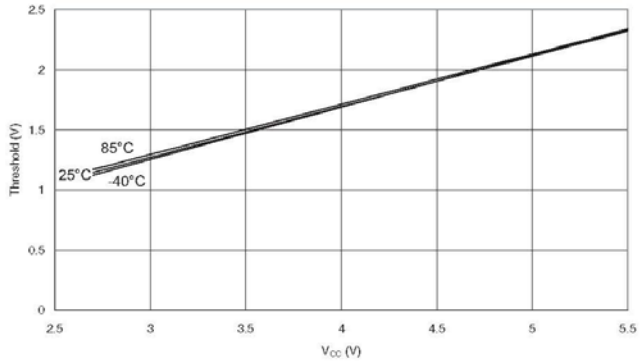
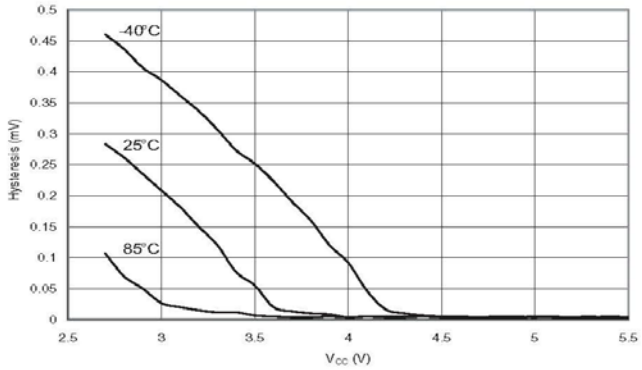


Figure 187. Reset Input Pin Hysteresis vs. V_{CC}



Bod Thresholds And Analog Comparator Offset

Figure 188. Bod Thresholds vs. Temperature (Bodlevel is 4.0V)

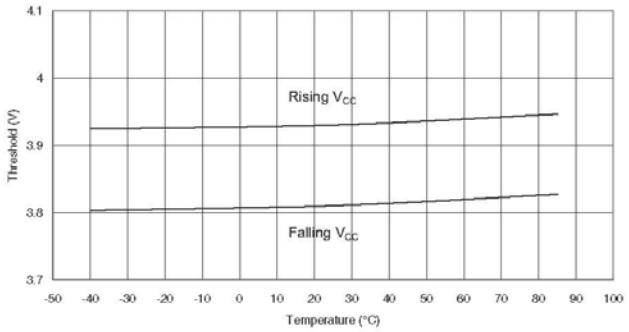


Figure 189. Bod Thresholds vs. Temperature (Bodlevel is 2.7V)

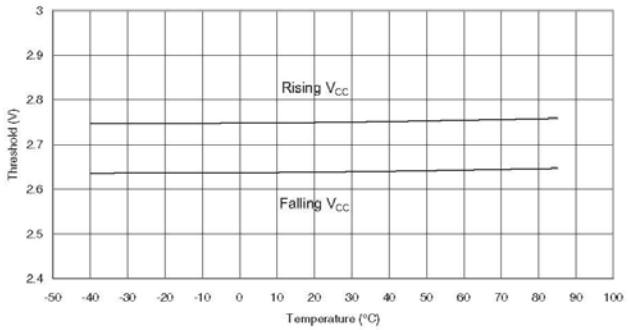


Figure 190. Bandgap Voltage vs. V_{CC}

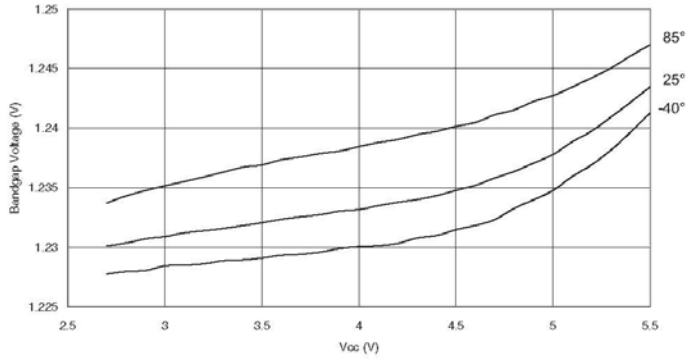


Figure 191. Analog Comparator Offset Voltage vs. Common Mode Voltage (V_{CC} = 5V)

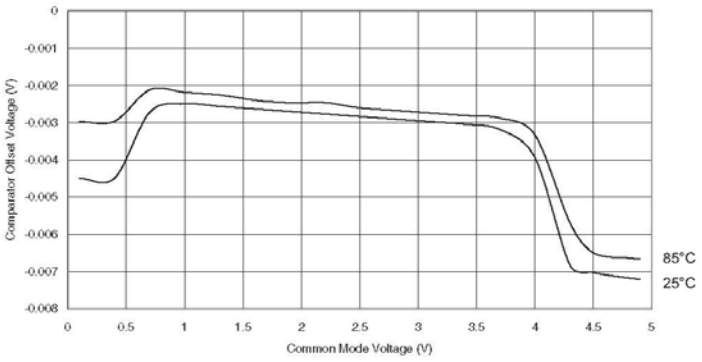
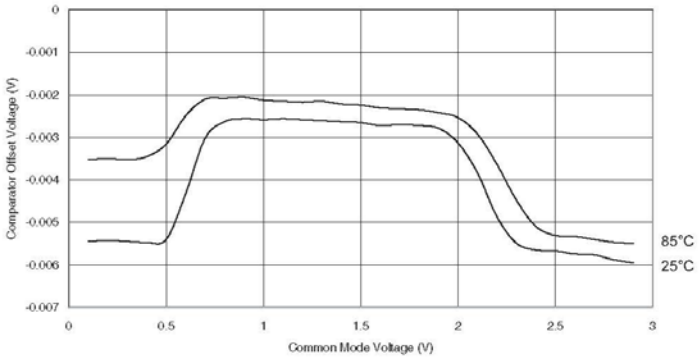


Figure 192. Analog Comparator Offset Voltage vs. Common Mode Voltage (V_{CC} = 3V)



Internal Oscillator Speed

Figure 193. Watchdog Oscillator Frequency vs. V_{CC}

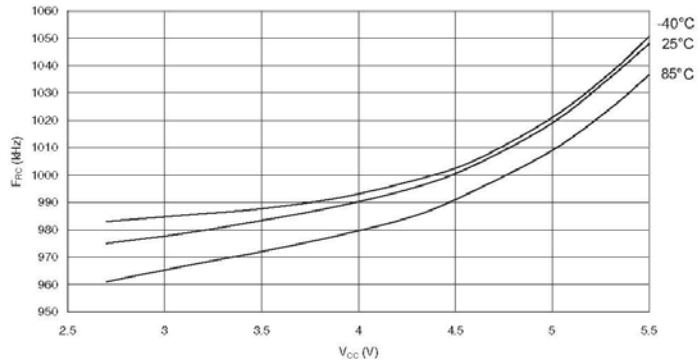


Figure 194. Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

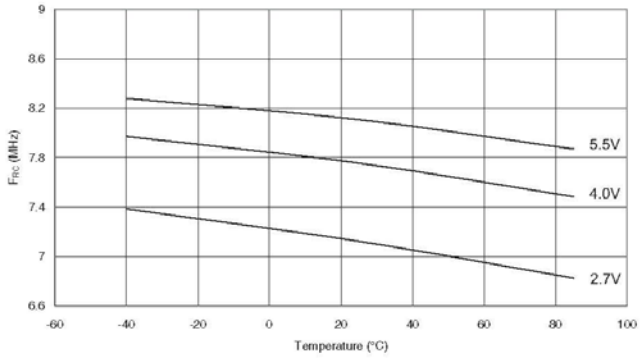


Figure 195. Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

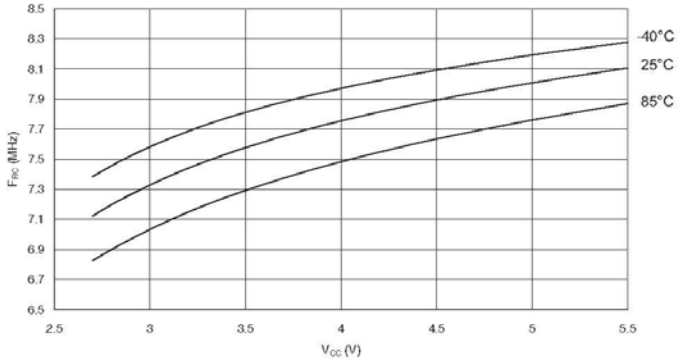


Figure 196. Calibrated 8 MHz RC Oscillator Frequency vs. Osccal Value

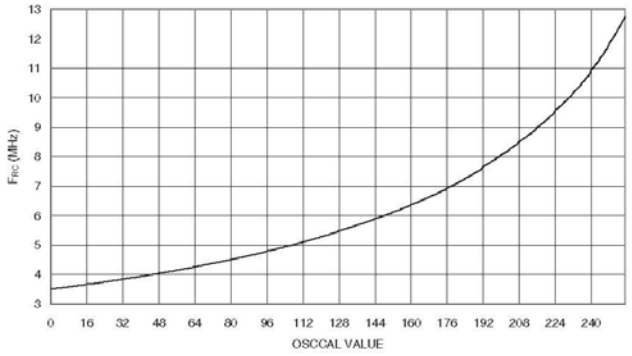


Figure 197. Calibrated 4 MHz RC Oscillator Frequency vs. Temperature

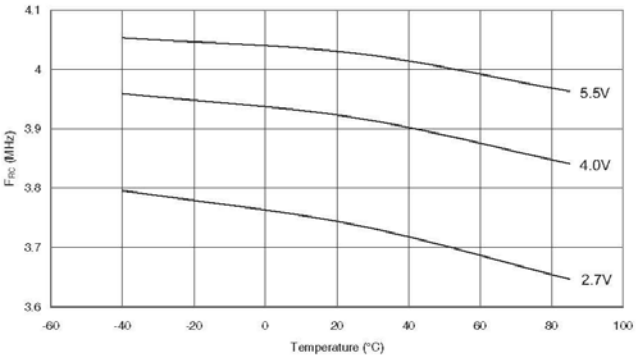


Figure 198. Calibrated 4 MHz RC Oscillator Frequency vs. V_{CC}

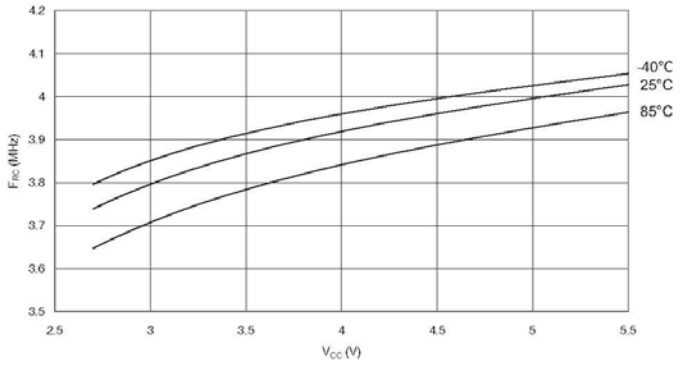


Figure 199. Calibrated 4 MHz RC Oscillator Frequency vs. Osccal Value

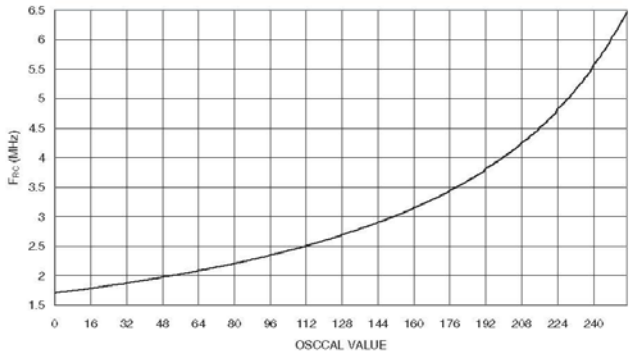


Figure 200. Calibrated 2 MHz RC Oscillator Frequency vs. Temperature

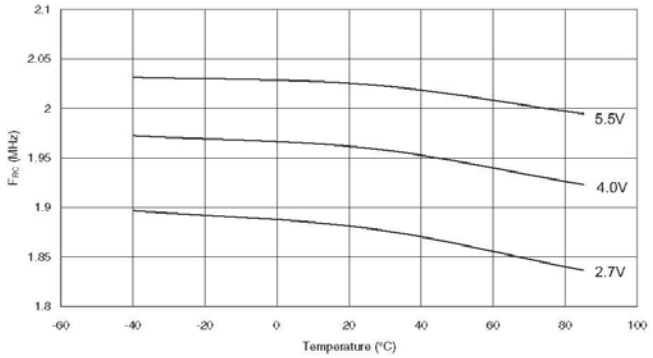


Figure 201. Calibrated 2 MHz RC Oscillator Frequency vs. V_{CC}

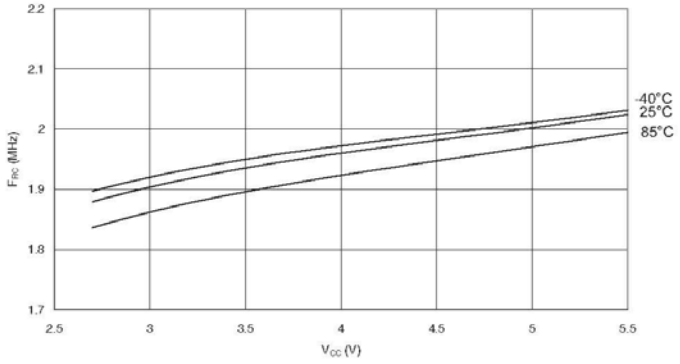


Figure 202. Calibrated 2 MHz RC Oscillator Frequency vs. Osccal Value

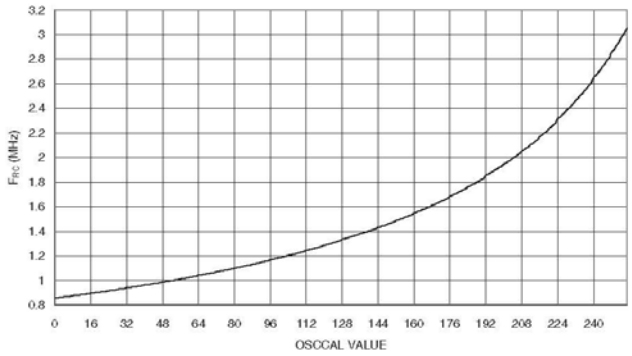


Figure 203. Calibrated 1 MHz RC Oscillator Frequency vs. Temperature

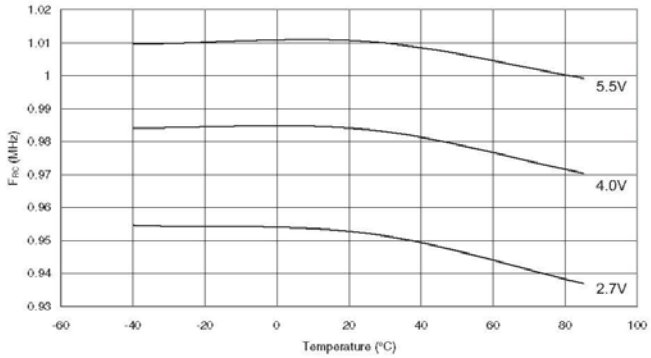


Figure 204. Calibrated 1 MHz RC Oscillator Frequency vs. V_{CC}

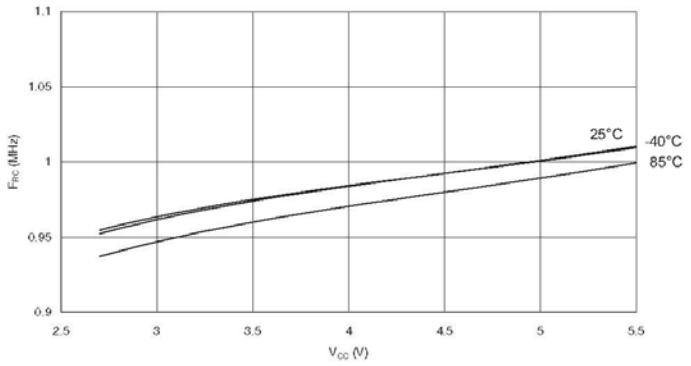
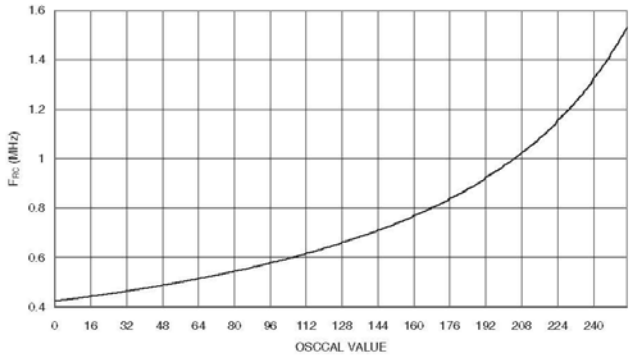


Figure 205. Calibrated 1 MHz RC Oscillator Frequency vs. Osccal Value



Current Consumption Of Peripheral Units

Figure 206. Brownout Detector Current vs. V_{CC}

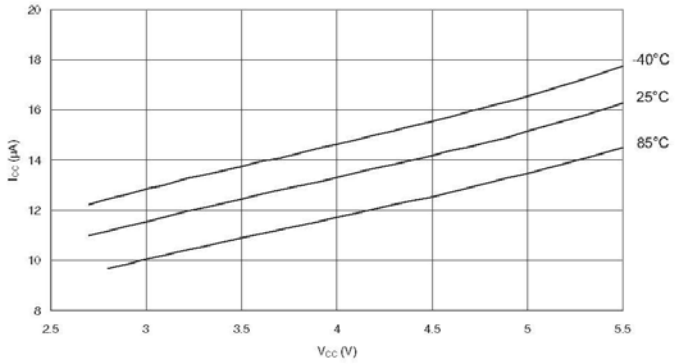


Figure 207. ADC Current vs. V_{CC}(Aref = AVCC)

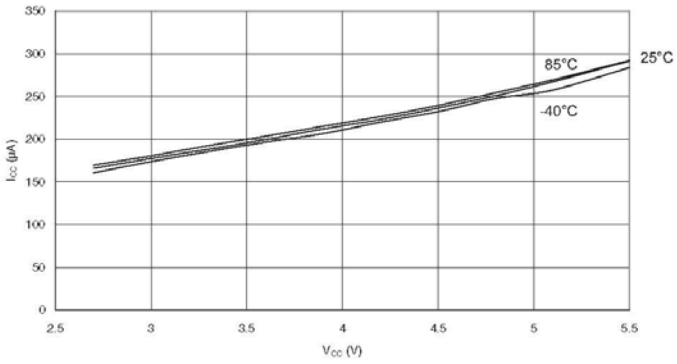


Figure 208. Aref External Reference Current vs. V_{CC}

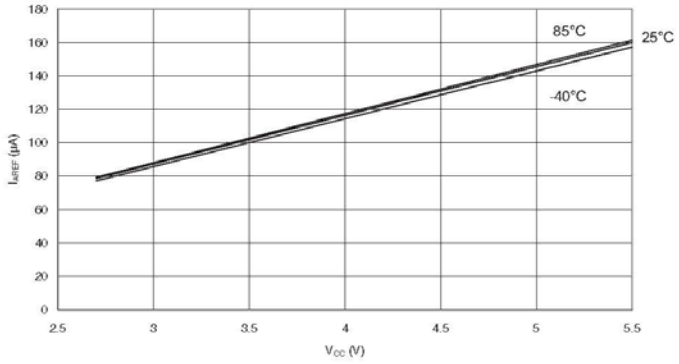


Figure 209. 32khz Tosc Current vs. V_{CC} (Watchdog Timer Disabled)

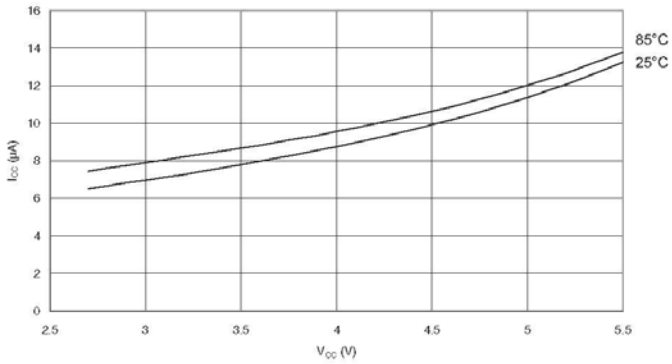


Figure 210. Watchdog Timer Current vs. V_{CC}

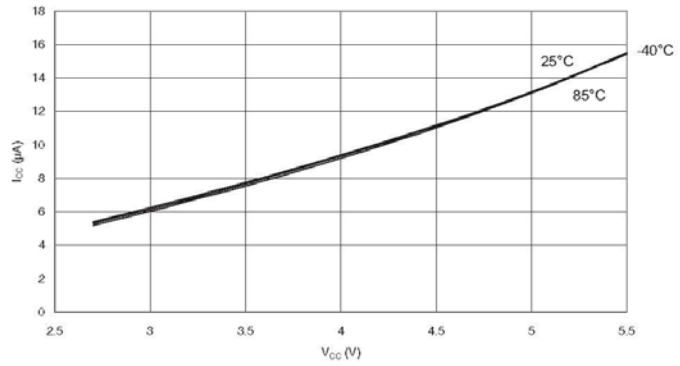
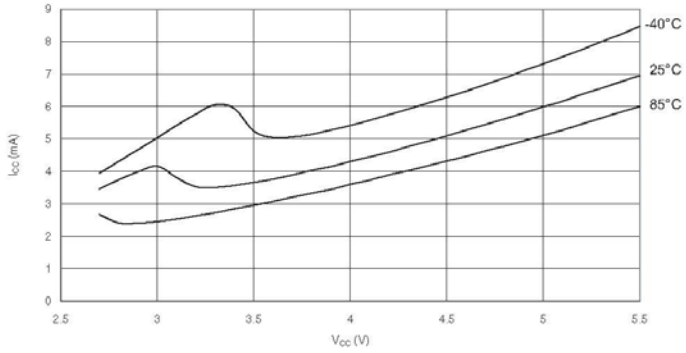


Figure 211. Programming Current vs. V_{CC}



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	9
$3E ($5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	12
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
$3C ($5C)	OCR0	Timer/Counter0 Output Compare Register								85
$3B ($5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	48, 69
$3A ($5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	70
$39 ($59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	85, 115, 133
$38 ($58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	86, 115, 133
$37 ($57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	250
$36 ($56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	180
$35 ($55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	32, 68
$34 ($54)	MCUCSR	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	41, 69, 231
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	83
$32 ($52)	TCNT0	Timer/Counter0 (8 Bits)								85
$31[1] ($51)[1]	OSCCAL	Oscillator Calibration Register								30
	OCDR	On-Chip Debug Register								227
$30 ($50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	57,88,134,201,221
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	110
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	113
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								114
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								114
$2B ($4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								114
$2A ($4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								114
$29 ($49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								114
$28 ($48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								114
$27 ($47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								114
$26 ($46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								114
$25 ($45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	128
$24 ($44)	TCNT2	Timer/Counter2 (8 Bits)								130
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								130
$22 ($42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	131
$21 ($41)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	43
$20[2] ($40)[2]	UBRRH	URSEL	–	–	–	UBRR[11:8]				167
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	166
$1F ($3F)	EEARH	–	–	–	–	–	–	–	EEAR8	19
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								19
$1D ($3D)	EEDR	EEPROM Data Register								19
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	19
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	66
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	66
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	66
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	66
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	66
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	66
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	67
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	67
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	67
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	67
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	67
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	67
$0F ($2F)	SPDR	SPI Data Register								142
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	142
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	140
$0C ($2C)	UDR	USART I/O Data Register								163
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	164
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	165
$09 ($29)	UBRRL	USART Baud Rate Register Low Byte								167
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	202
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	217
$06 ($26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	219
$05 ($25)	ADCH	ADC Data Register High Byte								220
$04 ($24)	ADCL	ADC Data Register Low Byte								220
$03 ($23)	TWDR	Two-wire Serial Interface Data Register								182
$02 ($22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	182

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$01 ($21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	181
$00 ($20)	TWBR	Two-wire Serial Interface Bit Rate Register								180

Notes:
1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debugger specific documentation for details on how to use the OCDR Register.
2. Refer to the USART description for details on how to access UBRRH and UCSRC.
3. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
4. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF – Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 – Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd – Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		**Clear Twos Complement Overflow**	**V ← 0**	**V**	**1**
SET		**Set T in SREG**	**T ← 1**	**T**	**1**
CLT		**Clear T in SREG**	**T ← 0**	**T**	**1**
SEH		**Set Half Carry Flag in SREG**	**H ← 1**	**H**	**1**

Mnemonics	Operands	Description	Operation	Flags	#Clocks
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-Chip Debug Only	None	N/A

Ordering Information

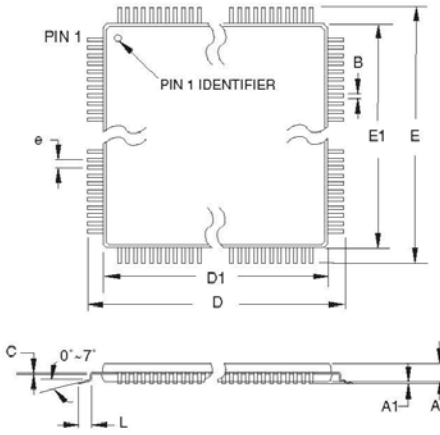
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7V - 5.5V	ATmega16L-8AU[1] ATmega16L-8PU[1] ATmega16L-8MU[1]	44A 40P6 44M1	Industrial (-40°C to 85°C)
16	4.5V - 5.5V	ATmega16-16AU[1] ATmega16-16PU[1] ATmega16-16MU[1]	44A 40P6 44M1	Industrial (-40°C to 85°C)

Note: 1. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44M1	44-pad, 7 × 7 × 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

Packaging Information

44A



COMMON DIMENSIONS
(Unit of Measure = mm)

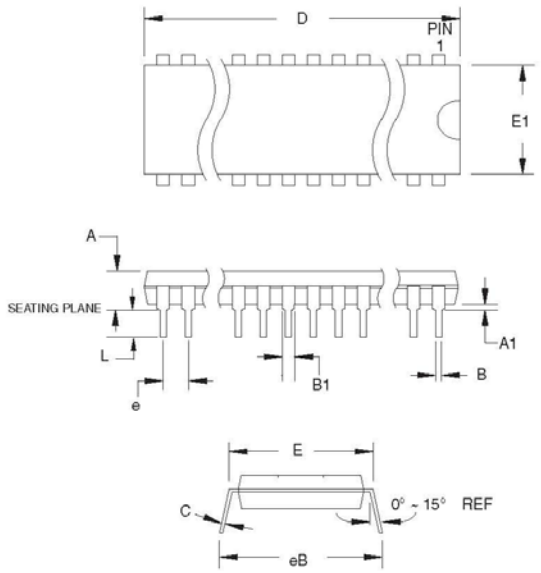
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**44A,** 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	44A	B

40P6



Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

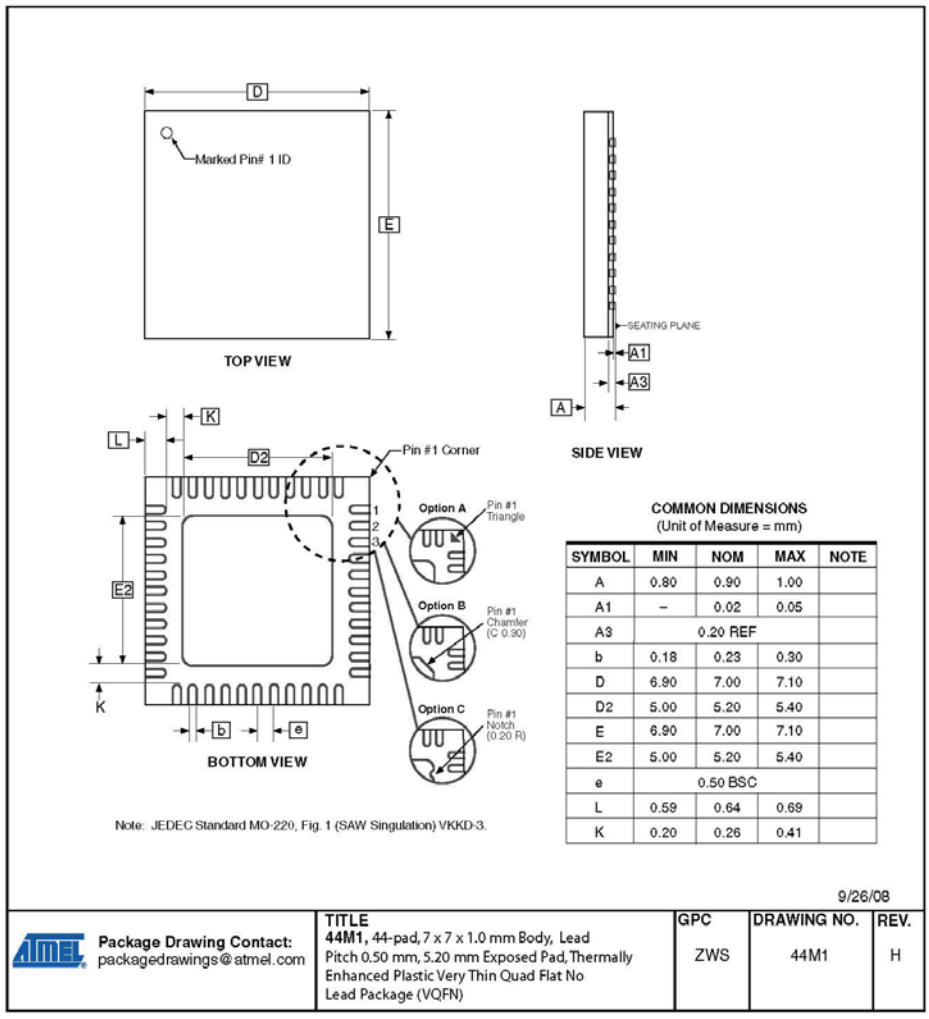
COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

09/28/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**40P6**, 40-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	40P6	B

44M1



Note: JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-3.

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A3		0.20 REF		
b	0.18	0.23	0.30	
D	6.90	7.00	7.10	
D2	5.00	5.20	5.40	
E	6.90	7.00	7.10	
E2	5.00	5.20	5.40	
e		0.50 BSC		
L	0.59	0.64	0.69	
K	0.20	0.26	0.41	

9/26/08

Package Drawing Contact: packagedrawings@atmel.com	TITLE	GPC	DRAWING NO.	REV.
	44M1, 44-pad, 7 x 7 x 1.0 mm Body, Lead Pitch 0.50 mm, 5.20 mm Exposed Pad, Thermally Enhanced Plastic Very Thin Quad Flat No Lead Package (VQFN)	ZWS	44M1	H

Errata

The revision letter in this section refers to the revision of the ATmega16 device.

ATmega16(L) Rev. M

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. **First Analog Comparator conversion may be delayed**

   If the device is powered by a slow rising V_{CC}, the first Analog Comparator conversion will take longer than expected on some devices.

   Problem Fix/Workaround

   When the device has been powered or reset, disable then enable theAnalog Comparator before the first conversion.

2. **Interrupts may be lost when writing the timer registers in the asynchronous timer**

   The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

   Problem Fix / Workaround

   Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

   - If ATmega16 is the only device in the scan chain, the problem is not visible.
   - Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
   - If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

4. **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.**

   Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. L

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. **First Analog Comparator conversion may be delayed**

   If the device is powered by a slow rising V_{CC}, the first Analog Comparator conversion will take longer than expected on some devices.

Problem Fix/Workaround

When the device has been powered or reset, disable then enable theAnalog Comparator before the first conversion.

2. **Interrupts may be lost when writing the timer registers in the asynchronous timer**

The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

Problem Fix / Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

4. **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.**

Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

Problem Fix / Workaround

Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. K

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. **First Analog Comparator conversion may be delayed**

If the device is powered by a slow rising V_{CC}, the first Analog Comparator conversion will take longer than expected on some devices.

Problem Fix/Workaround

When the device has been powered or reset, disable then enable theAnalog Comparator before the first conversion.

2. **Interrupts may be lost when writing the timer registers in the asynchronous timer**

The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

Problem Fix / Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

   – If ATmega16 is the only device in the scan chain, the problem is not visible.
   – Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
   – If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

4. **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.**

   Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. J

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. **First Analog Comparator conversion may be delayed**

   If the device is powered by a slow rising V_{CC}, the first Analog Comparator conversion will take longer than expected on some devices.

   Problem Fix/Workaround

   When the device has been powered or reset, disable then enable theAnalog Comparator before the first conversion.

2. **Interrupts may be lost when writing the timer registers in the asynchronous timer**

   The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

   Problem Fix / Workaround

   Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

4. **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.**

   Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. I

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. **First Analog Comparator conversion may be delayed**

   If the device is powered by a slow rising V_{CC}, the first Analog Comparator conversion will take longer than expected on some devices.

   Problem Fix/Workaround

   When the device has been powered or reset, disable then enable theAnalog Comparator before the first conversion.

2. **Interrupts may be lost when writing the timer registers in the asynchronous timer**

   The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

   Problem Fix / Workaround

   Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

   - If ATmega16 is the only device in the scan chain, the problem is not visible.
   - Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
   - If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

4. **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.**

   Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. H

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. **First Analog Comparator conversion may be delayed**

   If the device is powered by a slow rising V_{CC}, the first Analog Comparator conversion will take longer than expected on some devices.

   Problem Fix/Workaround

   When the device has been powered or reset, disable then enable theAnalog Comparator before the first conversion.

2. **Interrupts may be lost when writing the timer registers in the asynchronous timer**

   The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

   Problem Fix / Workaround

   Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

   – If ATmega16 is the only device in the scan chain, the problem is not visible.

   – Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.

   – If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

4. **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.**

   Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

Rev. 2466T-07/10

1. Corrected use of comma in formula Rp in Table 120, "Two-wire Serial Bus Requirements," on page 294.
2. Updated document according to Atmel's Technical Terminology
3. Note 6 and Note 7 under Table 120, "Two-wire Serial Bus Requirements," on page 294 have been removed.

Rev. 2466S-05/09

1. Updated "Errata" on page 340.
2. Updated the last page with Atmel's new adresses.

Rev. 2466R-06/08

1. Added "Not recommended for new designs" note in Figure on page 1.

Rev. 2466Q-05/08

1. Updated "Fast PWM Mode" on page 77 in "8-bit Timer/Counter0 with PWM" on page 71:
   – Removed the last section describing how to achieve a frequency with 50% duty cycle waveform output in fast PWM mode.
2. Removed note from Feature list in "Analog to Digital Converter" on page 204.
3. Removed note from Table 84 on page 218.
4. Updated "Ordering Information" on page 336:
   - Commercial ordering codes removed.
   - Non Pb-free package option removed.

Rev. 2466P-08/07

1. Updated "Features" on page 1.
2. Added "Data Retention" on page 6.
3. Updated "Errata" on page 340.
4. Updated "Slave Mode" on page 140.

Rev. 2466O-03/07

1. Updated "Calibrated Internal RC Oscillator" on page 29.
2. Updated C code example in "USART Initialization" on page 149.
3. Updated "ATmega16 Boundary-scan Order" on page 241.
4. Removed "premilinary" from "ADC Characteristics" on page 297.
5. Updated from V to mV in "I/O Pin Input Hysteresis vs. VCC" on page 317.
6. Updated from V to mV in "Reset Input Pin Hysteresis vs. VCC" on page 318.

Rev. 2466N-10/06

1. Updated "Timer/Counter Oscillator" on page 31.
2. Updated "Fast PWM Mode" on page 102.
3. Updated Table 38 on page 83, Table 40 on page 84, Table 45 on page 111, Table 47 on page 112, Table 50 on page 128 and Table 52 on page 129.
4. Updated C code example in "USART Initialization" on page 149.
5. Updated "Errata" on page 340.

Rev. 2466M-04/06

1. Updated typos.
2. Updated "Serial Peripheral Interface – SPI" on page 135.
3. Updated Table 86 on page 221, Table 116 on page 276 ,Table 121 on page 295 and Table 122 on page 297.

Rev. 2466L-06/05

1. Updated note in "Bit Rate Generator Unit" on page 178.
2. Updated values for V_{INT} in "ADC Characteristics" on page 297.
3. Updated "Serial Programming Instruction set" on page 276.
4. Updated USART init C-code example in "USART" on page 144.

Rev. 2466K-04/05

1. Updated "Ordering Information" on page 336.
2. MLF-package alternative changed to "Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF".
3. Updated "Electrical Characteristics" on page 291.

Rev. 2466J-10/04

1. Updated "Ordering Information" on page 336.

Rev. 2466I-10/04

1. Removed references to analog ground.
2. Updated Table 7 on page 28, Table 15 on page 38, Table 16 on page 42, Table 81 on page 209, Table 116 on page 276, and Table 119 on page 293.
3. Updated "Pinout ATmega16" on page 2.
4. Updated features in "Analog to Digital Converter" on page 204.
5. Updated "Version" on page 229.
6. Updated "Calibration Byte" on page 261.
7. Added "Page Size" on page 262.

Rev. 2466H-12/03

1. Updated "Calibrated Internal RC Oscillator" on page 29.

Rev. 2466G-10/03

1. Removed "Preliminary" from the datasheet.
2. Changed ICP to ICP1 in the datasheet.
3. Updated "JTAG Interface and On-chip Debug System" on page 36.
4. Updated assembly and C code examples in "Watchdog Timer Control Register – WDTCR" on page 43.
5. Updated Figure 46 on page 103.
6. Updated Table 15 on page 38, Table 82 on page 217 and Table 115 on page 276.
7. Updated "Test Access Port – TAP" on page 222 regarding JTAGEN.
8. Updated description for the JTD bit on page 231.
9. Added note 2 to Figure 126 on page 252.
10. Added a note regarding JTAGEN fuse to Table 105 on page 260.
11. Updated Absolute Maximum Ratings* and DC Characteristics in "Electrical Characteristics" on page 291.
12. Updated "ATmega16 Typical Characteristics" on page 299.
13. Fixed typo for 16 MHz QFN/MLF package in "Ordering Information" on page 336.
14. Added a proposal for solving problems regarding the JTAG instruction IDCODE in "Errata" on page 340.

Rev. 2466F-02/03

1. Added note about masking out unused bits when reading the Program Counter in "Stack Pointer" on page 12.
2. Added Chip Erase as a first step in "Programming the Flash" on page 288 and "Programming the EEPROM" on page 289.
3. Added the section "Unconnected pins" on page 55.
4. Added tips on how to disable the OCD system in "On-chip Debug System" on page 34.
5. Removed reference to the "Multi-purpose Oscillator" application note and "32 kHz Crystal Oscillator" application note, which do not exist.
6. Added information about PWM symmetry for Timer0 and Timer2.
7. Added note in "Filling the Temporary Buffer (Page Loading)" on page 253 about writing to the EEPROM during an SPM Page Load.
8. Removed ADHSM completely.

9. Added Table 73, "TWI Bit Rate Prescaler," on page 182 to describe the TWPS bits in the "TWI Status Register – TWSR" on page 181.
10. Added section "Default Clock Source" on page 25.
11. Added note about frequency variation when using an external clock. Note added in "External Clock" on page 31. An extra row and a note added in Table 118 on page 293.
12. Various minor TWI corrections.
13. Added "Power Consumption" data in "Features" on page 1.
14. Added section "EEPROM Write During Power-down Sleep Mode" on page 22.
15. Added note about Differential Mode with Auto Triggering in "Prescaling and Conversion Timing" on page 207.
16. Added updated "Packaging Information" on page 337.

Rev. 2466E-10/02

1. Updated "DC Characteristics" on page 291.

Rev. 2466D-09/02

1. Changed all Flash write/erase cycles from 1,000 to 10,000.
2. Updated the following tables: Table 4 on page 26, Table 15 on page 38, Table 42 on page 85, Table 45 on page 111, Table 46 on page 111, Table 59 on page 143, Table 67 on page 167, Table 90 on page 235, Table 102 on page 258, "DC Characteristics" on page 291, Table 119 on page 293, Table 121 on page 295, and Table 122 on page 297.
3. Updated "Errata" on page 340.

Rev. 2466C-03/02

1. Updated typical EEPROM programming time, Table 1 on page 20.
2. Updated typical start-up time in the following tables:
   Table 3 on page 25, Table 5 on page 27, Table 6 on page 28, Table 8 on page 29, Table 9 on page 29, and Table 10 on page 29.
3. Updated Table 17 on page 43 with typical WDT Time-out.
4. Added Some Preliminary Test Limits and Characterization Data.
   Removed some of the TBD's in the following tables and pages:
   Table 15 on page 38, Table 16 on page 42, Table 116 on page 272 (table removed in document review #D), "Electrical Characteristics" on page 291, Table 119 on page 293, Table 121 on page 295, and Table 122 on page 297.
5. Updated TWI Chapter.
   Added the note at the end of the "Bit Rate Generator Unit" on page 178.
6. Corrected description of ADSC bit in "ADC Control and Status Register A – ADCSRA" on page 219.
7. Improved description on how to do a polarity check of the ADC doff results in "ADC Conversion Result" on page 216.

8. **Added JTAG version number for rev. H in Table 87 on page 229.**

9. **Added not regarding OCDEN Fuse below Table 105 on page 260.**

10. **Updated Programming Figures:**
    Figure 127 on page 262 and Figure 136 on page 274 are updated to also reflect that AVCC must be connected during Programming mode. Figure 131 on page 270 added to illustrate how to program the fuses.

11. **Added a note regarding usage of the "PROG_PAGELOAD ($6)" on page 280 and "PROG_PAGEREAD ($7)" on page 280.**

12. **Removed alternative algortihm for leaving JTAG Programming mode.**
    See "Leaving Programming Mode" on page 288.

13. **Added Calibrated RC Oscillator characterization curves in section "ATmega16 Typical Characteristics" on page 299.**

14. **Corrected ordering code for QFN/MLF package (16 MHz) in "Ordering Information" on page 336.**

15. **Corrected Table 90, "Scan Signals for the Oscillators(1)(2)(3)," on page 235.**

Table of Contents

Table of Contents i

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

2466T–AVR–07/10

PIC16F84A
Data Sheet

18-pin Enhanced FLASH/EEPROM
8-bit Microcontroller

Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable".
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV MSC The Netherlands Accredited by the RvA — DNV — ISO 9001 / QS-9000 REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

MICROCHIP

PIC16F84A

18-pin *Enhanced* FLASH/EEPROM 8-Bit Microcontroller

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt-on-change
  - Data EEPROM write complete

Peripheral Features:

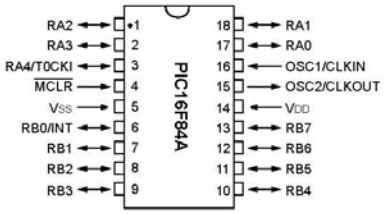
- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features:

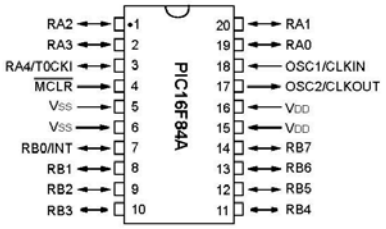
- 10,000 erase/write cycles *Enhanced* FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

Pin Diagrams

PDIP, SOIC



SSOP



CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 kHz
  - < 0.5 µA typical standby current @ 2V

PIC16F84A

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com/cn** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F84A device. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F84A belongs to the mid-range family of the PICmicro® microcontroller devices. A block diagram of the device is shown in Figure 1-1.

The program memory contains 1K words, which translates to 1024 instructions, since each 14-bit program memory word is the same width as each device instruction. The data memory (RAM) contains 68 bytes. Data EEPROM is 64 bytes.

There are also 13 I/O pins that are user-configured on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

- External interrupt
- Change on PORTB interrupt
- Timer0 clock input

Table 1-1 details the pinout of the device with descriptions and details for each pin.

FIGURE 1-1: PIC16F84A BLOCK DIAGRAM

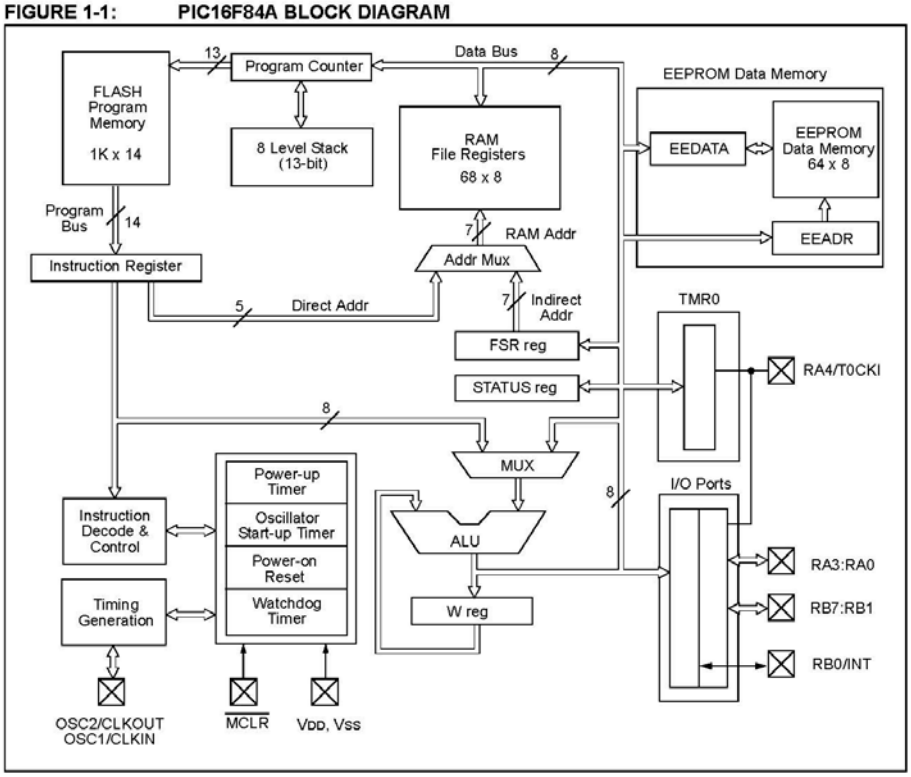


TABLE 1-1: PIC16F84A PINOUT DESCRIPTION

Pin Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	18	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0	17	17	19	I/O	TTL	
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	7	I/O	TTL/ST(1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	Interrupt-on-change pin.
RB5	11	11	12	I/O	TTL	Interrupt-on-change pin.
RB6	12	12	13	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming clock.
RB7	13	13	14	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming data.
VSS	5	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = Output I/O = Input/Output P = Power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F84A. These are the program memory and the data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle.

The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0h-3Fh. More details on the EEPROM memory can be found in Section 3.0.

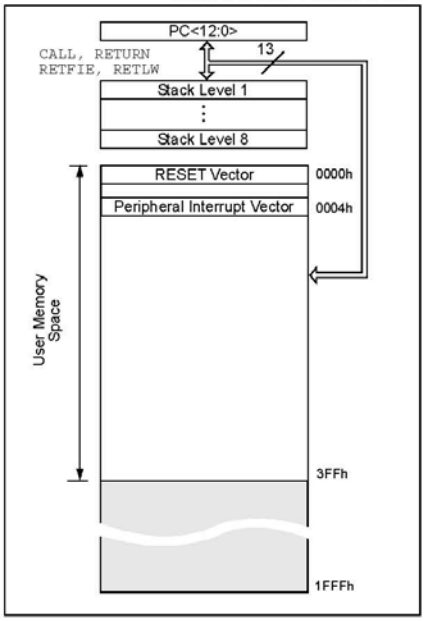
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16FXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F84A, the first 1K x 14 (0000h-03FFh) are physically implemented (Figure 2-1). Accessing a location above the physically implemented address will cause a wraparound. For example, for locations 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h, and 1C20h, the instruction will be the same.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK - PIC16F84A



2.2 Data Memory Organization

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area, while the second is the General Purpose Registers (GPR) area. The SFRs control the operation of the device.

Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM. The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. Figure 2-2 shows the data memory map organization.

Instructions `MOVWF` and `MOVF` can move values from the W register to any location in the register file ("F"), and vice-versa.

The entire data memory can be accessed either directly using the absolute address of each register file or indirectly through the File Select Register (FSR) (Section 2.5). Indirect addressing uses the present value of the RP0 bit for access into the banked areas of data memory.

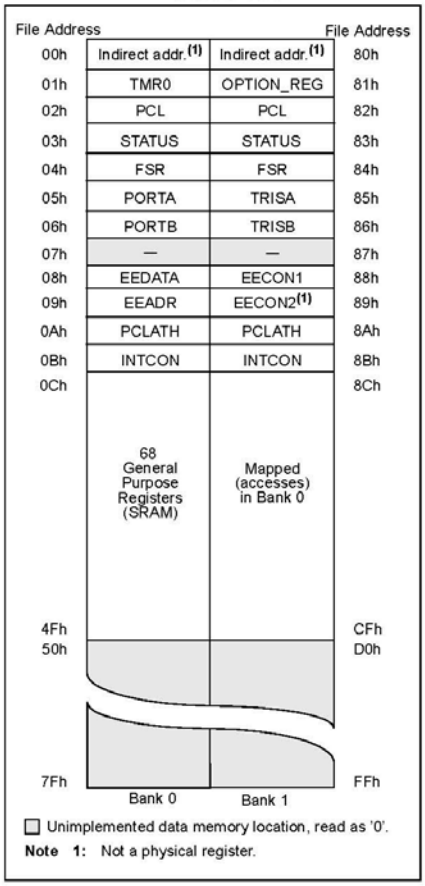
Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the RP0 bit (STATUS<5>). Setting the RP0 bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The first twelve locations of each Bank are reserved for the Special Function Registers. The remainder are General Purpose Registers, implemented as static RAM.

2.2.1 GENERAL PURPOSE REGISTER FILE

Each General Purpose Register (GPR) is 8-bits wide and is accessed either directly or indirectly through the FSR (Section 2.5).

The GPR addresses in Bank 1 are mapped to addresses in Bank 0. As an example, addressing location 0Ch or 8Ch will access the same GPR.

FIGURE 2-2: REGISTER FILE MAP - PIC16F84A



2.3 Special Function Registers

The Special Function Registers (Figure 2-2 and Table 2-1) are used by the CPU and Peripheral functions to control the device operation. These registers are static RAM.

The special function registers can be classified into two sets, core and peripheral. Those associated with the core functions are described in this section. Those related to the operation of the peripheral features are described in the section for that specific feature.

TABLE 2-1: SPECIAL FUNCTION REGISTER FILE SUMMARY

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page
Bank 0											
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- ----	11
01h	TMR0	8-bit Real-Time Clock/Counter								xxxx xxxx	20
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11
03h	STATUS(2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11
05h	PORTA(4)	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	16
06h	PORTB(5)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	18
07h	—	Unimplemented location, read as '0'								—	—
08h	EEDATA	EEPROM Data Register								xxxx xxxx	13,14
09h	EEADR	EEPROM Address Register								xxxx xxxx	13,14
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC(1)					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10
Bank 1											
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
83h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	16
86h	TRISB	PORTB Data Direction Register								1111 1111	18
87h	—	Unimplemented location, read as '0'								—	—
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
89h	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	14
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC(1)					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends on condition

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> are never transferred to PCLATH.

2: The $\overline{TO}$ and $\overline{PD}$ status bits in the STATUS register are not affected by a $\overline{MCLR}$ Reset.

3: Other (non power-up) RESETS include: external RESET through $\overline{MCLR}$ and the Watchdog Timer Reset.

4: On any device RESET, these pins are configured as inputs.

5: This is the value that will be in the port output latch.

2.3.1 STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bit for data memory.

As with any register, the STATUS register can be the destination for any instruction. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u uluu` (where `u` = unchanged).

Only the `BCF, BSF, SWAPF` and `MOVWF` instructions should be used to alter the STATUS register (Table 7-2), because these instructions do not affect any status bit.

Note 1: The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16F84A and should be programmed as cleared. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

2: The C and DC bits operate as a $\overline{borrow}$ and digit $\overline{borrow}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

3: When the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. The specified bit(s) will be updated according to device logic

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7-6 **Unimplemented:** Maintain as '0'

bit 5 **RP0**: Register Bank Select bits (used for direct addressing)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)

bit 4 $\overline{TO}$: Time-out bit
1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3 $\overline{PD}$: Power-down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

bit 2 **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{borrow}$ bit (`ADDWF, ADDLW, SUBLW, SUBWF` instructions) (for $\overline{borrow}$, the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ bit (`ADDWF, ADDLW, SUBLW, SUBWF` instructions) (for $\overline{borrow}$, the polarity is reversed)
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

Note: A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF, RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.3.2 OPTION REGISTER

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note: When the prescaler is assigned to the WDT (PSA = '1'), TMR0 has a 1:1 prescaler assignment.

REGISTER 2-2: OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.3.3 INTCON REGISTER

The INTCON register is a readable and writable register that contains the various enable bits for all interrupt sources.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **EEIE:** EE Write Complete Interrupt Enable bit
1 = Enables the EE Write Complete interrupts
0 = Disables the EE Write Complete interrupt

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.4 PCL and PCLATH

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 13 bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<12:8> bits and is not directly readable or writable. If the program counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP. All updates to the PCH register go through the PCLATH register.

2.4.1 STACK

The stack allows a combination of up to 8 program calls and interrupts to occur. The stack contains the return address from this branch in program execution.

Mid-range devices have an 8 level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not modified when the stack is PUSHed or POPed.

After the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

2.5 Indirect Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

EXAMPLE 2-1: INDIRECT ADDRESSING

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 06)
- A read of the INDF register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although STATUS bits may be affected).

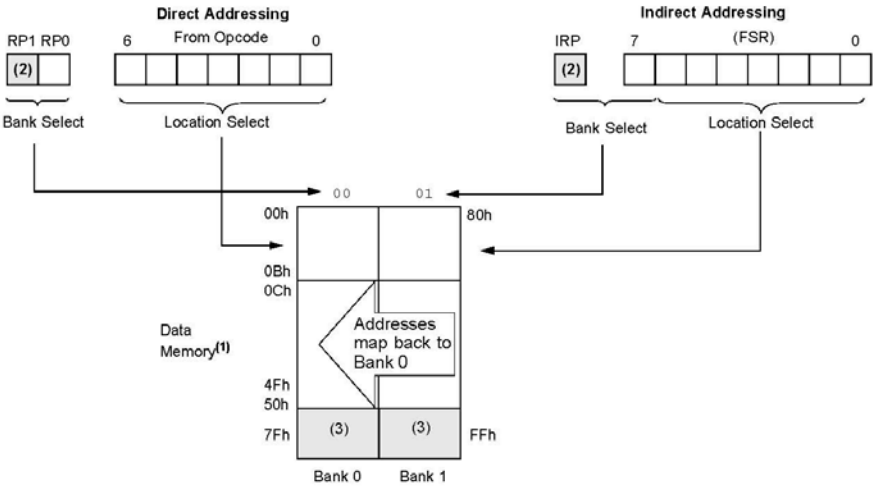
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

```
        movlw   0x20    ;initialize pointer
        movwf   FSR     ;to RAM
NEXT    clrf    INDF    ;clear INDF register
        incf    FSR     ;inc pointer
        btfss   FSR,4   ;all done?
        goto    NEXT    ;NO, clear next
CONTINUE
         :              ;YES, continue
```

An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-3. However, IRP is not used in the PIC16F84A.

FIGURE 2-3: DIRECT/INDIRECT ADDRESSING



Note 1: For memory map detail, see Figure 2-2.
2: Maintain as clear for upward compatibility with future products.
3: Not implemented.

3.0 DATA EEPROM MEMORY

The EEPROM data memory is readable and writable during normal operation (full VDD range). This memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers. There are four SFRs used to read and write this memory. These registers are:

- EECON1
- EECON2 (not a physically implemented register)
- EEDATA
- EEADR

EEDATA holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. PIC16F84A devices have 64 bytes of data EEPROM with an address range from 0h to 3Fh.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write-time will vary with voltage and temperature as well as from chip to chip. Please refer to AC specifications for exact limits.

When the device is code protected, the CPU may continue to read and write the data EEPROM memory. The device programmer can no longer access this memory.

Additional information on the Data EEPROM is available in the PICmicro™ Mid-Range Reference Manual (DS33023).

REGISTER 3-1: EECON1 REGISTER (ADDRESS 88h)

U-0	U-0	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
—	—	—	EEIF	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation is not complete or has not been started

bit 3 **WRERR:** EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{MCLR}$ Reset or any WDT Reset during normal operation)
0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the EEPROM

bit 1 **WR:** Write Control bit
1 = Initiates a write cycle. The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.
0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
0 = Does not initiate an EEPROM read

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3.1 Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next cycle, in the EEDATA register; therefore, it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 3-1: DATA EEPROM READ

```
BCF     STATUS, RP0  ; Bank 0
MOVLW   CONFIG_ADDR  ;
MOVWF   EEADR        ; Address to read
BSF     STATUS, RP0  ; Bank 1
BSF     EECON1, RD   ; EE Read
BCF     STATUS, RP0  ; Bank 0
MOVF    EEDATA, W    ; W = EEDATA
```

3.2 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte.

EXAMPLE 3-2: DATA EEPROM WRITE

```
        BSF     STATUS, RP0  ; Bank 1
        BCF     INTCON, GIE  ; Disable INTs.
        BSF     EECON1, WREN ; Enable Write
        MOVLW   55h          ;
Required
Sequence
        MOVWF   EECON2       ; Write 55h
        MOVLW   AAh          ;
        MOVWF   EECON2       ; Write AAh
        BSF     EECON1,WR    ; Set WR bit
                             ;   begin write
        BSF     INTCON, GIE  ; Enable INTs.
```

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

3.3 Write Verify

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM should be verified (Example 3-3) to the desired value to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit.

Generally, the EEPROM write failure will be a bit which was written as a '0', but reads back as a '1' (due to leakage off the bit).

EXAMPLE 3-3: WRITE VERIFY

```
        BCF   STATUS,RP0 ; Bank 0
        :                ; Any code
        :                ; can go here
        MOVF  EEDATA,W   ; Must be in Bank 0
        BSF   STATUS,RP0 ; Bank 1
READ
        BSF   EECON1, RD ; YES, Read the
                         ; value written
        BCF   STATUS, RP0 ; Bank 0
                         ;
                         ; Is the value written
                         ; (in W reg) and
                         ; read (in EEDATA)
                         ; the same?
                         ;
        SUBWF EEDATA, W  ;
        BTFSS STATUS, Z  ; Is difference 0?
        GOTO  WRITE_ERR  ; NO, Write error
```

TABLE 3-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other RESETS
08h	EEDATA	EEPROM Data Register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM Address Register								xxxx xxxx	uuuu uuuu
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM Control Register 2								---- ----	---- ----

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends upon condition.
Shaded cells are not used by data EEPROM.

4.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range Reference Manual (DS33023).

4.1 PORTA and TRISA Registers

PORTA is a 5-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Note:	On a Power-on Reset, these pins are configured as inputs and read as '0'.

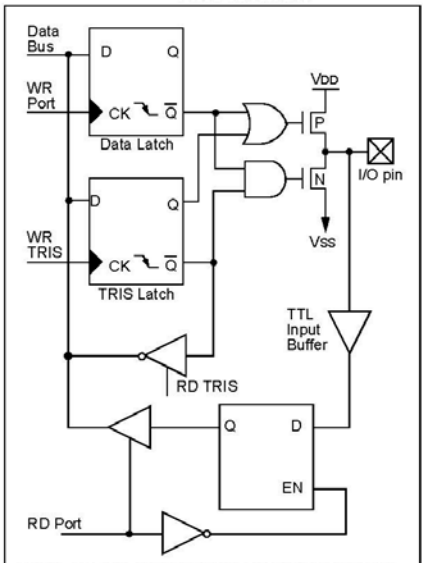
Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read. This value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

EXAMPLE 4-1: INITIALIZING PORTA

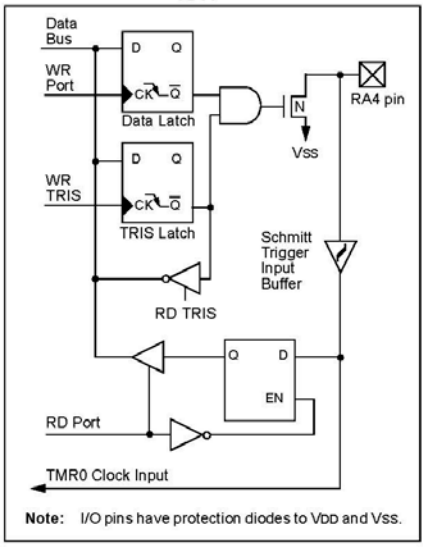
```
BCF     STATUS, RP0 ;
CLRF    PORTA       ; Initialize PORTA by
                    ; clearing output
                    ; data latches
BSF     STATUS, RP0 ; Select Bank 1
MOVLW   0x0F        ; Value used to
                    ; initialize data
                    ; direction
MOVWF   TRISA       ; Set RA<3:0> as inputs
                    ; RA4 as output
                    ; TRISA<7:5> are always
                    ; read as '0'.
```

FIGURE 4-1: BLOCK DIAGRAM OF PINS RA3:RA0



Note: I/O pins have protection diodes to VDD and Vss.

FIGURE 4-2: BLOCK DIAGRAM OF PIN RA4



Note: I/O pins have protection diodes to VDD and Vss.

TABLE 4-1: PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Input/output
RA1	bit1	TTL	Input/output
RA2	bit2	TTL	Input/output
RA3	bit3	TTL	Input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open drain type.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 4-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other RESETS
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are unimplemented, read as '0'.

4.2 PORTB and TRISB Registers

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

EXAMPLE 4-2: INITIALIZING PORTB

```
BCF     STATUS, RP0 ;
CLRF    PORTB       ; Initialize PORTB by
                    ; clearing output
                    ; data latches
BSF     STATUS, RP0 ; Select Bank 1
MOVLW   0xCF        ; Value used to
                    ; initialize data
                    ; direction
MOVWF   TRISB       ; Set RB<3:0> as inputs
                    ; RB<5:4> as outputs
                    ; RB<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{\text{RBPU}}$ (OPTION<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of PORTB's pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

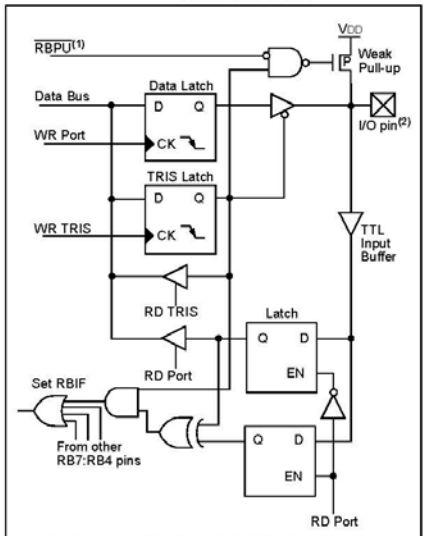
This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

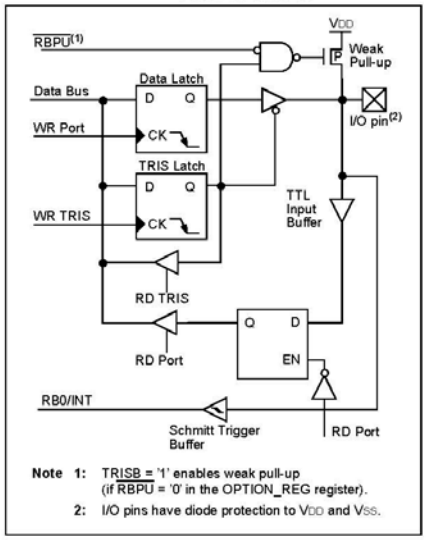
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

FIGURE 4-3: BLOCK DIAGRAM OF PINS RB7:RB4



Note 1: TRISB = '1' enables weak pull-up (if $\overline{\text{RBPU}}$ = '0' in the OPTION_REG register).
2: I/O pins have diode protection to VDD and VSS.

FIGURE 4-4: BLOCK DIAGRAM OF PINS RB3:RB0



Note 1: TRISB = '1' enables weak pull-up (if $\overline{\text{RBPU}}$ = '0' in the OPTION_REG register).
2: I/O pins have diode protection to VDD and VSS.

TABLE 4-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	I/O Consistency Function
RB0/INT	bit0	TTL/ST(1)	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST(2)	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST(2)	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger.
Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

TABLE 4-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other RESETS
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
0Bh,8Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- Internal or external clock select
- Edge select for external clock
- 8-bit software programmable prescaler
- Interrupt-on-overflow from FFh to 00h

Figure 5-1 is a simplified block diagram of the Timer0 module.

Additional information on timer modules is available in the PICmicro™ Mid-Range Reference Manual (DS33023).

5.1 Timer0 Operation

Timer0 can operate as a timer or as a counter.

Timer mode is selected by clearing bit T0CS (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed below.

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (Tosc). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

Additional information on external clock requirements is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

5.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer, respectively (Figure 5-2). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The prescaler is not readable or writable.

The PSA and PS2:PS0 bits (OPTION_REG<3:0>) determine the prescaler assignment and prescale ratio.

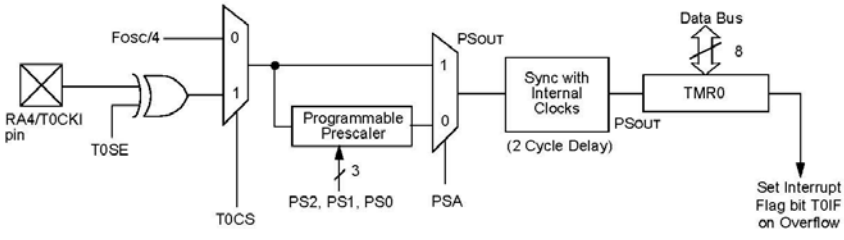
Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable.

Setting bit PSA will assign the prescaler to the Watchdog Timer (WDT). When the prescaler is assigned to the WDT, prescale values of 1:1, 1:2, ..., 1:128 are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1`, `MOVWF 1`, `BSF 1`, etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the WDT.

Note:	Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count, but will not change the prescaler assignment.

FIGURE 5-1: TIMER0 BLOCK DIAGRAM



Note 1: T0CS, T0SE, PSA, PS2:PS0 (OPTION_REG<5:0>).
2: The prescaler is shared with Watchdog Timer (refer to Figure 5-2 for detailed block diagram).

5.2.1 SWITCHING PRESCALER ASSIGNMENT

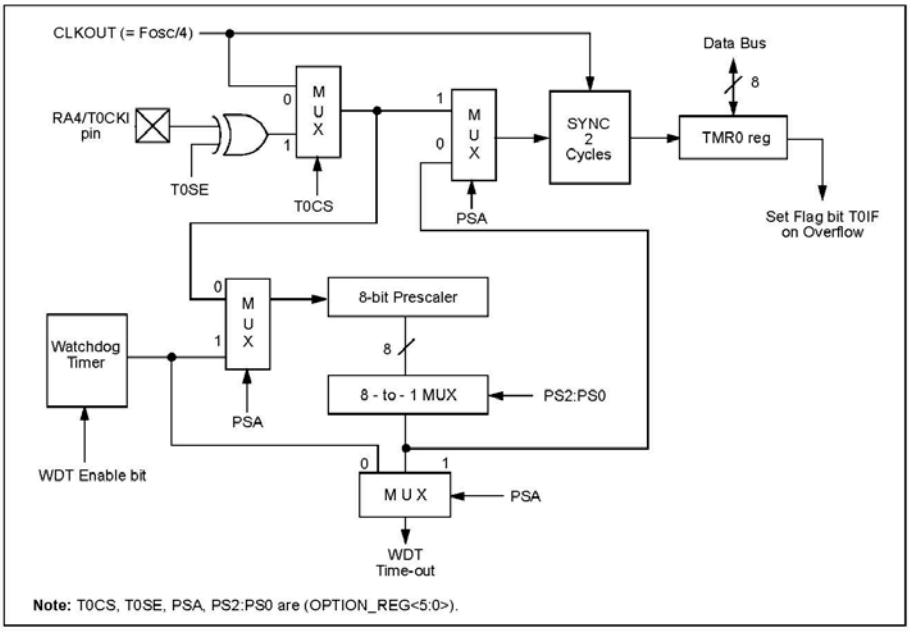
The prescaler assignment is fully under software control (i.e., it can be changed "on the fly" during program execution).

Note:	To avoid an unintended device RESET, a specific instruction sequence (shown in the PICmicro™ Mid-Range Reference Manual, DS33023) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

5.3 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP since the timer is shut-off during SLEEP.

FIGURE 5-2: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION_REG<5:0>).

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other RESETS
01h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	---1 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

6.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. The PIC16F84A has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These features are:

- OSC Selection
- RESET
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code Protection
- ID Locations
- In-Circuit Serial Programming™ (ICSP™)

The PIC16F84A has a Watchdog Timer which can be shut-off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. This design keeps the device in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external RESET circuitry.

SLEEP mode offers a very low current power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer Time-out or through an interrupt. Several oscillator options are provided to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select the various options.

Additional information on special features is available in the PICmicro™ Mid-Range Reference Manual (DS33023).

6.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped in program memory location 2007h.

Address 2007h is beyond the user program memory space and it belongs to the special test/configuration memory space (2000h - 3FFFh). This space can only be accessed during programming.

REGISTER 6-1: PIC16F84A CONFIGURATION WORD

R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	$\overline{\text{PWRTE}}$	WDTE	F0SC1	F0SC0
bit13													bit0

bit 13-4 **CP:** Code Protection bit
1 = Code protection disabled
0 = All program memory is code protected

bit 3 **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit
1 = Power-up Timer is disabled
0 = Power-up Timer is enabled

bit 2 **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0 **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

6.2 Oscillator Configurations

6.2.1 OSCILLATOR TYPES

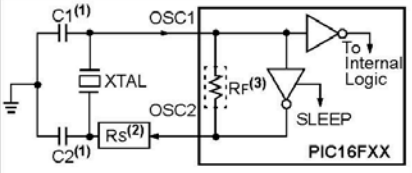
The PIC16F84A can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

6.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP, or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 6-1).

FIGURE 6-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



Note 1: See Table 6-1 for recommended values of C1 and C2.

2: A series resistor (RS) may be required for AT strip cut crystals.

The PIC16F84A oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP, or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 6-2).

FIGURE 6-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

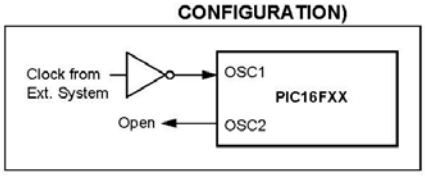


TABLE 6-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Ranges Tested:			
Mode	**Freq**	**OSC1/C1**	**OSC2/C2**
XT	455 kHz	47 - 100 pF	47 - 100 pF
	2.0 MHz	15 - 33 pF	15 - 33 pF
	4.0 MHz	15 - 33 pF	15 - 33 pF
HS	8.0 MHz	15 - 33 pF	15 - 33 pF
	10.0 MHz	15 - 33 pF	15 - 33 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested in this table. Higher capacitance increases the stability of the oscillator, but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for the appropriate values of external components.

Note: When using resonators with frequencies above 3.5 MHz, the use of HS mode rather than XT mode, is recommended. HS mode may be used at any VDD for which the controller is rated.

TABLE 6-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

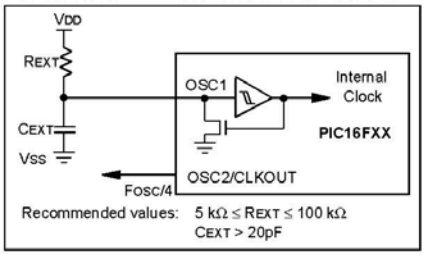
Mode	Freq	OSC1/C1	OSC2/C2
LP	32 kHz 200 kHz	68 - 100 pF 15 - 33 pF	68 - 100 pF 15 - 33 pF
XT	100 kHz 2 MHz 4 MHz	100 - 150 pF 15 - 33 pF 15 - 33 pF	100 - 150 pF 15 - 33 pF 15 - 33 pF
HS	4 MHz 20 MHz	15 - 33 pF 15 - 33 pF	15 - 33 pF 15 - 33 pF

Note: Higher capacitance increases the stability of the oscillator, but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.
For V_{DD} > 4.5V, C1 = C2 ≈ 30 pF is recommended.

6.2.3 RC OSCILLATOR

For timing insensitive applications, the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R_{EXT}) values, capacitor (C_{EXT}) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types also affects the oscillation frequency, especially for low C_{EXT} values. The user needs to take into account variation, due to tolerance of the external R and C components. Figure 6-3 shows how an R/C combination is connected to the PIC16F84A.

FIGURE 6-3: RC OSCILLATOR MODE



6.3 RESET

The PIC16F84A differentiates between various kinds of RESET:

- Power-on Reset (POR)
- $\overline{MCLR}$ during normal operation
- $\overline{MCLR}$ during SLEEP
- WDT Reset (during normal operation)
- WDT Wake-up (during SLEEP)

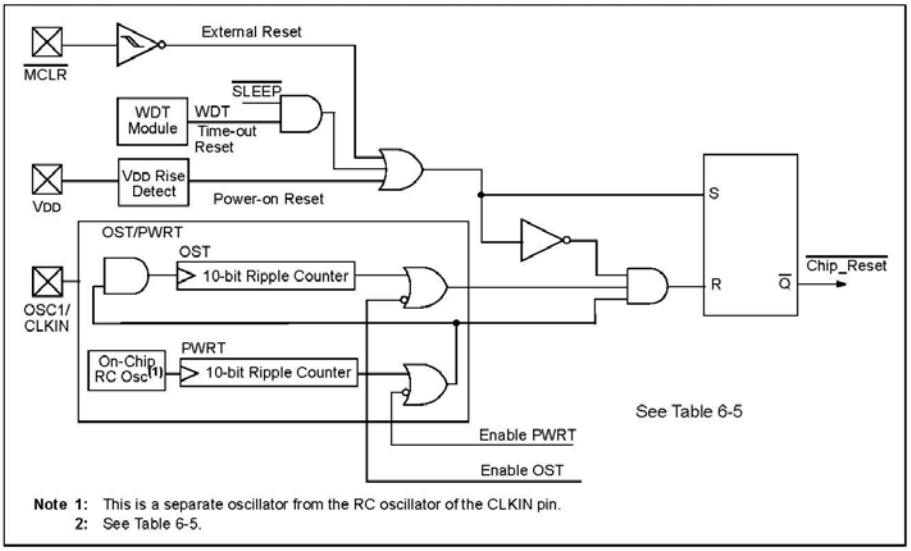
Figure 6-4 shows a simplified block diagram of the On-Chip RESET Circuit. The $\overline{MCLR}$ Reset path has a noise filter to ignore small pulses. The electrical specifications state the pulse width requirements for the $\overline{MCLR}$ pin.

Some registers are not affected in any RESET condition; their status is unknown on a POR and unchanged in any other RESET. Most other registers are reset to a "RESET state" on POR, $\overline{MCLR}$ or WDT Reset during normal operation and on $\overline{MCLR}$ during SLEEP. They are not affected by a WDT Reset during SLEEP, since this RESET is viewed as the resumption of normal operation.

Table 6-3 gives a description of RESET conditions for the program counter (PC) and the STATUS register. Table 6-4 gives a full description of RESET states for all registers.

The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different RESET situations (Section 6.7). These bits are used in software to determine the nature of the RESET.

FIGURE 6-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: This is a separate oscillator from the RC oscillator of the CLKIN pin.
2: See Table 6-5.

TABLE 6-3: RESET CONDITION FOR PROGRAM COUNTER AND THE STATUS REGISTER

Condition	Program Counter	STATUS Register
Power-on Reset	000h	0001 1xxx
$\overline{MCLR}$ during normal operation	000h	000u uuuu
$\overline{MCLR}$ during SLEEP	000h	0001 0uuu
WDT Reset (during normal operation)	000h	0000 1uuu
WDT Wake-up	PC + 1	uuu0 0uuu
Interrupt wake-up from SLEEP	PC + 1(1)	uuu1 0uuu

Legend: u = unchanged, x = unknown
Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

TABLE 6-4: RESET CONDITIONS FOR ALL REGISTERS

Register	Address	Power-on Reset	$\overline{MCLR}$ during: – normal operation – SLEEP WDT Reset during normal operation	Wake-up from SLEEP: – through interrupt – through WDT Time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	---- ----	---- ----	---- ----
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000 0000	0000 0000	PC + 1(2)
STATUS	03h	0001 1xxx	000q quuu(3)	uuuq quuu(3)
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA(4)	05h	---x xxxx	---u uuuu	---u uuuu
PORTB(5)	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu(1)
INDF	80h	---- ----	---- ----	---- ----
OPTION_REG	81h	1111 1111	1111 1111	uuuu uuuu
PCL	82h	0000 0000	0000 0000	PC + 1(2)
STATUS	83h	0001 1xxx	000q quuu(3)	uuuq quuu(3)
FSR	84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EECON1	88h	---0 x000	---0 q000	---0 uuuu
EECON2	89h	---- ----	---- ----	---- ----
PCLATH	8Ah	---0 0000	---0 0000	---u uuuu
INTCON	8Bh	0000 000x	0000 000u	uuuu uuuu(1)

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

Note 1: One or more bits in INTCON will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: Table 6-3 lists the RESET value for each specific condition.

4: On any device RESET, these pins are configured as inputs.

5: This is the value that will be in the port output latch.

6.4 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V - 1.7V). To take advantage of the POR, just tie the MCLR pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A minimum rise time for VDD must be met for this to operate properly. See Electrical Specifications for details.

When the device starts normal operation (exits the RESET condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in RESET until the operating conditions are met.

For additional information, refer to Application Note AN607, "*Power-up Trouble Shooting*."

The POR circuit does not produce an internal RESET when VDD declines.

6.5 Power-up Timer (PWRT)

The Power-up Timer (PWRT) provides a fixed 72 ms nominal time-out (TPWRT) from POR (Figures 6-6 through 6-9). The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level (possible exception shown in Figure 6-9).

A configuration bit, PWRTE, can enable/disable the PWRT. See Register 6-1 for the operation of the PWRTE bit for a particular device.

The power-up time delay TPWRT will vary from chip to chip due to VDD, temperature, and process variation. See DC parameters for details.

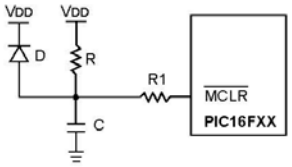
6.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle delay (from OSC1 input) after the PWRT delay ends (Figure 6-6, Figure 6-7, Figure 6-8 and Figure 6-9). This ensures the crystal oscillator or resonator has started and stabilized.

The OST time-out (TOST) is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

When VDD rises very slowly, it is possible that the TPWRT time-out and TOST time-out will expire before VDD has reached its final value. In this case (Figure 6-9), an external Power-on Reset circuit may be necessary (Figure 6-5).

FIGURE 6-5: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: External Power-on Reset circuit is required only if VDD power-up rate is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on MCLR pin is 5 μA). A larger voltage drop will degrade VIH level on the MCLR pin.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C, in the event of a MCLR pin breakdown due to ESD or EOS.

FIGURE 6-6: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

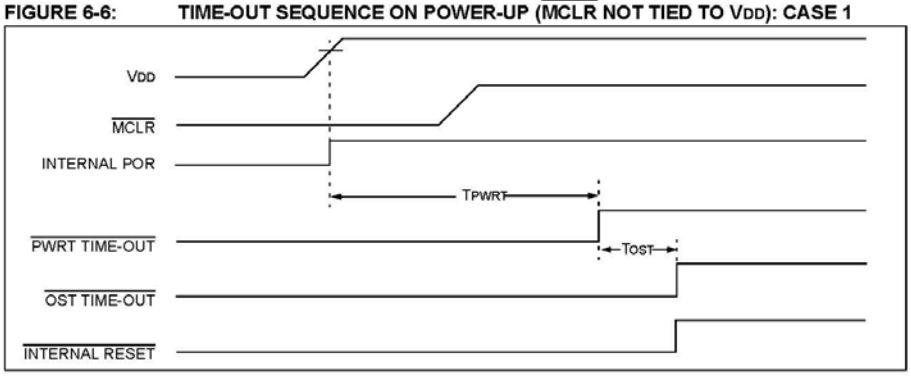


FIGURE 6-7: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

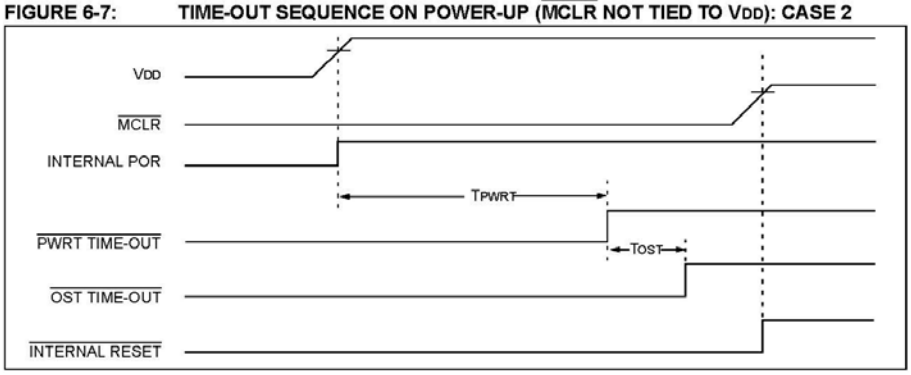


FIGURE 6-8: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD): FAST VDD RISE TIME

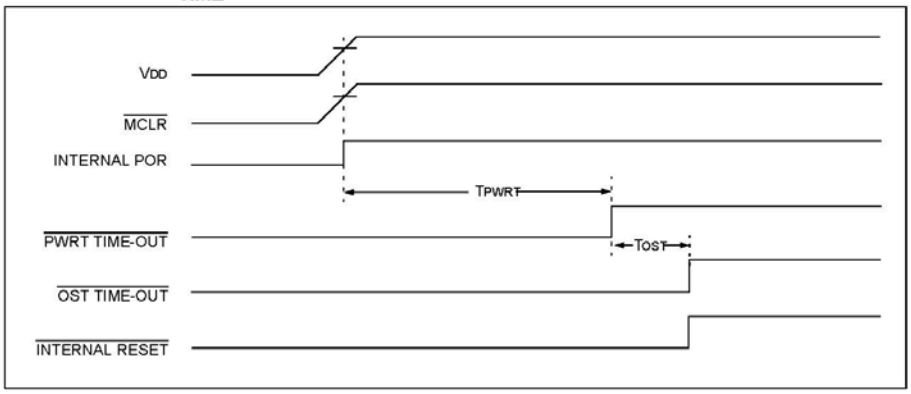
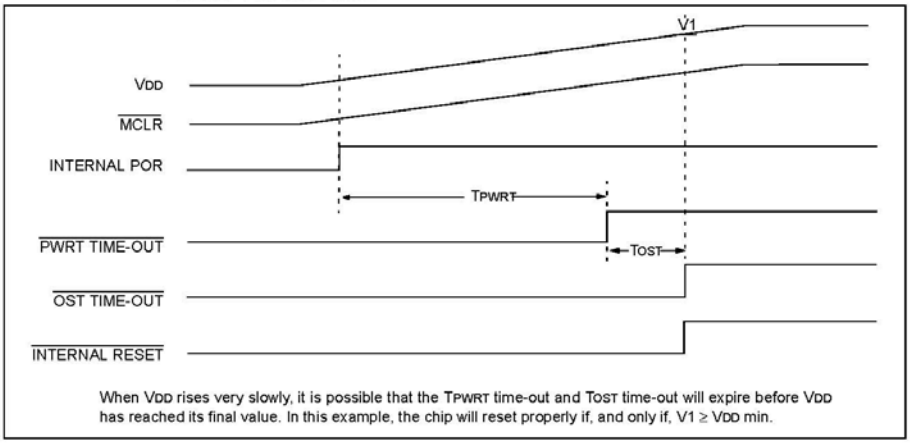


FIGURE 6-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD): SLOW VDD RISE TIME



When VDD rises very slowly, it is possible that the TPWRT time-out and TOST time-out will expire before VDD has reached its final value. In this example, the chip will reset properly if, and only if, V1 ≥ VDD min.

6.7 Time-out Sequence and Power-down Status Bits ($\overline{TO}$/$\overline{PD}$)

On power-up (Figures 6-6 through 6-9), the time-out sequence is as follows:

1. PWRT time-out is invoked after a POR has expired.
2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and PWRTE configuration bit status. For example, in RC mode with the PWRT disabled, there will be no time-out at all.

TABLE 6-5: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Wake-up from SLEEP
	PWRT Enabled	PWRT Disabled	
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	1024Tosc
RC	72 ms	—	—

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high, execution will begin immediately (Figure 6-6). This is useful for testing purposes or to synchronize more than one PIC16F84A device when operating in parallel.

Table 6-6 shows the significance of the $\overline{TO}$ and $\overline{PD}$ bits. Table 6-3 lists the RESET conditions for some special registers, while Table 6-4 lists the RESET conditions for all the registers.

TABLE 6-6: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{TO}$	$\overline{PD}$	Condition
1	1	Power-on Reset
0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
0	1	WDT Reset (during normal operation)
0	0	WDT Wake-up
1	1	$\overline{MCLR}$ during normal operation
1	0	$\overline{MCLR}$ during SLEEP or interrupt wake-up from SLEEP

6.8 Interrupts

The PIC16F84A has 4 sources of interrupt:

- External interrupt RB0/INT pin
- TMR0 overflow interrupt
- PORTB change interrupts (pins RB7:RB4)
- Data EEPROM write complete interrupt

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also contains the individual and global interrupt enable bits.

The global interrupt enable bit, GIE (INTCON<7>), enables (if set) all unmasked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. Bit GIE is cleared on RESET.

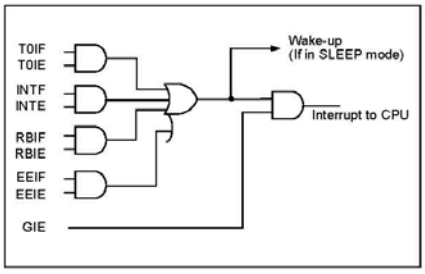
The "return from interrupt" instruction, RETFIE, exits interrupt routine as well as sets the GIE bit, which re-enables interrupts.

The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

When an interrupt is responded to, the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. For external interrupt events, such as the RB0/INT pin or PORTB change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency depends when the interrupt event occurs. The latency is the same for both one and two cycle instructions. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests.

Note:	Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

FIGURE 6-10: INTERRUPT LOGIC



6.8.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION_REG<6>) is set, or falling if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing control bit INTE (INTCON<4>). Flag bit INTF must be cleared in software via the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake the processor from SLEEP (Section 6.11) only if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether the processor branches to the interrupt vector following wake-up.

6.8.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in TMR0 will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>) (Section 5.0).

6.8.3 PORTB INTERRUPT

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<3>) (Section 4.2).

Note:	For a change on the I/O pin to be recognized, the pulse width must be at least TCY wide.

6.8.4 DATA EEPROM INTERRUPT

At the completion of a data EEPROM write cycle, flag bit EEIF (EECON1<4>) will be set. The interrupt can be enabled/disabled by setting/clearing enable bit EEIE (INTCON<6>) (Section 3.0).

6.9 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users wish to save key register values during an interrupt (e.g., W register and STATUS register). This is implemented in software.

The code in Example 6-1 stores and restores the STATUS and W register's values. The user defined registers, W_TEMP and STATUS_TEMP are the temporary storage locations for the W and STATUS registers values.

Example 6-1 does the following:

a) Stores the W register.
b) Stores the STATUS register in STATUS_TEMP.
c) Executes the Interrupt Service Routine code.
d) Restores the STATUS (and bank select bit) register.
e) Restores the W register.

EXAMPLE 6-1: SAVING STATUS AND W REGISTERS IN RAM

```
PUSH    MOVWF   W_TEMP              ; Copy W to TEMP register,
        SWAPF   STATUS,     W       ; Swap status to be saved into W
        MOVWF   STATUS_TEMP         ; Save status to STATUS_TEMP register
ISR     :                           :
        :                           ; Interrupt Service Routine
        :                           ; should configure Bank as required
        :                           ;
POP     SWAPF   STATUS_TEMP,W       ; Swap nibbles in STATUS_TEMP register
                                    ; and place result into W
        MOVWF   STATUS              ; Move W into STATUS register
                                    ; (sets bank to original state)
        SWAPF   W_TEMP,     F       ; Swap nibbles in W_TEMP and place result in W_TEMP
        SWAPF   W_TEMP,     W       ; Swap nibbles in W_TEMP and place result into W
```

6.10 Watchdog Timer (WDT)

The Watchdog Timer is a free running On-Chip RC Oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT wake-up causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming configuration bit WDTE as a '0' (Section 6.1).

6.10.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION_REG register. Thus, time-out periods up to 2.3 seconds can be realized.

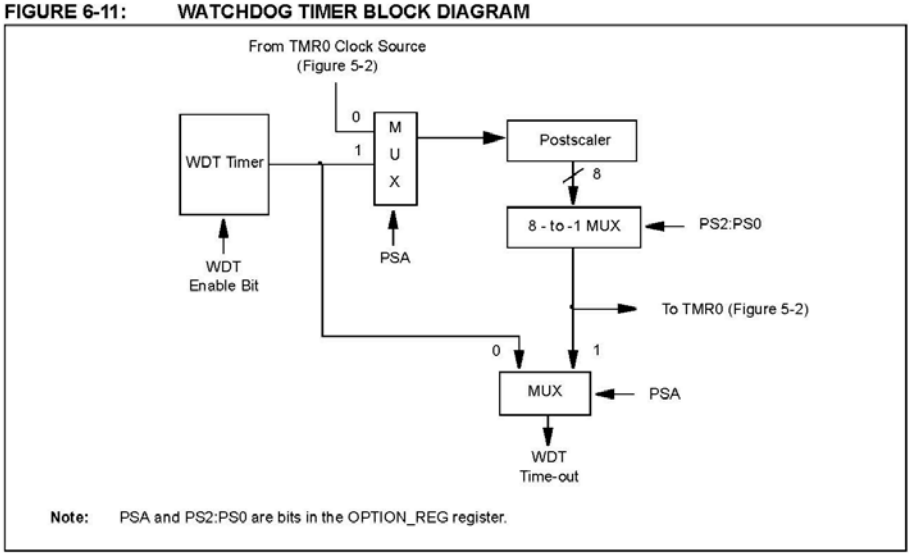
The CLRWDT and SLEEP instructions clear the WDT and the postscaler (if assigned to the WDT) and prevent it from timing out and generating a device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a WDT time-out.

6.10.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions (V_{DD} = Min., Temperature = Max., Max. WDT Prescaler), it may take several seconds before a WDT time-out occurs.

FIGURE 6-11: WATCHDOG TIMER BLOCK DIAGRAM



Note: PSA and PS2:PS0 are bits in the OPTION_REG register.

TABLE 6-7: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other RESETS
2007h	Config. bits	(2)	(2)	(2)	(2)	$\overline{PWRTE}$(1)	WDTE	FOSC1	FOSC0	(2)	
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown. Shaded cells are not used by the WDT.

Note 1: See Register 6-1 for operation of the $\overline{PWRTE}$ bit.

2: See Register 6-1 and Section 6.12 for operation of the code and data protection bits.

6.11 Power-down Mode (SLEEP)

A device may be powered down (SLEEP) and later powered up (wake-up from SLEEP).

6.11.1 SLEEP

The Power-down mode is entered by executing the SLEEP instruction.

If enabled, the Watchdog Timer is cleared (but keeps running), the $\overline{PD}$ bit (STATUS<3>) is cleared, the $\overline{TO}$ bit (STATUS<4>) is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, low, or hi-impedance).

For the lowest current consumption in SLEEP mode, place all I/O pins at either VDD or VSS, with no external circuitry drawing current from the I/O pins, and disable external clocks. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS. The contribution from on-chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (VIHMC).

It should be noted that a $\overline{RESET}$ generated by a WDT time-out does not drive the $\overline{MCLR}$ pin low.

6.11.2 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

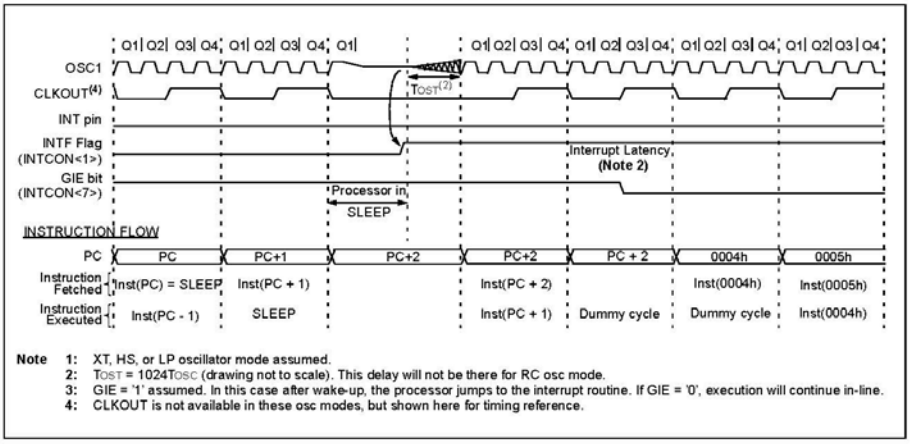
1. External RESET input on $\overline{MCLR}$ pin.
2. WDT wake-up (if WDT was enabled).
3. Interrupt from RB0/INT pin, RB port change, or data EEPROM write complete.

Peripherals cannot generate interrupts during SLEEP, since no on-chip Q clocks are present.

The first event ($\overline{MCLR}$ Reset) will cause a device RESET. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits can be used to determine the cause of a device RESET. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up).

While the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

FIGURE 6-12: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT, HS, or LP oscillator mode assumed.
2: TOST = 1024TOSC (drawing not to scale). This delay will not be there for RC osc mode.
3: GIE = '1' assumed. In this case after wake-up, the processor jumps to the interrupt routine. If GIE = '0', execution will continue in-line.
4: CLKOUT is not available in these osc modes, but shown here for timing reference.

6.11.3 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake-up from SLEEP. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

6.12 Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

6.13 ID Locations

Four memory locations (2000h - 2004h) are designated as ID locations to store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable only during program/verify. Only the four Least Significant bits of ID location are usable.

6.14 In-Circuit Serial Programming

PIC16F84A microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. Customers can manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product, allowing the most recent firmware or custom firmware to be programmed.

For complete details of Serial Programming, please refer to the In-Circuit Serial Programming™ (ICSP™) Guide, (DS30277).

NOTES:

7.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word, divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 7-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 7-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 7-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Table 7-2 lists the instructions recognized by the MPASM™ Assembler.

Figure 7-1 shows the general formats that the instructions can have.

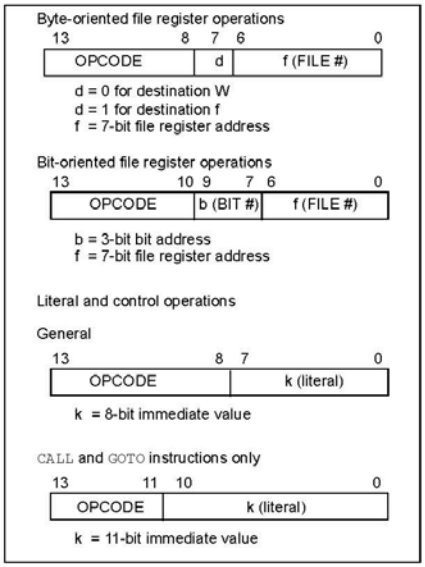
Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 7-1: GENERAL FORMAT FOR INSTRUCTIONS



A description of each instruction is available in the PICmicro™ Mid-Range Reference Manual (DS33023).

TABLE 7-2: PIC16CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb	LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS							
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff		C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff		Z	1,2
CLRF	f	Clear f	1	00 0001 lfff ffff		Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx		Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff		Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff		Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1 (2)	00 1011 dfff ffff			1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff		Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1 (2)	00 1111 dfff ffff			1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff		Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff		Z	1,2
MOVWF	f	Move W to f	1	00 0000 lfff ffff			
NOP	-	No Operation	1	00 0000 0xx0 0000			
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff		C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff		C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff		C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff			1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff		Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS							
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff			1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff			1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff			3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff			3
LITERAL AND CONTROL OPERATIONS							
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk		C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk		Z	
CALL	k	Call subroutine	2	10 0kkk kkkk kkkk			
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100		$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk			
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk		Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk			
RETFIE	-	Return from interrupt	2	00 0000 0000 1001			
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk			
RETURN	-	Return from Subroutine	2	00 0000 0000 1000			
SLEEP	-	Go into standby mode	1	00 0000 0110 0011		$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk		C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk		Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

7.1 Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

ANDLW	AND Literal with W
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2TCY instruction.

BTFSC	Bit Test, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b' in register 'f' is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2TCY instruction.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\bar{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead, making it a 2TCY instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2TCY instruction.

IORLW	Inclusive OR Literal with W
Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) → (destination)
Status Affected:	Z
Description:	The contents of register f are moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register, since status flag Z is affected.

MOVLW	Move Literal to W
Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	0 ≤ f ≤ 127
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

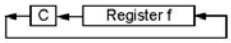
RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

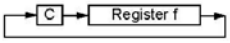
RLF — **Rotate Left f through Carry**

Syntax:	[*label*] RLF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.



RRF — **Rotate Right f through Carry**

Syntax:	[*label*] RRF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



SLEEP

Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.

SUBLW — **Subtract W from Literal**

Syntax:	[*label*] SUBLW k
Operands:	0 ≤ k ≤ 255
Operation:	k - (W) → (W)
Status Affected:	C, DC, Z
Description:	The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

SUBWF — **Subtract W from f**

Syntax:	[*label*] SUBWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - (W) → (destination)
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

SWAPF — **Swap Nibbles in f**

Syntax:	[*label*] SWAPF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.

XORLW	Exclusive OR Literal with W
Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

8.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - ICEPIC™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Entry-Level Development Programmer
- Low Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM 2 Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 17 Demonstration Board
  - KEELOQ® Demonstration Board

8.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. The MPLAB IDE is a Windows®-based application that contains:

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
  - in-circuit debugger (sold separately)
- A full-featured editor
- A project manager
- Customizable toolbar and key mapping
- A status bar
- On-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PICmicro emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - machine code

The ability to use MPLAB IDE with multiple debugging tools allows users to easily switch from the cost-effective simulator to a full-featured emulator with minimal retraining.

8.2 MPASM Assembler

The MPASM assembler is a full-featured universal macro assembler for all PICmicro MCU's.

The MPASM assembler has a command line interface and a Windows shell. It can be used as a stand-alone application on a Windows 3.x or greater system, or it can be used through MPLAB IDE. The MPASM assembler generates relocatable object files for the MPLINK object linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file that contains source lines and generated machine code, and a COD file for debugging.

The MPASM assembler features include:

- Integration into MPLAB IDE projects.
- User-defined macros to streamline assembly code.
- Conditional assembly for multi-purpose source files.
- Directives that allow complete control over the assembly process.

8.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI 'C' compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

8.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

8.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

8.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows® environment were chosen to best make these features available to you, the end user.

8.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

8.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PICmicro MCUs and can be used to develop for this and other PICmicro microcontrollers. The MPLAB ICD utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming™ protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

8.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable VDD and VPP supplies, which allow it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode, the PRO MATE II device programmer can read, verify, or program PICmicro devices. It can also set code protection in this mode.

8.10 PICSTART Plus Entry Level Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

The PICSTART Plus development programmer supports all PICmicro devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

8.11 PICDEM 1 Low Cost PICmicro Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

8.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I2CTM bus and separate headers for connection to an LCD module and a keypad.

8.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with an LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 3 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

8.14 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included and the user may erase it and program it with the other sample programs using the PRO MATE II device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the PICMASTER emulator and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

8.15 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchip's HCS Secure Data Products. The HCS evaluation kit includes a LCD display to show changing codes, a decoder to decode transmissions and a programming interface to program test transmitters.

TABLE 8-1: DEVELOPMENT TOOLS FROM MICROCHIP

		PIC12CXXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	PIC18FXXX	24CXX/ 25CXX/ 93CXX	HCSXXX	MCRFXXX	MCP2510
Software Tools	MPLAB® Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	MPLAB® C17 C Compiler												✓	✓						
	MPLAB® C18 C Compiler														✓	✓				
	MPASM™ Assembler/ MPLINK™ Object Linker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Emulators	MPLAB® ICE In-Circuit Emulator	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	ICEPIC™ In-Circuit Emulator	✓		✓	✓	✓		✓	✓	✓		✓								
Debugger	MPLAB® ICD In-Circuit Debugger				✓*			✓*			✓					✓				
Programmers	PICSTART® Plus Entry Level Development Programmer	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	PRO MATE® II Universal Device Programmer	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Demo Boards and Eval Kits	PICDEM™ 1 Demonstration Board			✓		✓		✓†		✓			✓							
	PICDEM™ 2 Demonstration Board				✓†			✓†							✓	✓				
	PICDEM™ 3 Demonstration Board											✓								
	PICDEM™ 14A Demonstration Board		✓																	
	PICDEM™ 17 Demonstration Board													✓						
	KEELOQ® Evaluation Kit																	✓		
	KEELOQ® Transponder Kit																	✓		
	microID™ Programmer's Kit																		✓	
	125 kHz microID™ Developer's Kit																✓		✓	
	125 kHz Anticollision microID™ Developer's Kit																✓		✓	
	13.56 MHz Anticollision microID™ Developer's Kit																✓		✓	
	MCP2510 CAN Developer's Kit																			✓

* Contact the Microchip Technology Inc. web site at www.microchip.com for information on how to use the MPLAB® ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77.

** Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

NOTES:

9.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on any pin with respect to VSS (except VDD, $\overline{MCLR}$, and RA4) ..... -0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS ..... -0.3 to +7.5V
Voltage on $\overline{MCLR}$ with respect to VSS(1) ..... -0.3 to +14V
Voltage on RA4 with respect to VSS ..... -0.3 to +8.5V
Total power dissipation(2) ..... 800 mW
Maximum current out of VSS pin ..... 150 mA
Maximum current into VDD pin ..... 100 mA
Input clamp current, IIK (VI < 0 or VI > VDD) ..... ± 20 mA
Output clamp current, IOK (VO < 0 or VO > VDD) ..... ± 20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum current sunk by PORTA ..... 80 mA
Maximum current sourced by PORTA ..... 50 mA
Maximum current sunk by PORTB ..... 150 mA
Maximum current sourced by PORTB ..... 100 mA

Note 1: Voltage spikes below VSS at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to VSS.

2: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

FIGURE 9-1: PIC16F84A-20 VOLTAGE-FREQUENCY GRAPH

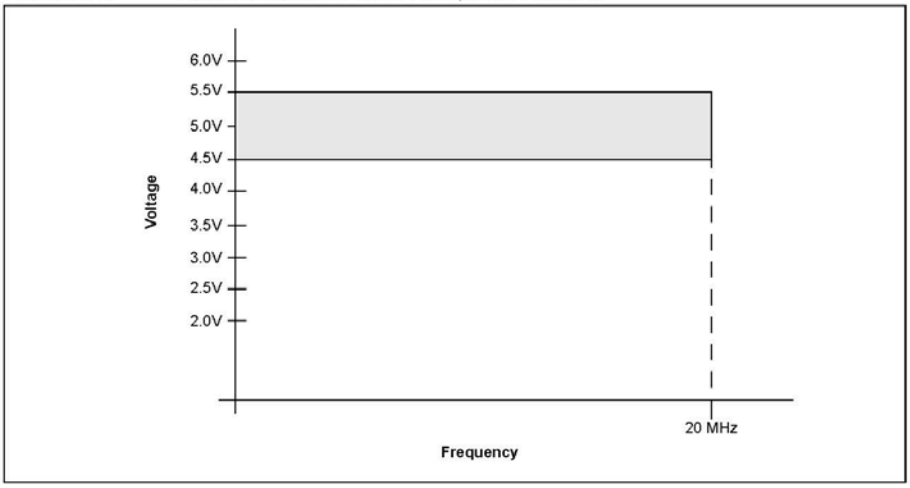
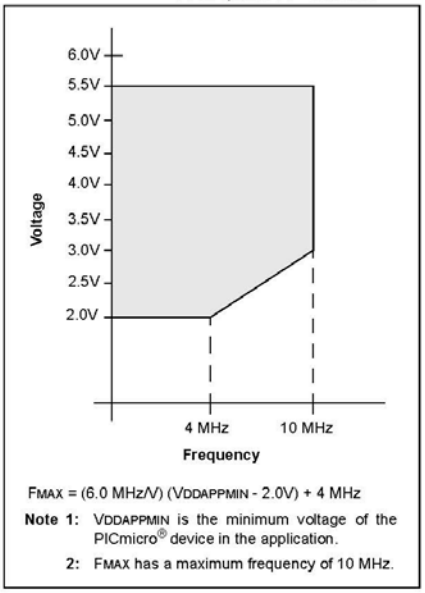


FIGURE 9-2: PIC16LF84A-04 VOLTAGE-FREQUENCY GRAPH

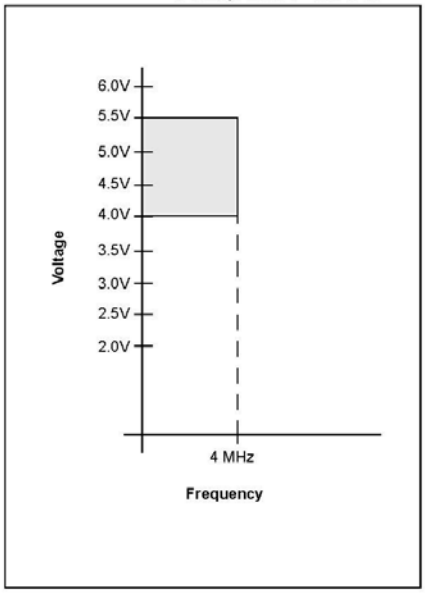


FMAX = (6.0 MHz/V) (VDDAPPMIN - 2.0V) + 4 MHz

Note 1: VDDAPPMIN is the minimum voltage of the PICmicro® device in the application.

2: FMAX has a maximum frequency of 10 MHz.

FIGURE 9-3: PIC16F84A-04 VOLTAGE-FREQUENCY GRAPH



9.1 DC Characteristics

PIC16LF84A-04 (Commercial, Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)
PIC16F84A-04 (Commercial, Industrial, Extended) **PIC16F84A-20** (Commercial, Industrial, Extended)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
	VDD	**Supply Voltage**					
D001		16LF84A	2.0	—	5.5	V	XT, RC, and LP osc configuration
D001		16F84A	4.0	—	5.5	V	XT, RC and LP osc configuration
D001A			4.5	—	5.5	V	HS osc configuration
D002	VDR	**RAM Data Retention Voltage (Note 1)**	1.5	—	—	V	Device in SLEEP mode
D003	VPOR	**VDD Start Voltage** to ensure internal Power-on Reset signal	—	Vss	—	V	See section on Power-on Reset for details
D004	SVDD	**VDD Rise Rate** to ensure internal Power-on Reset signal	0.05	—	—	V/ms	
	IDD	**Supply Current (Note 2)**					
D010		16LF84A	—	1	4	mA	RC and XT osc configuration **(Note 4)** FOSC = 2.0 MHz, VDD = 5.5V
D010		16F84A	—	1.8	4.5	mA	RC and XT osc configuration **(Note 4)** FOSC = 4.0 MHz, VDD = 5.5V
D010A			—	3	10	mA	RC and XT osc configuration **(Note 4)** FOSC = 4.0 MHz, VDD = 5.5V (During FLASH programming)
D013			—	10	20	mA	HS osc configuration (PIC16F84A-20) FOSC = 20 MHz, VDD = 5.5V
D014		16LF84A	—	15	45	µA	LP osc configuration FOSC = 32 kHz, VDD = 2.0V, WDT disabled

Legend: Rows with standard voltage device data only are shaded for improved readability.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

NR Not rated for operation.

Note 1: This is the limit to which VDD can be lowered without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD,
T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula IR = VDD/2REXT (mA) with REXT in kOhm.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD measurement.

9.1 DC Characteristics (Continued)

PIC16LF84A-04 (Commercial, Industrial)			**Standard Operating Conditions (unless otherwise stated)** Operating temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)				
PIC16F84A-04 (Commercial, Industrial, Extended) **PIC16F84A-20** (Commercial, Industrial, Extended)			**Standard Operating Conditions (unless otherwise stated)** Operating temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	IPD	**Power-down Current (Note 3)**					
D020		16LF84A					
D020		16F84A-20 16F84A-04					
D021A		16LF84A	—	0.4	1.0	µA	VDD = 2.0V, WDT disabled, industrial
D021A		16F84A-20	—	1.5	3.5	µA	VDD = 4.5V, WDT disabled, industrial
		16F84A-04	—	1.0	3.0	µA	VDD = 4.0V, WDT disabled, industrial
D021B		16F84A-20	—	1.5	5.5	µA	VDD = 4.5V, WDT disabled, extended
		16F84A-04	—	1.0	5.0	µA	VDD = 4.0V, WDT disabled, extended
		Module Differential Current (Note 5)					
D022	ΔIWDT	Watchdog Timer	—	.20	16	µA	VDD = 2.0V, Industrial, Commercial
			—	3.5	20	µA	VDD = 4.0V, Commercial
			—	3.5	28	µA	VDD = 4.0V, Industrial, Extended
			—	4.8	25	µA	VDD = 4.5V, Commercial
			—	4.8	30	µA	VDD = 4.5V, Industrial, Extended

Legend: Rows with standard voltage device data only are shaded for improved readability.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

NR Not rated for operation.

Note 1: This is the limit to which VDD can be lowered without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD,
T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula IR = VDD/2REXT (mA) with REXT in kOhm.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD measurement.

9.2 DC Characteristics: PIC16F84A-04 (Commercial, Industrial) PIC16F84A-20 (Commercial, Industrial) PIC16LF84A-04 (Commercial, Industrial)

DC Characteristics All Pins Except Power Supply Pins			Standard Operating Conditions (unless otherwise stated) Operating temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial) Operating voltage VDD range as described in DC specifications (Section 9.1)				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O ports:					
D030		with TTL buffer	VSS	—	0.8	V	4.5V ≤ VDD ≤ 5.5V **(Note 4)**
D030A			VSS	—	0.16VDD	V	Entire range **(Note 4)**
D031		with Schmitt Trigger buffer	VSS	—	0.2VDD	V	Entire range
D032		$\overline{MCLR}$, RA4/T0CKI	VSS	—	0.2VDD	V	
D033		OSC1 (XT, HS and LP modes)	VSS	—	0.3VDD	V	**(Note 1)**
D034		OSC1 (RC mode)	VSS	—	0.1VDD	V	
	VIH	**Input High Voltage**					
		I/O ports:		—			
D040		with TTL buffer	2.0	—	VDD	V	4.5V ≤ VDD ≤ 5.5V **(Note 4)**
D040A			0.25VDD+0.8	—	VDD	V	Entire range **(Note 4)**
D041		with Schmitt Trigger buffer	0.8 VDD	—	VDD		Entire range
D042		$\overline{MCLR}$,	0.8 VDD	—	VDD	V	
D042A		RA4/T0CKI	0.8 VDD	—	8.5	V	
D043		OSC1 (XT, HS and LP modes)	0.8 VDD	—	VDD	V	**(Note 1)**
D043A		OSC1 (RC mode)	0.9 VDD		VDD	V	
D050	VHYS	**Hysteresis of Schmitt Trigger Inputs**	—	0.1	—	V	
D070	IPURB	**PORTB Weak Pull-up Current**	50	250	400	µA	VDD = 5.0V, VPIN = VSS
	IIL	**Input Leakage Current (Notes 2, 3)**					
D060		I/O ports	—	—	±1	µA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance
D061		$\overline{MCLR}$, RA4/T0CKI	—	—	±5	µA	VSS ≤ VPIN ≤ VDD
D063		OSC1	—	—	±5	µA	VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. Do not drive the PIC16F84A with an external clock while the device is in RC mode, or chip damage may result.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: The user may choose the better of the two specs.

9.2 DC Characteristics: PIC16F84A-04 (Commercial, Industrial) PIC16F84A-20 (Commercial, Industrial) PIC16LF84A-04 (Commercial, Industrial) (Continued)

DC Characteristics All Pins Except Power Supply Pins			Standard Operating Conditions (unless otherwise stated) Operating temperature 0°C ≤ TA ≤ +70°C (commercial) -40°C ≤ TA ≤ +85°C (industrial) Operating voltage VDD range as described in DC specifications (Section 9.1)				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VOL	**Output Low Voltage**					
D080		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V
D083		OSC2/CLKOUT	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V, (RC mode only)
	VOH	**Output High Voltage**					
D090		I/O ports **(Note 3)**	VDD-0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V
D092		OSC2/CLKOUT **(Note 3)**	VDD-0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V (RC mode only)
	VOD	**Open Drain High Voltage**					
D150		RA4 pin	—	—	8.5	V	
		Capacitive Loading Specs on Output Pins					
D100	COSC2	OSC2 pin	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101	CIO	All I/O pins and OSC2 (RC mode)	—	—	50	pF	
		Data EEPROM Memory					
D120	ED	Endurance	1M	10M	—	E/W	25°C at 5V
D121	VDRW	VDD for read/write	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D122	TDEW	Erase/Write cycle time	—	4	8	ms	
		Program FLASH Memory					
D130	EP	Endurance	1000	10K	—	E/W	
D131	VPR	VDD for read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VPEW	VDD for erase/write	4.5	—	5.5	V	
D133	TPEW	Erase/Write cycle time	—	4	8	ms	

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. Do not drive the PIC16F84A with an external clock while the device is in RC mode, or chip damage may result.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: The user may choose the better of the two specs.

9.3 AC (Timing) Characteristics

9.3.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

T				
	F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp				
	2	to	os, osc	OSC1
	ck	CLKOUT	ost	oscillator start-up timer
	cy	cycle time	pwrt	power-up timer
	io	I/O port	rbt	RBx pins
	inp	INT pin	t0	T0CKI
	mp	$\overline{\text{MCLR}}$	wdt	watchdog timer

Uppercase letters and their meanings:

S				
	F	Fall	P	Period
	H	High	R	Rise
	I	Invalid (high impedance)	V	Valid
	L	Low	Z	High Impedance

9.3.2 TIMING CONDITIONS

The temperature and voltages specified in Table 9-1 apply to all timing specifications unless otherwise noted. All timings are measured between high and low measurement points as indicated in Figure 9-4. Figure 9-5 specifies the load conditions for the timing specifications.

TABLE 9-1: TEMPERATURE AND VOLTAGE SPECIFICATIONS - AC

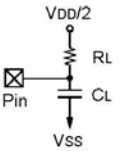
AC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature 0°C ≤ TA ≤ +70°C for commercial -40°C ≤ TA ≤ +85°C for industrial Operating voltage VDD range as described in DC specifications (Section 9.1)

FIGURE 9-4: PARAMETER MEASUREMENT INFORMATION

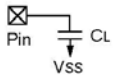
OSC1 Measurement Points I/O Port Measurement Points

FIGURE 9-5: LOAD CONDITIONS

Load Condition 1



Load Condition 2



RL = 464Ω

CL = 50 pF for all pins except OSC2
15 pF for OSC2 output

9.3.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 9-6: EXTERNAL CLOCK TIMING

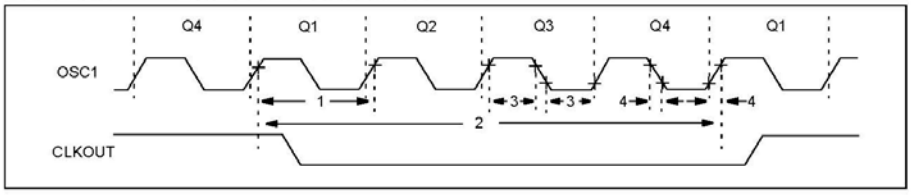


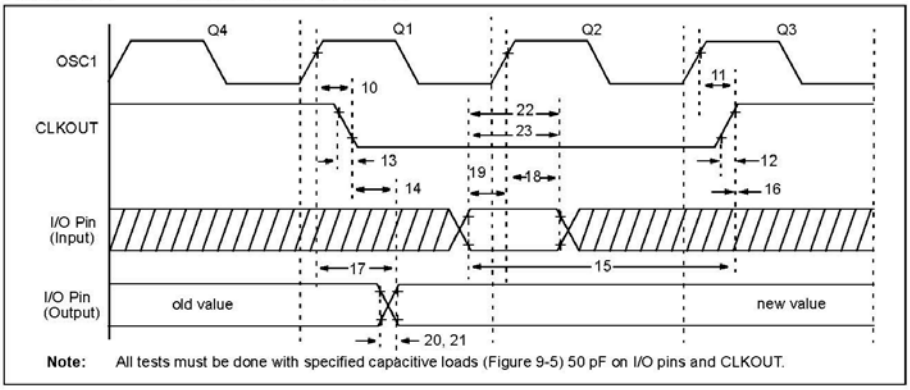
TABLE 9-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
	FOSC	**External CLKIN Frequency(1)**	DC	—	2	MHz	XT, RC osc	(-04, LF)
			DC	—	4	MHz	XT, RC osc	(-04)
			DC	—	20	MHz	HS osc	(-20)
			DC	—	200	kHz	LP osc	(-04, LF)
		Oscillator Frequency(1)	DC	—	2	MHz	RC osc	(-04, LF)
			DC	—	4	MHz	RC osc	(-04)
			0.1	—	2	MHz	XT osc	(-04, LF)
			0.1	—	4	MHz	XT osc	(-04)
			1.0	—	20	MHz	HS osc	(-20)
			DC	—	200	kHz	LP osc	(-04, LF)
1	TOSC	**External CLKIN Period(1)**	500	—	—	ns	XT, RC osc	(-04, LF)
			250	—	—	ns	XT, RC osc	(-04)
			50	—	—	ns	HS osc	(-20)
			5.0	—	—	µs	LP osc	(-04, LF)
		Oscillator Period(1)	500	—	—	ns	RC osc	(-04, LF)
			250	—	—	ns	RC osc	(-04)
			500	—	10,000	ns	XT osc	(-04, LF)
			250	—	10,000	ns	XT osc	(-04)
			50	—	1,000	ns	HS osc	(-20)
			5.0	—	—	µs	LP osc	(-04, LF)
2	TCY	**Instruction Cycle Time(1)**	0.2	4/FOSC	DC	µs		
3	TOSL, TOSH	**Clock in (OSC1) High or Low Time**	60	—	—	ns	XT osc	(-04, LF)
			50	—	—	ns	XT osc	(-04)
			2.0	—	—	µs	LP osc	(-04, LF)
			17.5	—	—	ns	HS osc	(-20)
4	TOSR, TOSF	**Clock in (OSC1) Rise or Fall Time**	25	—	—	ns	XT osc	(-04)
			50	—	—	ns	LP osc	(-04, LF)
			7.5	—	—	ns	HS osc	(-20)

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "Min." values with an external clock applied to the OSC1 pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 9-7: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (Figure 9-5) 50 pF on I/O pins and CLKOUT.

TABLE 9-3: CLKOUT AND I/O TIMING REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	Standard	—	15	30	ns	**(Note 1)**
10A			Extended (LF)	—	15	120	ns	**(Note 1)**
11	TosH2ckH	OSC1↑ to CLKOUT↑	Standard	—	15	30	ns	**(Note 1)**
11A			Extended (LF)	—	15	120	ns	**(Note 1)**
12	TckR	CLKOUT rise time	Standard	—	15	30	ns	**(Note 1)**
12A			Extended (LF)	—	15	100	ns	**(Note 1)**
13	TckF	CLKOUT fall time	Standard	—	15	30	ns	**(Note 1)**
13A			Extended (LF)	—	15	100	ns	**(Note 1)**
14	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	0.5TCY +20	ns	**(Note 1)**
15	TioV2ckH	Port in valid before CLKOUT ↑	Standard	0.30TCY + 30	—	—	ns	**(Note 1)**
			Extended (LF)	0.30TCY + 80	—	—	ns	**(Note 1)**
16	TckH2iol	Port in hold after CLKOUT ↑		0	—	—	ns	**(Note 1)**
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	Standard	—	—	125	ns	
			Extended (LF)	—	—	250	ns	
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	Standard	10	—	—	ns	
			Extended (LF)	10	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	Standard	-75	—	—	ns	
			Extended (LF)	-175	—	—	ns	
20	TioR	Port output rise time	Standard	—	10	35	ns	
20A			Extended (LF)	—	10	70	ns	
21	TioF	Port output fall time	Standard	—	10	35	ns	
21A			Extended (LF)	—	10	70	ns	
22	TINP	INT pin high or low time	Standard	20	—	—	ns	
22A			Extended (LF)	55	—	—	ns	
23	TRBP	RB7:RB4 change INT high or low time	Standard	TOSC§	—	—	ns	
23A			Extended (LF)	TOSC§	—	—	ns	

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.
§ By design.
Note 1: Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

FIGURE 9-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

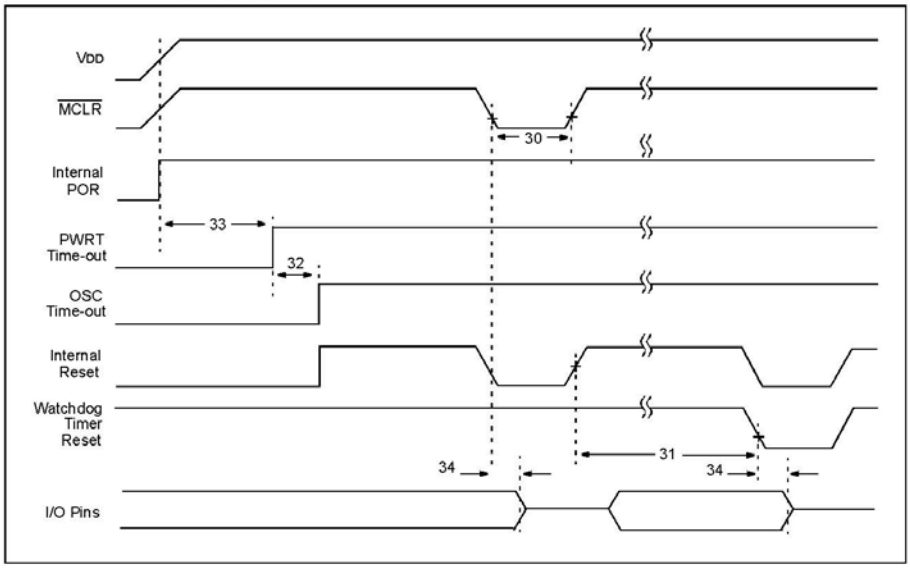


TABLE 9-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	2	—	—	µs	VDD = 5.0V
31	TWDT	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5.0V
32	TOST	Oscillation Start-up Timer Period		1024Tosc		ms	Tosc = OSC1 period
33	TPWRT	Power-up Timer Period	28	72	132	ms	VDD = 5.0V
34	TIOZ	I/O hi-impedance from $\overline{MCLR}$ Low or RESET	—	—	100	ns	

† Data in "Typ" column is at 5V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 9-9: TIMER0 CLOCK TIMINGS

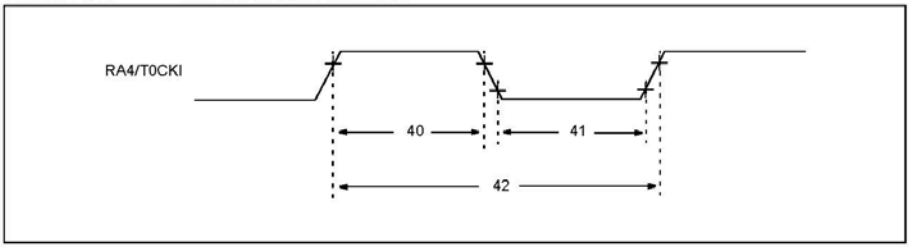


TABLE 9-5: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	50	—	—	ns	2.0V ≤ VDD ≤ 3.0V
				30	—	—	ns	3.0V ≤ VDD ≤ 6.0V
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	50	—	—	ns	2.0V ≤ VDD ≤ 3.0V
				20	—	—	ns	3.0V ≤ VDD ≤ 6.0V
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)

† Data in "Typ" column is at 5.0V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

10.0 DC/AC CHARACTERISTIC GRAPHS

The graphs provided in this section are for **design guidance** and are **not tested**.

In some graphs, the data presented are **outside specified operating range** (i.e., outside specified VDD range). This is for **information only** and devices are ensured to operate properly only within the specified range.

The data presented in this section is a **statistical summary** of data collected on units from different lots over a period of time and matrix samples. 'Typical' represents the mean of the distribution at 25°C. 'Max' or 'Min' represents (mean + 3σ) or (mean - 3σ), respectively, where σ is a standard deviation over the whole temperature range.

FIGURE 10-1: TYPICAL IDD vs. FOSC OVER VDD (HS MODE, 25°C)

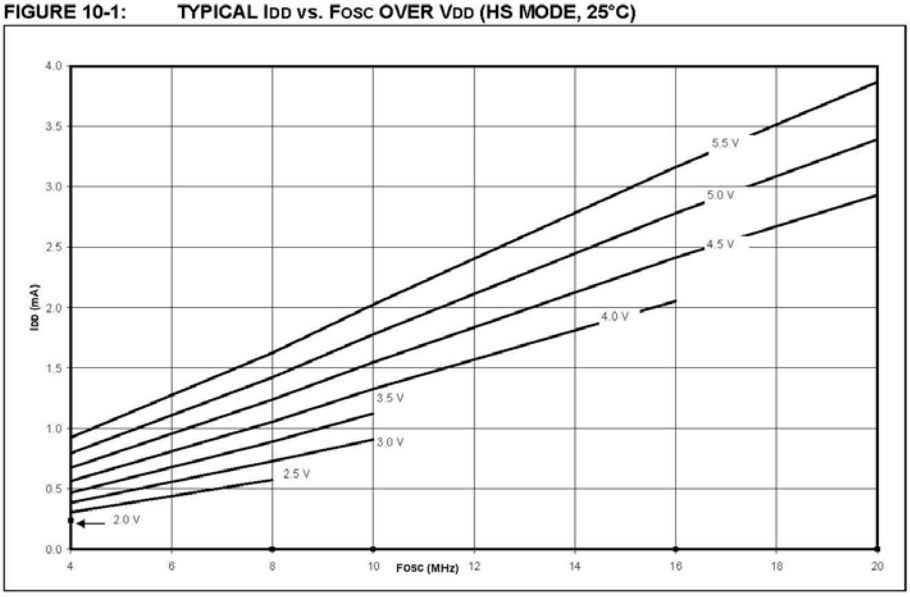


FIGURE 10-2: MAXIMUM IDD vs. FOSC OVER VDD (HS MODE, -40° TO +125°C)

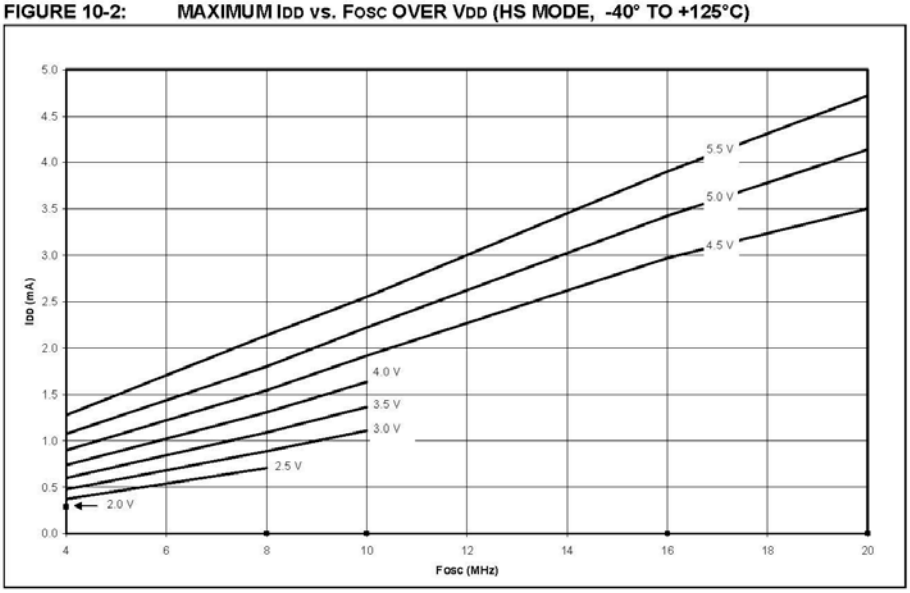


FIGURE 10-3: TYPICAL IDD vs. FOSC OVER VDD (XT MODE, 25°C)

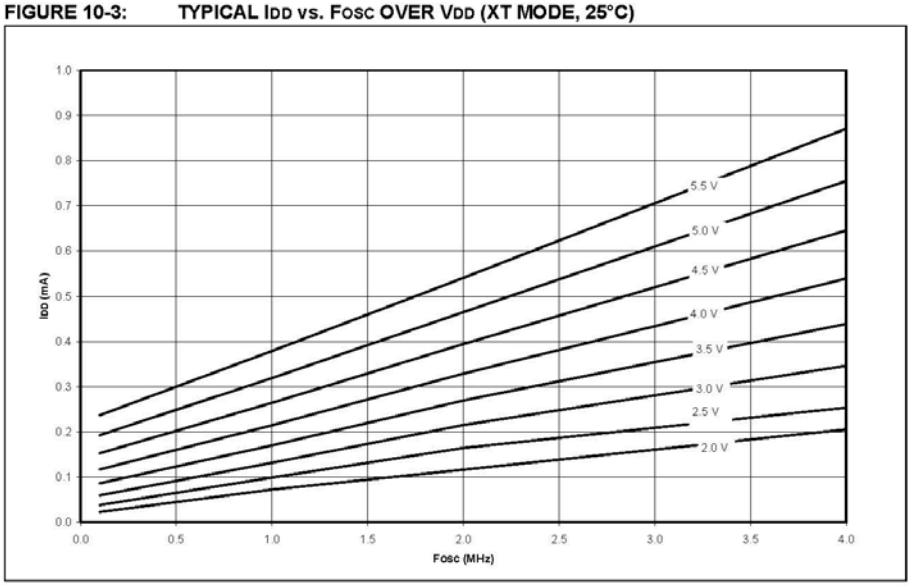


FIGURE 10-4: MAXIMUM IDD vs. FOSC OVER VDD (XT MODE, -40° TO +125°C)

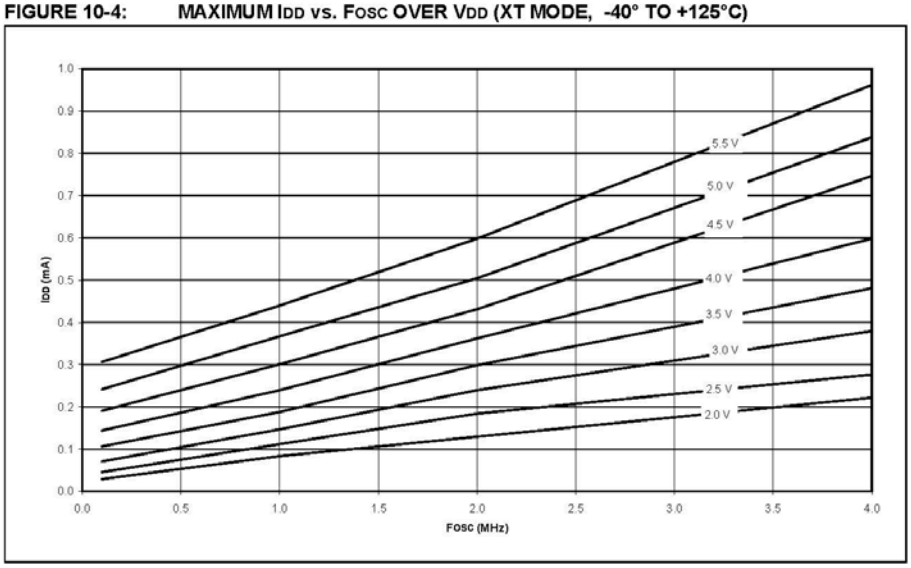


FIGURE 10-5: TYPICAL IDD vs. FOSC OVER VDD (LP MODE, 25°C)

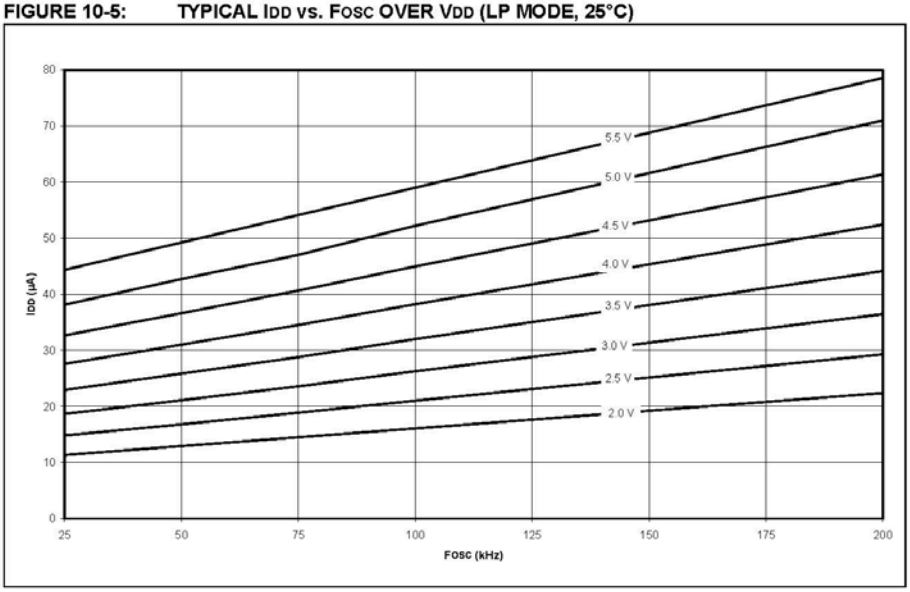


FIGURE 10-6: MAXIMUM IDD vs. FOSC OVER VDD (LP MODE, -40° TO +125°C)

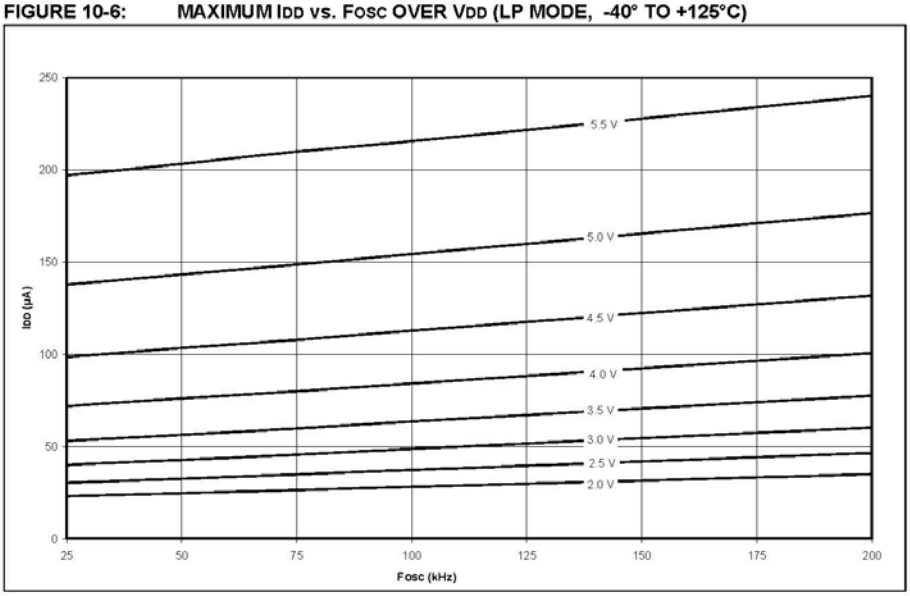


FIGURE 10-7: AVERAGE FOSC vs. VDD FOR R (RC MODE, C = 22 pF, 25°C)

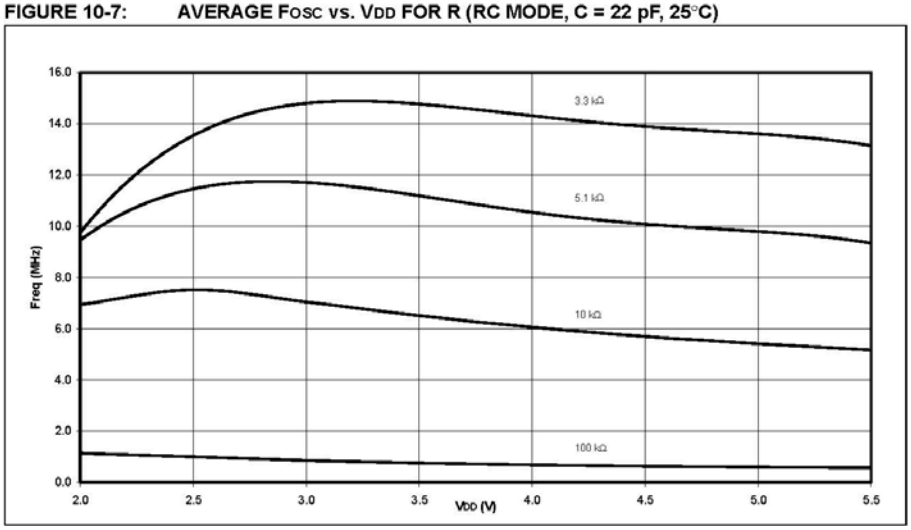


FIGURE 10-8: AVERAGE FOSC vs. VDD FOR R (RC MODE, C = 100 pF, 25°C)

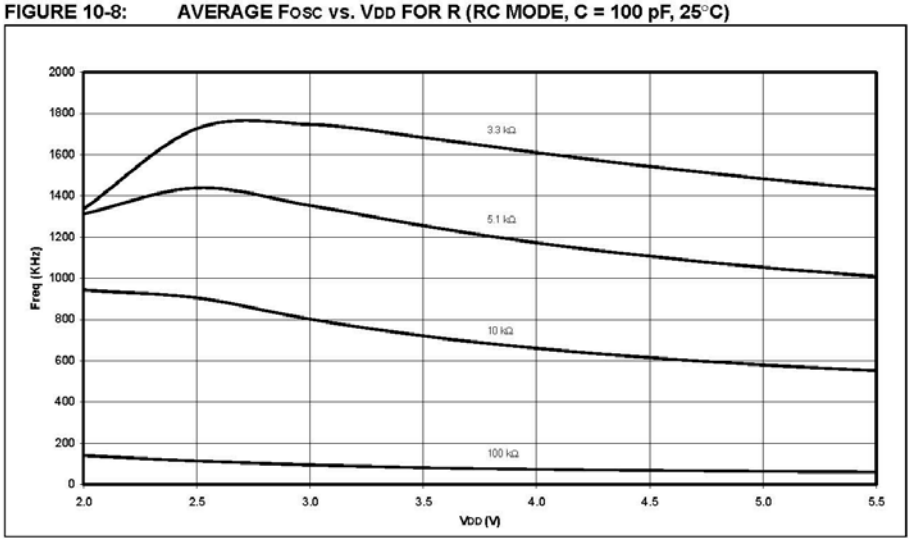


FIGURE 10-9: AVERAGE FOSC vs. VDD FOR R (RC MODE, C = 300 pF, 25°C)

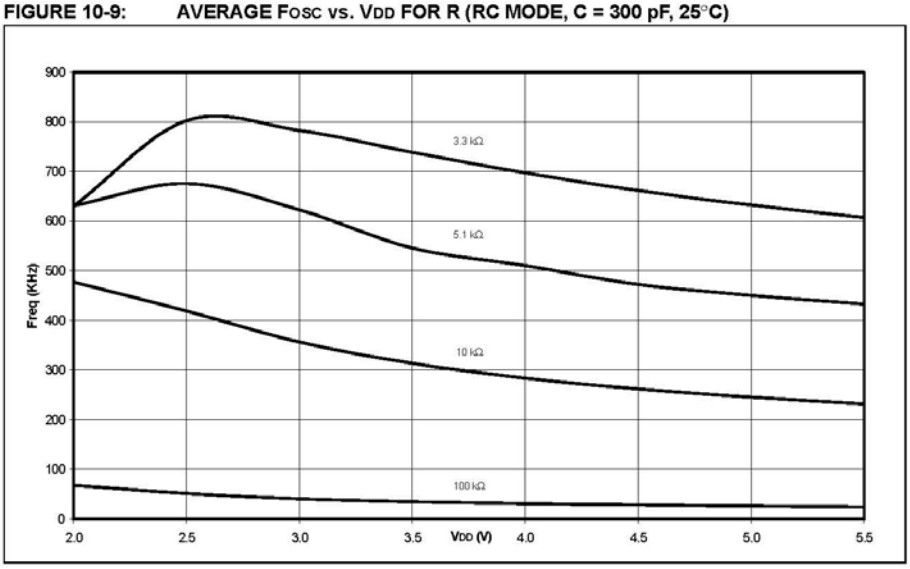


FIGURE 10-10: IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

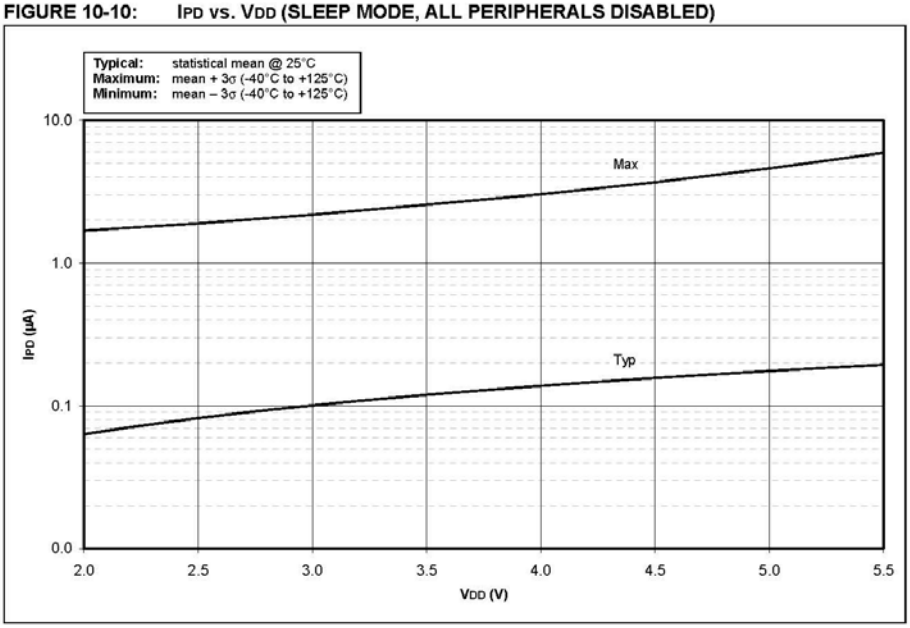


FIGURE 10-11: IPD vs. VDD (WDT MODE)

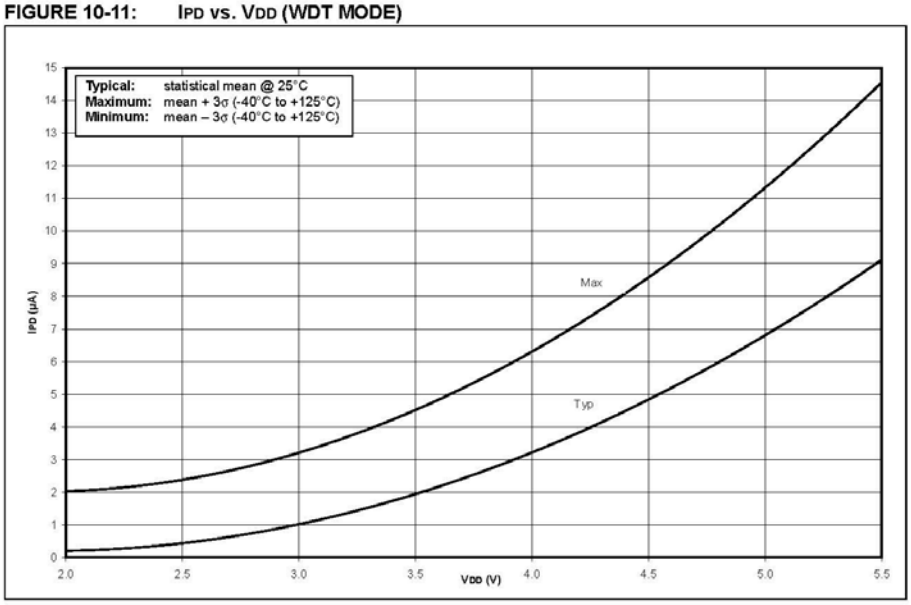


FIGURE 10-12: TYPICAL, MINIMUM, AND MAXIMUM WDT PERIOD vs. VDD OVER TEMP

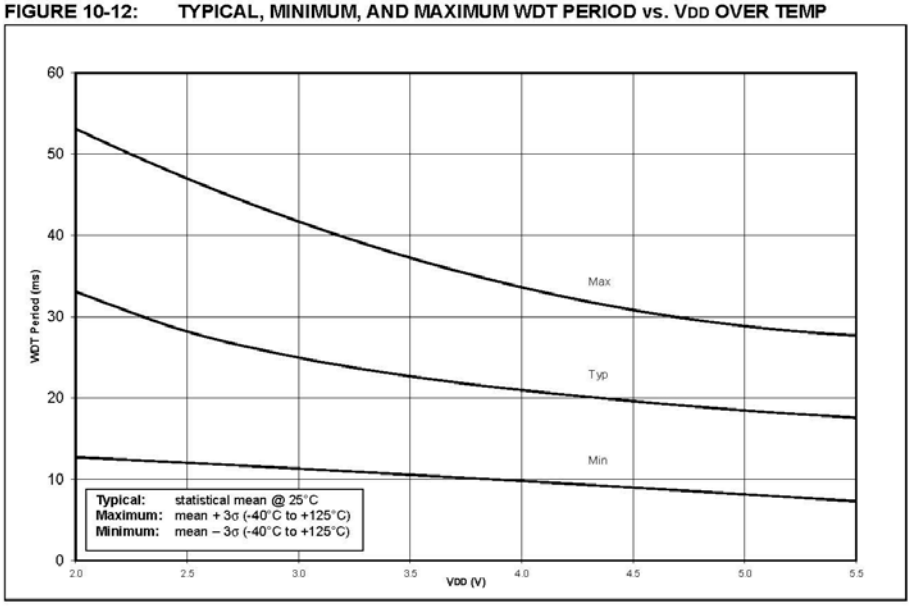


FIGURE 10-13: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 5V, -40°C TO +125°C)

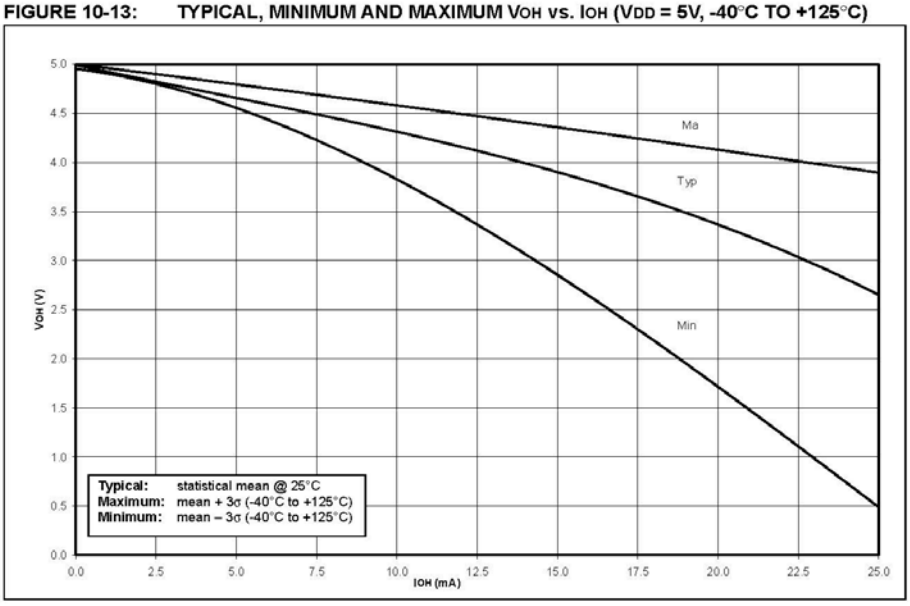


FIGURE 10-14: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 3V, -40°C TO +125°C)

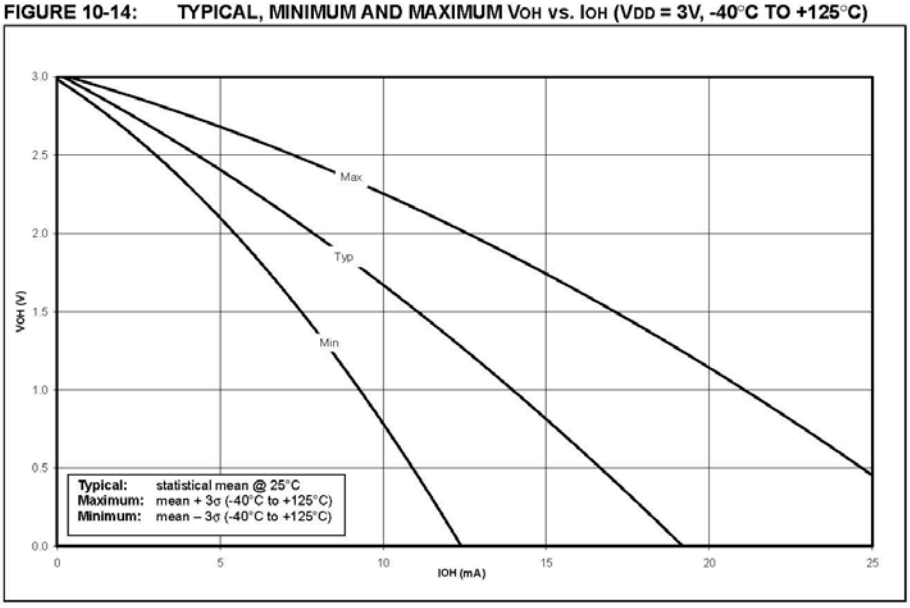


FIGURE 10-15: TYPICAL, MINIMUM AND MAXIMUM VOL vs. IOL (VDD = 5V, -40°C TO +125°C)

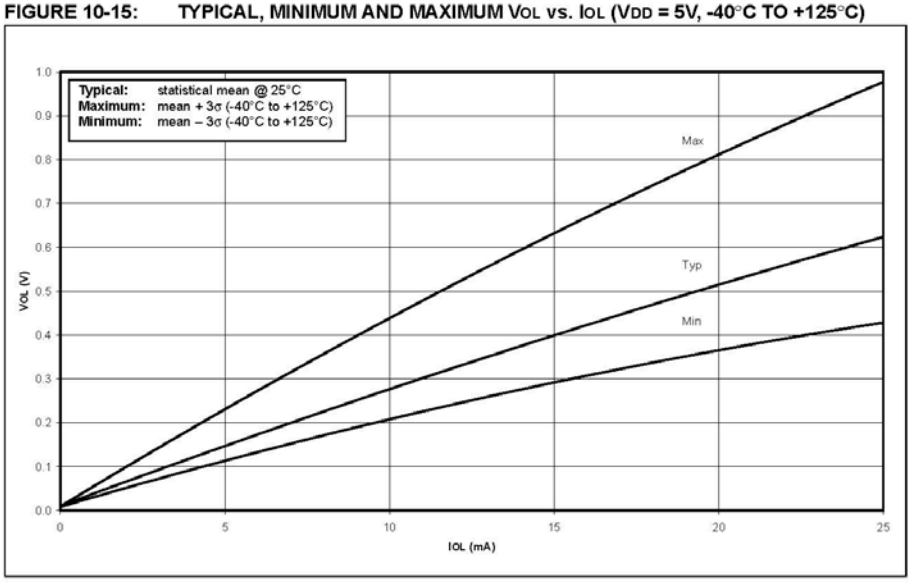


FIGURE 10-16: TYPICAL, MINIMUM AND MAXIMUM VOL vs. IOL (VDD = 3V, -40°C TO +125°C)

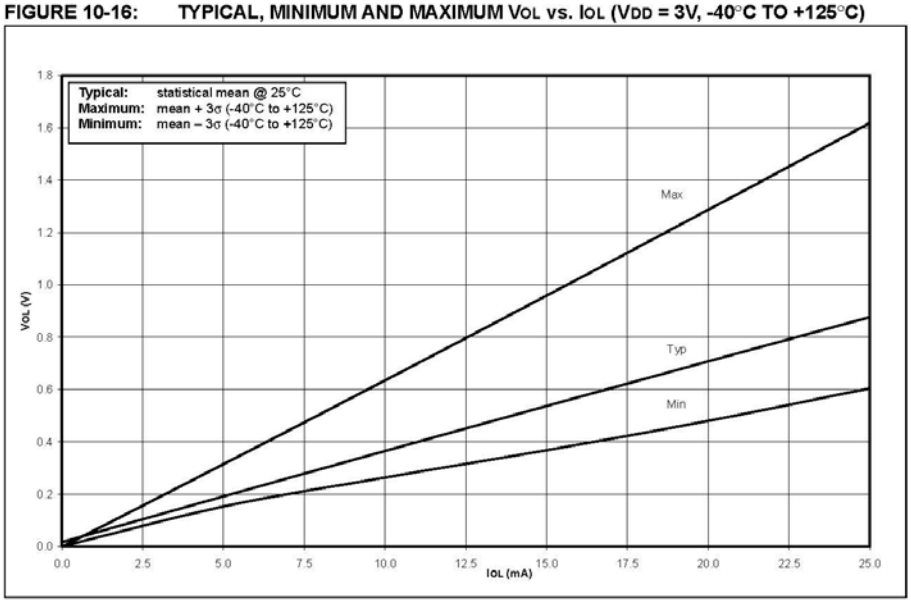


FIGURE 10-17: MINIMUM AND MAXIMUM VIN vs. VDD, (TTL INPUT, -40°C TO +125°C)

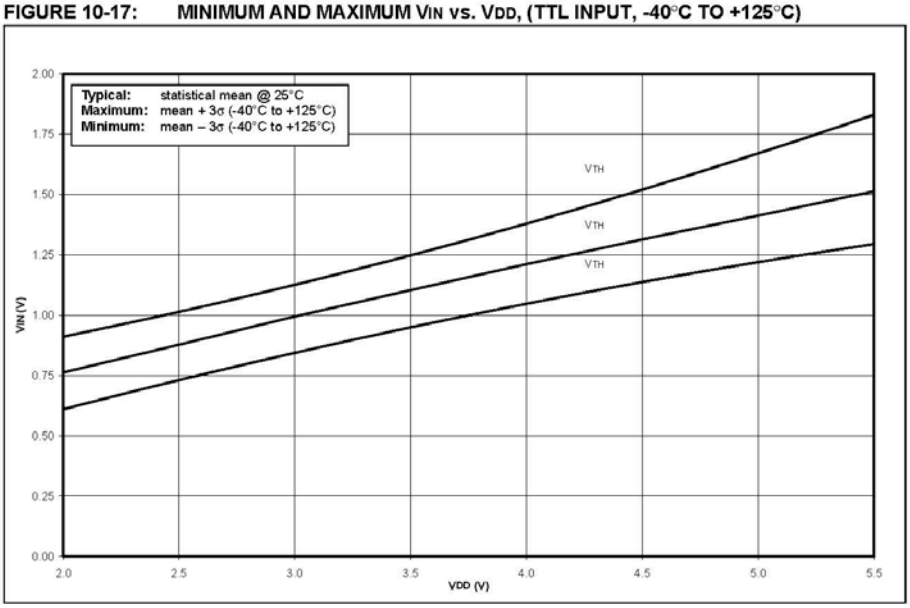
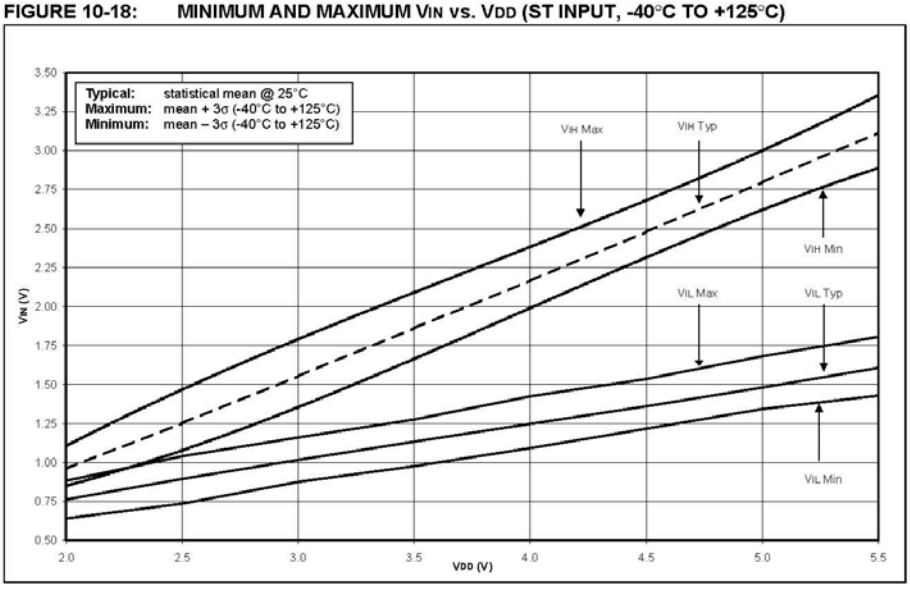


FIGURE 10-18: MINIMUM AND MAXIMUM VIN vs. VDD (ST INPUT, -40°C TO +125°C)

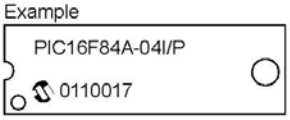


11.0 PACKAGING INFORMATION

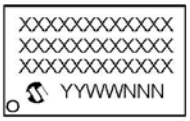
11.1 Package Marking Information

18-Lead PDIP

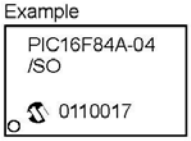
Example



18-Lead SOIC



Example



20-Lead SSOP

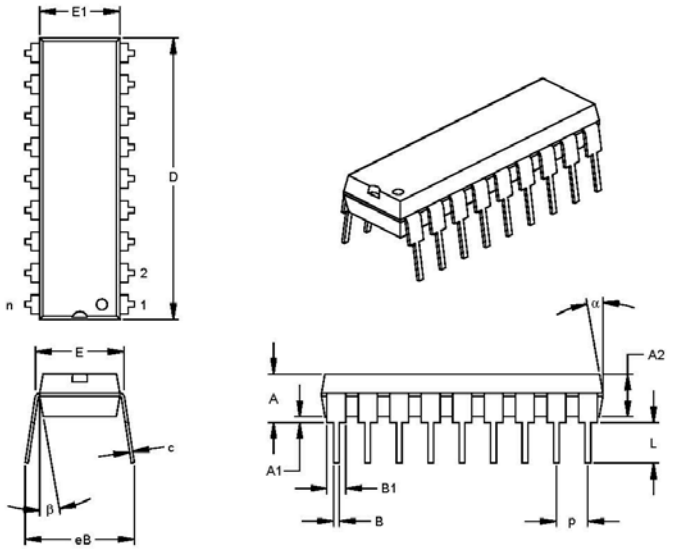
Example

Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip sales office. For QTP devices, any special marking adders are included in QTP price.

18-Lead Plastic Dual In-line (P) – 300 mil (PDIP)

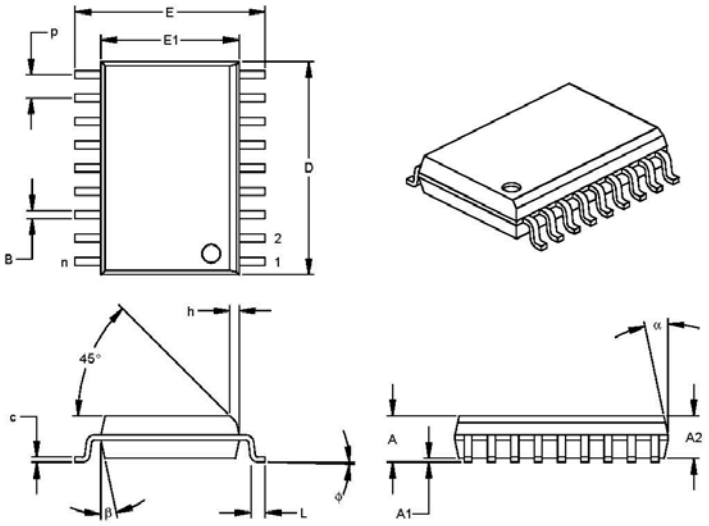


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-007

18-Lead Plastic Small Outline (SO) – Wide, 300 mil (SOIC)

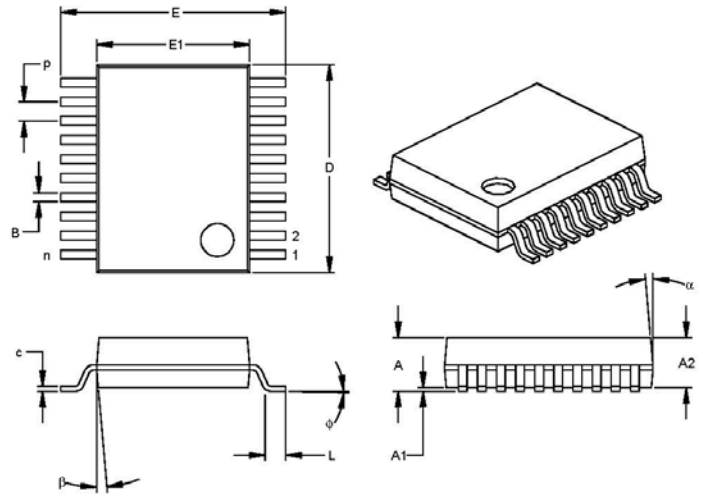


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MS-013
Drawing No. C04-051

20-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		20			20	
Pitch	p		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.322	7.59	7.85	8.18
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.278	.284	.289	7.06	7.20	7.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	φ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MO-150
Drawing No. C04-072

APPENDIX A: REVISION HISTORY

Version	Date	Revision Description
A	9/98	This is a new data sheet. However, the devices described in this data sheet are the upgrades to the devices found in the *PIC16F8X Data Sheet*, DS30430.
B	8/01	Added DC and AC Characteristics Graphs and Tables to Section 10.

APPENDIX B: CONVERSION CONSIDERATIONS

Considerations for converting from one PIC16X8X device to another are listed in Table 1.

TABLE 1: CONVERSION CONSIDERATIONS - PIC16C84, PIC16F83/F84, PIC16CR83/CR84, PIC16F84A

Difference	PIC16C84	PIC16F83/F84	PIC16CR83/ CR84	PIC16F84A
Program Memory Size	1K x 14	512 x 14 / 1K x 14	512 x 14 / 1K x 14	1K x 14
Data Memory Size	36 x 8	36 x 8 / 68 x 8	36 x 8 / 68 x 8	68 x 8
Voltage Range	2.0V - 6.0V (-40°C to +85°C)	2.0V - 6.0V (-40°C to +85°C)	2.0V - 6.0V (-40°C to +85°C)	2.0V - 5.5V (-40°C to +125°C)
Maximum Operating Frequency	10 MHz	10 MHz	10 MHz	20 MHz
Supply Current (I_{DD}). See parameter # D014 in the electrical specs for more detail.	I_{DD} (typ) = 60 µA I_{DD} (max) = 400 µA (LP osc, F_{OSC} = 32 kHz, V_{DD} = 2.0V, WDT disabled)	I_{DD} (typ) = 15 µA I_{DD} (max) = 45 µA (LP osc, F_{OSC} = 32 kHz, V_{DD} = 2.0V, WDT disabled)	I_{DD} (typ) = 15 µA I_{DD} (max) = 45 µA (LP osc, F_{OSC} = 32 kHz, V_{DD} = 2.0V, WDT disabled)	I_{DD} (typ) = 15 µA I_{DD} (max) = 45 µA (LP osc, F_{OSC} = 32 kHz, V_{DD} = 2.0V, WDT disabled)
Power-down Current (I_{PD}). See parameters # D020, D021, and D021A in the electrical specs for more detail.	I_{PD} (typ) = 26 µA I_{PD} (max) = 100 µA (V_{DD} = 2.0V, WDT disabled, industrial)	I_{PD} (typ) = 0.4 µA I_{PD} (max) = 9 µA (V_{DD} = 2.0V, WDT disabled, industrial)	I_{PD} (typ) = 0.4 µA I_{PD} (max) = 6 µA (V_{DD} = 2.0V, WDT disabled, industrial)	I_{PD} (typ) = 0.4 µA I_{PD} (max) = 1 µA (V_{DD} = 2.0V, WDT disabled, industrial)
Input Low Voltage (V_{IL}). See parameters # D032 and D034 in the electrical specs for more detail.	V_{IL} (max) = 0.2V_{DD} (OSC1, RC mode)	V_{IL} (max) = 0.1V_{DD} (OSC1, RC mode)	V_{IL} (max) = 0.1V_{DD} (OSC1, RC mode)	V_{IL} (max) = 0.1V_{DD} (OSC1, RC mode)
Input High Voltage (V_{IH}). See parameter # D040 in the electrical specs for more detail.	V_{IH} (min) = 0.36V_{DD} (I/O Ports with TTL, 4.5V ≤ V_{DD} ≤ 5.5V)	V_{IH} (min) = 2.4V (I/O Ports with TTL, 4.5V ≤ V_{DD} ≤ 5.5V)	V_{IH} (min) = 2.4V (I/O Ports with TTL, 4.5V ≤ V_{DD} ≤ 5.5V)	V_{IH} (min) = 2.4V (I/O Ports with TTL, 4.5V ≤ V_{DD} ≤ 5.5V)
Data EEPROM Memory Erase/Write cycle time (T_{DEW}). See parameter # D122 in the electrical specs for more detail.	T_{DEW} (typ) = 10 ms T_{DEW} (max) = 20 ms	T_{DEW} (typ) = 10 ms T_{DEW} (max) = 20 ms	T_{DEW} (typ) = 10 ms T_{DEW} (max) = 20 ms	T_{DEW} (typ) = 4 ms T_{DEW} (max) = 8 ms
Port Output Rise/Fall time (TioR, TioF). See parameters #20, 20A, 21, and 21A in the electrical specs for more detail.	TioR, TioF (max) = 25 ns (C84) TioR, TioF (max) = 60 ns (LC84)	TioR, TioF (max) = 35 ns (C84) TioR, TioF (max) = 70 ns (LC84)	TioR, TioF (max) = 35 ns (C84) TioR, TioF (max) = 70 ns (LC84)	TioR, TioF (max) = 35 ns (C84) TioR, TioF (max) = 70 ns (LC84)
$\overline{MCLR}$ on-chip filter. See parameter #30 in the electrical specs for more detail.	No	Yes	Yes	Yes
PORTA and crystal oscillator values less than 500 kHz	For crystal oscillator configurations operating below 500 kHz, the device may generate a spurious internal Q-clock when PORTA<0> switches state.	N/A	N/A	N/A
RB0/INT pin	TTL	TTL/ST* (*Schmitt Trigger)	TTL/ST* (*Schmitt Trigger)	TTL/ST* (*Schmitt Trigger)

TABLE 1: CONVERSION CONSIDERATIONS - PIC16C84, PIC16F83/F84, PIC16CR83/CR84, PIC16F84A (CONTINUED)

Difference	PIC16C84	PIC16F83/F84	PIC16CR83/ CR84	PIC16F84A
EEADR<7:6> and IDD	It is recommended that the EEADR<7:6> bits be cleared. When either of these bits is set, the maximum IDD for the device is higher than when both are cleared.	N/A	N/A	N/A
The polarity of the $\overline{\text{PWRTE}}$ bit	PWRTE	$\overline{\text{PWRTE}}$	$\overline{\text{PWRTE}}$	$\overline{\text{PWRTE}}$
Recommended value of REXT for RC oscillator circuits	REXT = 3kΩ - 100kΩ	REXT = 5kΩ - 100kΩ	REXT = 5kΩ - 100kΩ	REXT = 3kΩ - 100kΩ
GIE bit unintentional enable	If an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction).	N/A	N/A	N/A
Packages	PDIP, SOIC	PDIP, SOIC	PDIP, SOIC	PDIP, SOIC, SSOP
Open Drain High Voltage (VOD)	14V	12V	12V	8.5V

APPENDIX C: MIGRATION FROM BASELINE TO MID-RANGE DEVICES

This section discusses how to migrate from a baseline device (i.e., PIC16C5X) to a mid-range device (i.e., PIC16CXXX).

The following is the list of feature improvements over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14-bits. This allows larger page sizes, both in program memory (2K now as opposed to 512K before) and the register file (128 bytes now versus 32 bytes before).
2. A PC latch register (PCLATH) is added to handle program memory paging. PA2, PA1 and PA0 bits are removed from the STATUS register and placed in the OPTION register.
3. Data memory paging is redefined slightly. The STATUS register is modified.
4. Four new instructions have been added: `RETURN`, `RETFIE`, `ADDLW`, and `SUBLW`. Two instructions, `TRIS` and `OPTION`, are being phased out, although they are kept for compatibility with PIC16C5X.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to eight-deep.
8. RESET vector is changed to 0000h.
9. RESET of all registers is revisited. Five different RESET (and wake-up) types are recognized. Registers are reset differently.
10. Wake-up from SLEEP through interrupt is added.
11. Two separate timers, the Oscillator Start-up Timer (OST) and Power-up Timer (PWRT), are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt-on-change features.
13. T0CKI pin is also a port pin (RA4/T0CKI).
14. FSR is a full 8-bit register.
15. "In system programming" is made possible. The user can program PIC16CXX devices using only five pins: VDD, VSS, VPP, RB6 (clock) and RB7 (data in/out).

To convert code written for PIC16C5X to PIC16F84A, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for `CALL`, `GOTO`.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables for reallocation.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change RESET vector to 0000h.

INDEX

NOTES:

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® or Microsoft® Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-792-7302 for the rest of the world.

013001

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16F84A Literature Number: DS35007B

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?

PIC16F84A PRODUCT IDENTIFICATION SYSTEM

To order or obtain information (e.g., on pricing or delivery) refer to the factory or the listed sales office.

PART NO.	-XX	X	/XX	XXX
Device	Frequency Range	Temperature Range	Package	Pattern

Device	PIC16F84A(1), PIC16F84AT(2) PIC16LF84A(1), PIC16LF84AT(2)
Frequency Range	04 = 4 MHz 20 = 20 MHz
Temperature Range	- = 0°C to +70°C I = -40°C to +85°C
Package	P = PDIP SO = SOIC (Gull Wing, 300 mil body) SS = SSOP
Pattern	QTP, SQTP, ROM Code (factory specified) or Special Requirements . Blank for OTP and Windowed devices.

Examples:

a) PIC16F84A -04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301.

b) PIC16LF84A - 04I/SO = Industrial temp., SOIC package, 200 kHz, Extended VDD limits.

c) PIC16F84A - 20I/P = Industrial temp., PDIP package, 20 MHz, normal VDD limits.

Note 1: F = Standard VDD range
LF = Extended VDD range
2: T = in tape and reel - SOIC and SSOP packages only.

Sales and Support

Data Sheets
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 792-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System
Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Austin - Analog
13740 North Highway 183
Building J, Suite 4
Austin, TX 78750
Tel: 512-257-3370 Fax: 512-257-8526

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Boston - Analog
Unit A-8-1 Millbrook Tarry Condominium
97 Lowell Road
Concord, MA 01742
Tel: 978-371-6400 Fax: 978-371-0050

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Dayton
Two Prestige Place, Suite 130
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Rm. 531, North Building
Fujian Foreign Trade Center Hotel
73 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7557563 Fax: 86-591-7557572

China - Shanghai
Microchip Technology Consulting (Shanghai) Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Denmark ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Arizona Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - Ier Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Germany - Analog
Lochhamer Strasse 13
D-82152 Martinsried, Germany
Tel: 49-89-895650-0 Fax: 49-89-895650-22

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

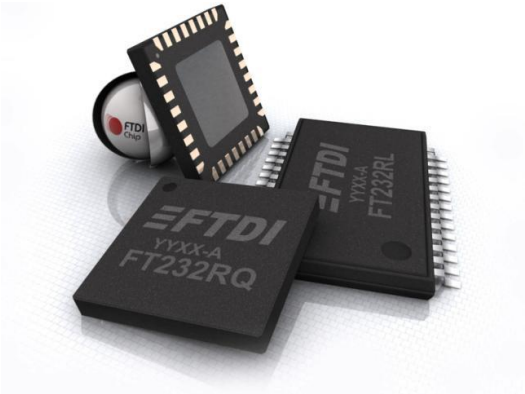
United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

08/01/01

Document No.: FT_000053
FT232R USB UART IC Datasheet Version 2.10
Clearance No.: FTDI# 38

Future Technology Devices International Ltd.

FT232R USB UART IC



The FT232R is a USB to serial UART interface with the following advanced features:

- Single chip USB to asynchronous serial data transfer interface.
- Entire USB protocol handled on the chip. No USB specific firmware programming required.
- Fully integrated 1024 bit EEPROM storing device descriptors and CBUS I/O configuration.
- Fully integrated USB termination resistors.
- Fully integrated clock generation with no external crystal required plus optional clock output selection enabling a glue-less interface to external MCU or FPGA.
- Data transfer rates from 300 baud to 3 Mbaud (RS422, RS485, RS232) at TTL levels.
- 128 byte receive buffer and 256 byte transmit buffer utilising buffer smoothing technology to allow for high data throughput.
- FTDI's royalty-free Virtual Com Port (VCP) and Direct (D2XX) drivers eliminate the requirement for USB driver development in most cases.
- Unique USB FTDIChip-ID™ feature.
- Configurable CBUS I/O pins.
- Transmit and receive LED drive signals.
- UART interface support for 7 or 8 data bits, 1 or 2 stop bits and odd / even / mark / space / no parity
- FIFO receive and transmit buffers for high data throughput.
- Synchronous and asynchronous bit bang interface options with RD# and WR# strobes.
- Device supplied pre-programmed with unique USB serial number.
- Supports bus powered, self powered and high-power bus powered USB configurations.
- Integrated +3.3V level converter for USB I/O.
- Integrated level converter on UART and CBUS for interfacing to between +1.8V and +5V logic.
- True 5V/3.3V/2.8V/1.8V CMOS drive output and TTL input.
- Configurable I/O pin output drive strength.
- Integrated power-on-reset circuit.
- Fully integrated AVCC supply filtering - no external filtering required.
- UART signal inversion option.
- +3.3V (using external oscillator) to +5.25V (internal oscillator) Single Supply Operation.
- Low operating and USB suspend current.
- Low USB bandwidth consumption.
- UHCI/OHCI/EHCI host controller compatible.
- USB 2.0 Full Speed compatible.
- -40°C to 85°C extended operating temperature range.
- Available in compact Pb-free 28 Pin SSOP and QFN-32 packages (both RoHS compliant).

1 Typical Applications

- USB to RS232/RS422/RS485 Converters
- Upgrading Legacy Peripherals to USB
- Cellular and Cordless Phone USB data transfer cables and interfaces
- Interfacing MCU/PLD/FPGA based designs to USB
- USB Audio and Low Bandwidth Video data transfer
- PDA to USB data transfer
- USB Smart Card Readers
- USB Instrumentation
- USB Industrial Control
- USB MP3 Player Interface
- USB FLASH Card Reader and Writers
- Set Top Box PC - USB interface
- USB Digital Camera Interface
- USB Hardware Modems
- USB Wireless Modems
- USB Bar Code Readers
- USB Software and Hardware Encryption Dongles

1.1 Driver Support

Royalty free VIRTUAL COM PORT (VCP) DRIVERS for...

- Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
- Windows 7 32,64-bit
- Windows XP and XP 64-bit
- Windows Vista and Vista 64-bit
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Mac OS 8/9, OS-X
- Linux 2.4 and greater

Royalty free D2XX *Direct* Drivers (USB Drivers + DLL S/W Interface)

- Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
- Windows 7 32,64-bit
- Windows XP and XP 64-bit
- Windows Vista and Vista 64-bit
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Linux 2.4 and greater

The drivers listed above are all available to download for free from FTDI website (www.ftdichip.com). Various 3rd party drivers are also available for other operating systems - see FTDI website (www.ftdichip.com) for details.

For driver installation, please refer to http://www.ftdichip.com/Documents/InstallGuides.htm

1.2 Part Numbers

Part Number	Package
FT232RQ-xxxx	32 Pin QFN
FT232RL-xxxx	28 Pin SSOP

Note: Packing codes for xxxx is:

- Reel: Taped and Reel, (SSOP is 2,000pcs per reel, QFN is 6,000pcs per reel).
- Tube: Tube packing, 47pcs per tube (SSOP only)
- Tray: Tray packing, 490pcs per tray (QFN only)

For example: FT232RQ-Reel is 6,000pcs taped and reel packing

1.3 USB Compliant

The FT232R is fully compliant with the USB 2.0 specification and has been given the USB-IF Test-ID (TID) 40680004 (Rev B) and 40770018 (Rev C).

2 FT232R Block Diagram

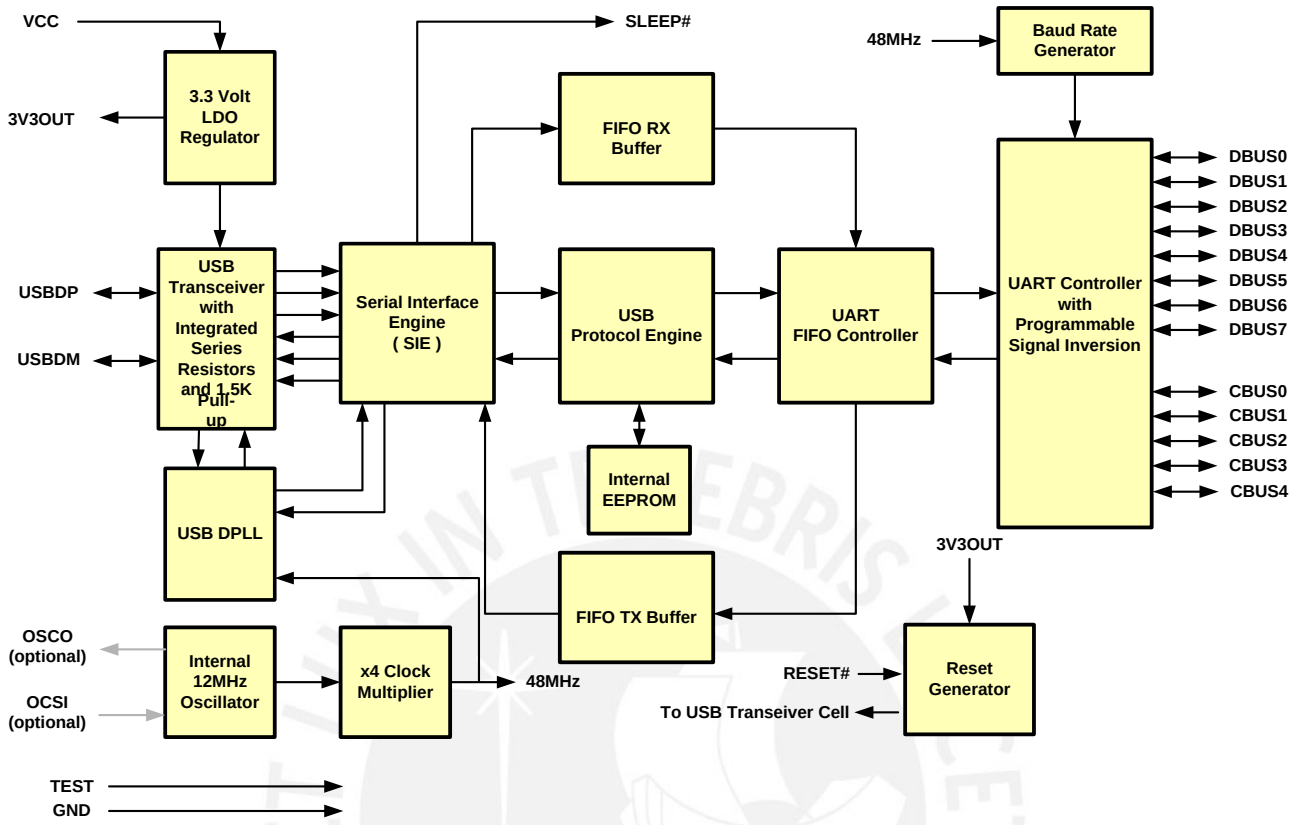


Figure 2.1 FT232R Block Diagram

For a description of each function please refer to Section 4.

Table of Contents

3 Device Pin Out and Signal Description

3.1 28-LD SSOP Package

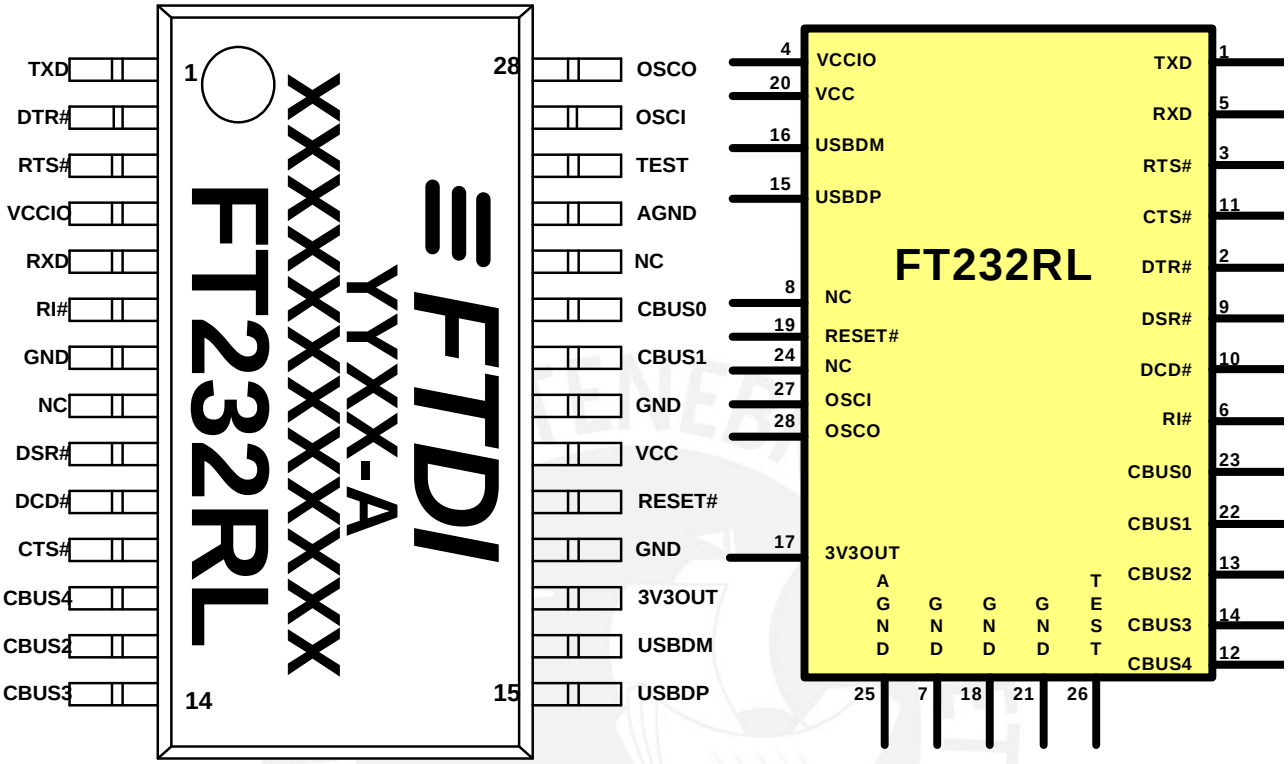


Figure 3.1 SSOP Package Pin Out and Schematic Symbol

3.2 SSOP Package Pin Out Description

Note: The convention used throughout this document for active low signals is the signal name followed by a #

Pin No.	Name	Type	Description
15	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to 3.3V.
16	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.

Table 3.1 USB Interface Group

Pin No.	Name	Type	Description
4	VCCIO	PWR	+1.8V to +5.25V supply to the UART Interface and CBUS group pins (1...3, 5, 6, 9...14, 22, 23). In USB bus powered designs connect this pin to 3V3OUT pin to drive out at +3.3V levels, or connect to VCC to drive out at 5V CMOS level. This pin can also be supplied with an external +1.8V to +2.8V supply in order to drive outputs at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to VCC. This means that in bus powered designs a regulator which is supplied by the +5V on the USB bus should be used.
7, 18, 21	GND	PWR	Device ground supply pins

Pin No.	Name	Type	Description
17	3V3OUT	Output	+3.3V output from integrated LDO regulator. This pin should be decoupled to ground using a 100nF capacitor. The main use of this pin is to provide the internal +3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the VCCIO pin.
20	VCC	PWR	+3.3V to +5.25V supply to the device core. (see Note 1)
25	AGND	PWR	Device analogue ground supply for internal clock multiplier

Table 3.2 Power and Ground Group

Pin No.	Name	Type	Description
8, 24	NC	NC	No internal connection
19	RESET#	Input	Active low reset pin. This can be used by an external device to reset the FT232R. If not required can be left unconnected, or pulled up to VCC.
26	TEST	Input	Puts the device into IC test mode. Must be tied to GND for normal operation, otherwise the device will appear to fail.
27	OSCI	Input	Input 12MHz Oscillator Cell. Optional – Can be left unconnected for normal operation. (see Note 2)
28	OSCO	Output	Output from 12MHZ Oscillator Cell. Optional – Can be left unconnected for normal operation if internal Oscillator is used. (see Note 2)

Table 3.3 Miscellaneous Signal Group

Pin No.	Name	Type	Description
1	TXD	Output	Transmit Asynchronous Data Output.
2	DTR#	Output	Data Terminal Ready Control Output / Handshake Signal.
3	RTS#	Output	Request to Send Control Output / Handshake Signal.
5	RXD	Input	Receiving Asynchronous Data Input.
6	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low (20ms active low pulse) can be used to resume the PC USB host controller from suspend.
9	DSR#	Input	Data Set Ready Control Input / Handshake Signal.
10	DCD#	Input	Data Carrier Detect Control Input.
11	CTS#	Input	Clear To Send Control Input / Handshake Signal.
12	CBUS4	I/O	Configurable CBUS output only Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is SLEEP#. See CBUS Signal Options, Table 3.9.
13	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXDEN. See CBUS Signal Options, Table 3.9.

Pin No.	Name	Type	Description
14	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is PWREN#. See CBUS Signal Options, Table 3.9. PWREN# should be used with a 10kΩ resistor pull up.
22	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is RXLED#. See CBUS Signal Options, Table 3.9.
23	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXLED#. See CBUS Signal Options, Table 3.9.

Table 3.4 UART Interface and CUSB Group (see note 3)

Notes:

1. The minimum operating voltage VCC must be +4.0V (could use VBUS=+5V) when using the internal clock generator. Operation at +3.3V is possible using an external crystal oscillator.
2. For details on how to use an external crystal, ceramic resonator, or oscillator with the FT232R, please refer Section 7.6
3. When used in Input Mode, the input pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

3.3 QFN-32 Package

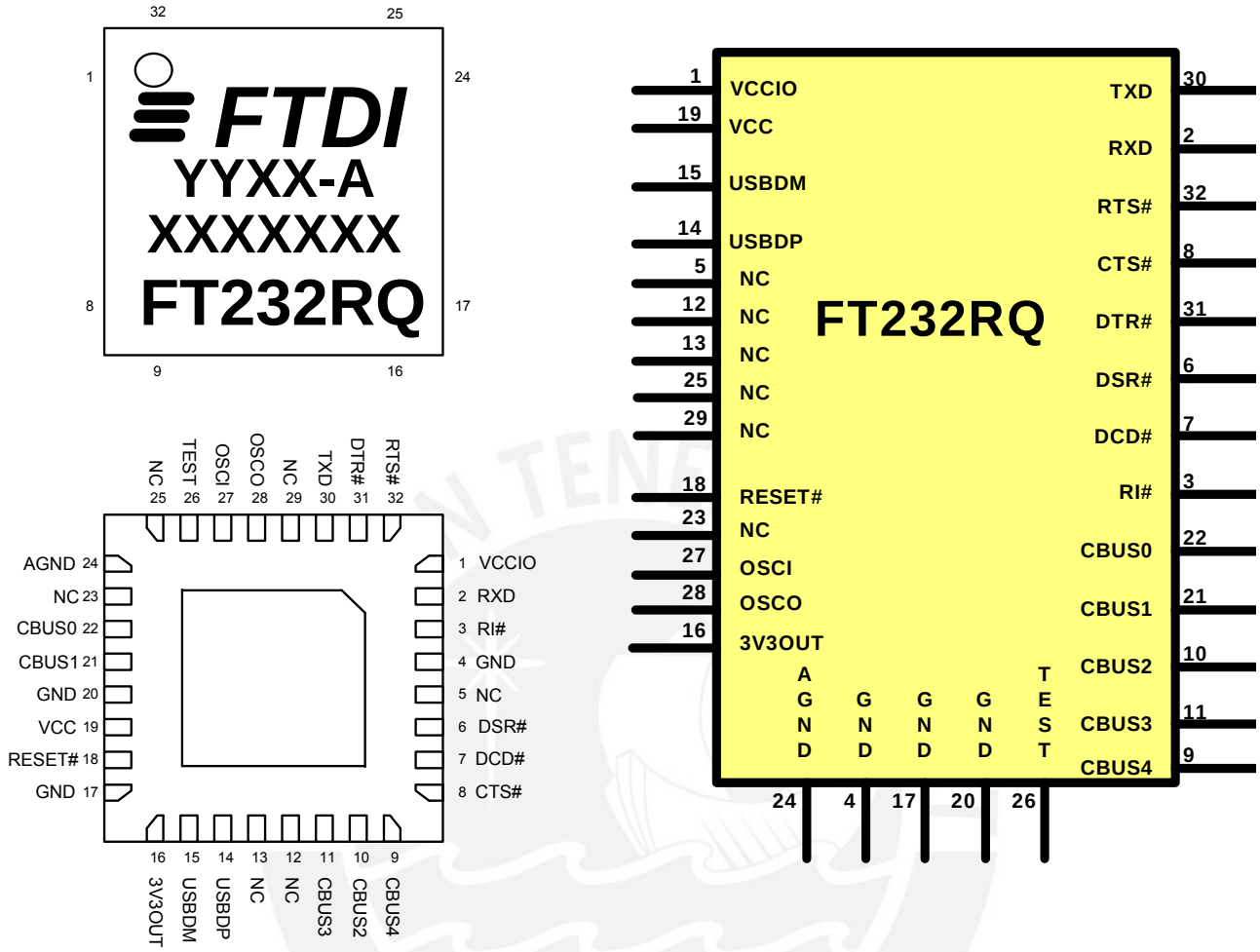


Figure 3.2 QFN-32 Package Pin Out and schematic symbol

3.4 QFN-32 Package Signal Description

Pin No.	Name	Type	Description
14	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to +3.3V.
15	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.

Table 3.5 USB Interface Group

Pin No.	Name	Type	Description
1	VCCIO	PWR	+1.8V to +5.25V supply for the UART Interface and CBUS group pins (2, 3, 6,7,8,9,10 11, 21, 22, 30,31,32). In USB bus powered designs connect this pin to 3V3OUT to drive out at +3.3V levels, or connect to VCC to drive out at +5V CMOS level. This pin can also be supplied with an external +1.8V to +2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to VCC. This means that in bus powered designs a regulator which is supplied by the +5V on the USB bus should be used.
4, 17, 20	GND	PWR	Device ground supply pins.

Pin No.	Name	Type	Description
16	3V3OUT	Output	+3.3V output from integrated LDO regulator. This pin should be decoupled to ground using a 100nF capacitor. The purpose of this output is to provide the internal +3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the VCCIO pin.
19	VCC	PWR	+3.3V to +5.25V supply to the device core. (See Note 1).
24	AGND	PWR	Device analogue ground supply for internal clock multiplier.

Table 3.6 Power and Ground Group

Pin No.	Name	Type	Description
5, 12, 13, 23, 25, 29	NC	NC	No internal connection. Do not connect.
18	RESET#	Input	Active low reset. Can be used by an external device to reset the FT232R. If not required can be left unconnected, or pulled up to VCC.
26	TEST	Input	Puts the device into IC test mode. Must be tied to GND for normal operation, otherwise the device will appear to fail.
27	OSCI	Input	Input 12MHz Oscillator Cell. Optional – Can be left unconnected for normal operation. (See Note 2).
28	OSCO	Output	Output from 12MHZ Oscillator Cell. Optional – Can be left unconnected for normal operation if internal Oscillator is used. (See Note 2).

Table 3.7 Miscellaneous Signal Group

Pin No.	Name	Type	Description
30	TXD	Output	Transmit Asynchronous Data Output.
31	DTR#	Output	Data Terminal Ready Control Output / Handshake Signal.
32	RTS#	Output	Request to Send Control Output / Handshake Signal.
2	RXD	Input	Receiving Asynchronous Data Input.
3	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low (20ms active low pulse) can be used to resume the PC USB host controller from suspend.
6	DSR#	Input	Data Set Ready Control Input / Handshake Signal.
7	DCD#	Input	Data Carrier Detect Control Input.
8	CTS#	Input	Clear To Send Control Input / Handshake Signal.
9	CBUS4	I/O	Configurable CBUS output only Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is SLEEP#. See CBUS Signal Options, Table 3.9.
10	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXDEN. See CBUS Signal Options, Table 3.9.

Pin No.	Name	Type	Description
11	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is PWREN#. See CBUS Signal Options, Table 3.9. PWREN# should be used with a 10kΩ resistor pull up.
21	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is RXLED#. See CBUS Signal Options, Table 3.9.
22	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXLED#. See CBUS Signal Options, Table 3.9.

Table 3.8 UART Interface and CBUS Group (see note 3)

Notes:

1. The minimum operating voltage VCC must be +4.0V (could use VBUS=+5V) when using the internal clock generator. Operation at +3.3V is possible using an external crystal oscillator.
2. For details on how to use an external crystal, ceramic resonator, or oscillator with the FT232R, please refer to Section 7.6.
3. When used in Input Mode, the input pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

3.5 CBUS Signal Options

The following options can be configured on the CBUS I/O pins. CBUS signal options are common to both package versions of the FT232R. These options can be configured in the internal EEPROM using the software utility FT_PPROG or MPROG, which can be downloaded from the FTDI Utilities (www.ftdichip.com). The default configuration is described in Section 8.

CBUS Signal Option	Available On CBUS Pin	Description
TXDEN	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Enable transmit data for RS485
PWREN#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Output is low after the device has been configured by USB, then high during USB suspend mode. This output can be used to control power to external logic P-Channel logic level MOSFET switch. Enable the interface pull-down option when using the PWREN# in this way.*
TXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Transmit data LED drive: Data from USB Host to FT232R. Pulses low when transmitting data via USB. See Section 7.5 for more details.
RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Receive data LED drive: Data from FT232R to USB Host. Pulses low when receiving data via USB. See Section 7.5 for more details.
TX&RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	LED drive – pulses low when transmitting or receiving data via USB. See Section 7.5 for more details.
SLEEP#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Goes low during USB suspend mode. Typically used to power down an external TTL to RS232 level converter IC in USB to RS232 converter designs.
CLK48	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	48MHz ±0.7% Clock output. **
CLK24	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	24 MHz Clock output.**
CLK12	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	12 MHz Clock output.**
CLK6	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	6 MHz ±0.7% Clock output. **
CBitBangI/O	CBUS0, CBUS1, CBUS2, CBUS3	CBUS bit bang mode option. Allows up to 4 of the CBUS pins to be used as general purpose I/O. Configured individually for CBUS0, CBUS1, CBUS2 and CBUS3 in the internal EEPROM. A separate application note, AN232R-01, available from FTDI website (www.ftdichip.com) describes in more detail how to use CBUS bit bang mode.
BitBangWRn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode WR# strobe output.
BitBangRDn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode RD# strobe output.

Table 3.9 CBUS Configuration Control

* PWREN# must be used with a 10kΩ resistor pull up.

**When in USB suspend mode the outputs clocks are also suspended.

4 Function Description

The FT232R is a USB to serial UART interface device which simplifies USB to serial designs and reduces external component count by fully integrating an external EEPROM, USB termination resistors and an integrated clock circuit which requires no external crystal, into the device. It has been designed to operate efficiently with a USB host controller by using as little as possible of the total USB bandwidth available.

4.1 Key Features

Functional Integration. Fully integrated EEPROM, USB termination resistors, clock generation, AVCC filtering, POR and LDO regulator.

Configurable CBUS I/O Pin Options. The fully integrated EEPROM allows configuration of the Control Bus (CBUS) functionality, signal inversion and drive strength selection. There are 5 configurable CBUS I/O pins. These configurable options are

1. **TXDEN** - transmit enable for RS485 designs.
2. **PWREN#** - Power control for high power, bus powered designs.
3. **TXLED#** - for pulsing an LED upon transmission of data.
4. **RXLED#** - for pulsing an LED upon receiving data.
5. **TX&RXLED#** - which will pulse an LED upon transmission OR reception of data**.**
6. **SLEEP#** - indicates that the device going into USB suspend mode.
7. **CLK48 / CLK24 / CLK12 / CLK6** - 48MHz, 24MHz, 12MHz, and 6MHz clock output signal options.

The CBUS pins can also be individually configured as GPIO pins, similar to asynchronous bit bang mode. It is possible to use this mode while the UART interface is being used, thus providing up to 4 general purpose I/O pins which are available during normal operation. An application note, AN232R-01, available from FTDI website (www.ftdichip.com) describes this feature.

The CBUS lines can be configured with any one of these output options by setting bits in the internal EEPROM. The device is supplied with the most commonly used pin definitions pre-programmed - see Section 8 for details.

Asynchronous Bit Bang Mode with RD# and WR# Strobes. The FT232R supports FTDI's previous chip generation bit-bang mode. In bit-bang mode, the eight UART lines can be switched from the regular interface mode to an 8-bit general purpose I/O port. Data packets can be sent to the device and they will be sequentially sent to the interface at a rate controlled by an internal timer (equivalent to the baud rate pre-scaler). With the FT232R device this mode has been enhanced by outputting the internal RD# and WR# strobes signals which can be used to allow external logic to be clocked by accesses to the bit-bang I/O bus. This option will be described more fully in a separate application note available from FTDI website (www.ftdichip.com).

Synchronous Bit Bang Mode. The FT232R supports synchronous bit bang mode. This mode differs from asynchronous bit bang mode in that the interface pins are only read when the device is written to. This makes it easier for the controlling program to measure the response to an output stimulus as the data returned is synchronous to the output data. An application note, AN232R-01, available from FTDI website (www.ftdichip.com) describes this feature.

FTDIChip-ID™. The FT232R also includes the new FTDIChip-ID™ security dongle feature. This FTDIChip-ID™ feature allows a unique number to be burnt into each device during manufacture. This number cannot be reprogrammed. This number is only readable over USB and forms a basis of a security dongle which can be used to protect any customer application software being copied. This allows the possibility of using the FT232R in a dongle for software licensing. Further to this, a renewable license scheme can be implemented based on the FTDIChip-ID™ number when encrypted with other information. This encrypted number can be stored in the user area of the FT232R internal EEPROM, and can be decrypted, then compared with the protected FTDIChip-ID™ to verify that a license is valid. Web based applications can be used to maintain product licensing this way. An application note, AN232R-02, available from FTDI website (www.ftdichip.com) describes this feature.

The FT232R is capable of operating at a voltage supply between +3.3V and +5V with a nominal operational mode current of 15mA and a nominal USB suspend mode current of 70µA. This allows greater margin for peripheral designs to meet the USB suspend mode current limit of 2.5mA. An integrated level converter within the UART interface allows the FT232R to interface to UART logic running at +1.8V, 2.5V, +3.3V or +5V.

4.2 Functional Block Descriptions

The following paragraphs detail each function within the FT232R. Please refer to the block diagram shown in Figure 2.1

Internal EEPROM. The internal EEPROM in the FT232R is used to store USB Vendor ID (VID), Product ID (PID), device serial number, product description string and various other USB configuration descriptors. The internal EEPROM is also used to configure the CBUS pin functions. The FT232R is supplied with the internal EEPROM pre-programmed as described in Section 8. A user area of the internal EEPROM is available to system designers to allow storing additional data. The internal EEPROM descriptors can be programmed in circuit, over USB without any additional voltage requirement. It can be programmed using the FTDI utility software called MPROG, which can be downloaded from FTDI Utilities on the FTDI website (www.ftdichip.com).

+3.3V LDO Regulator. The +3.3V LDO regulator generates the +3.3V reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It also provides +3.3V power to the 1.5kΩ internal pull up resistor on USBDP. The main function of the LDO is to power the USB Transceiver and the Reset Generator Cells rather than to power external logic. However, it can be used to supply external circuitry requiring a +3.3V nominal supply with a maximum current of 50mA.

USB Transceiver. The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB cable. The output drivers provide +3.3V level slew rate control signalling, whilst a differential input receiver and two single ended input receivers provide USB data in, Single-Ended-0 (SE0) and USB reset detection conditions respectfully. This function also incorporates the internal USB series termination resistors on the USB data lines and a 1.5kΩ pull up resistor on USBDP.

USB DPLL. The USB DPLL cell locks on to the incoming NRZI USB data and generates recovered clock and data signals for the Serial Interface Engine (SIE) block.

Internal 12MHz Oscillator - The Internal 12MHz Oscillator cell generates a 12MHz reference clock. This provides an input to the x4 Clock Multiplier function. The 12MHz Oscillator is also used as the reference clock for the SIE, USB Protocol Engine and UART FIFO controller blocks.

Clock Multiplier / Divider. The Clock Multiplier / Divider takes the 12MHz input from the Internal Oscillator function and generates the 48MHz, 24MHz, 12MHz and 6MHz reference clock signals. The 48Mz clock reference is used by the USB DPLL and the Baud Rate Generator blocks.

Serial Interface Engine (SIE). The Serial Interface Engine (SIE) block performs the parallel to serial and serial to parallel conversion of the USB data. In accordance with the USB 2.0 specification, it performs bit stuffing/un-stuffing and CRC5/CRC16 generation. It also checks the CRC on the USB data stream.

USB Protocol Engine. The USB Protocol Engine manages the data stream from the device USB control endpoint. It handles the low level USB protocol requests generated by the USB host controller and the commands for controlling the functional parameters of the UART in accordance with the USB 2.0 specification chapter 9.

FIFO RX Buffer (128 bytes). Data sent from the USB host controller to the UART via the USB data OUT endpoint is stored in the FIFO RX (receive) buffer. Data is removed from the buffer to the UART transmit register under control of the UART FIFO controller. (Rx relative to the USB interface).

FIFO TX Buffer (256 bytes). Data from the UART receive register is stored in the TX buffer. The USB host controller removes data from the FIFO TX Buffer by sending a USB request for data from the device data IN endpoint. (Tx relative to the USB interface).

UART FIFO Controller. The UART FIFO controller handles the transfer of data between the FIFO RX and TX buffers and the UART transmit and receive registers.

UART Controller with Programmable Signal Inversion and High Drive. Together with the UART FIFO Controller the UART Controller handles the transfer of data between the FIFO RX and FIFO TX buffers and the UART transmit and receive registers. It performs asynchronous 7 or 8 bit parallel to serial and serial to parallel conversion of the data on the RS232 (or RS422 or RS485) interface.

Control signals supported by UART mode include RTS, CTS, DSR, DTR, DCD and RI. The UART Controller also provides a transmitter enable control signal pin option (TXDEN) to assist with interfacing to RS485 transceivers. RTS/CTS, DSR/DTR and XON / XOFF handshaking options are also supported. Handshaking is handled in hardware to ensure fast response times. The UART interface also supports the RS232 BREAK setting and detection conditions.

Additionally, the UART signals can each be individually inverted and have a configurable high drive strength capability. Both these features are configurable in the EEPROM.

Baud Rate Generator - The Baud Rate Generator provides a 16x clock input to the UART Controller from the 48MHz reference clock. It consists of a 14 bit pre-scaler and 3 register bits which provide fine tuning of the baud rate (used to divide by a number plus a fraction or "sub-integer"). This determines the baud rate of the UART, which is programmable from 183 baud to 3 Mbaud.

The FT232R supports all standard baud rates and non-standard baud rates from 183 Baud up to 3 Mbaud. Achievable non-standard baud rates are calculated as follows -

Baud Rate = 3000000 / (n + *x*)

where 'n' can be any integer between 2 and 16,384 (= 2^{14}) and '*x*' can be a sub-integer of the value 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, or 0.875. When n = 1, *x* = 0, i.e. baud rate divisors with values between 1 and 2 are not possible.

This gives achievable baud rates in the range 183.1 baud to 3,000,000 baud. When a non-standard baud rate is required simply pass the required baud rate value to the driver as normal, and the FTDI driver will calculate the required divisor, and set the baud rate. See FTDI application note AN232B-05 on the FTDI website (www.ftdichip.com) for more details.

RESET Generator - The integrated Reset Generator Cell provides a reliable power-on reset to the device internal circuitry at power up. The RESET# input pin allows an external device to reset the FT232R.

RESET# can be tied to VCC or left unconnected if not being used.

5 Devices Characteristics and Ratings

5.1 Absolute Maximum Ratings

The absolute maximum ratings for the FT232R devices are as follows. These are in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

Parameter	Value	Unit
Storage Temperature	-65°C to 150°C	Degrees C
Floor Life (Out of Bag) At Factory Ambient (30°C / 60% Relative Humidity)	168 Hours (IPC/JEDEC J-STD-033A MSL Level 3 Compliant)*	Hours
Ambient Temperature (Power Applied)	-40°C to 85°C	Degrees C
MTTF FT232RL	11162037	hours
MTTF FT232RQ	4464815	hours
VCC Supply Voltage	-0.5 to +6.00	V
DC Input Voltage – USBDP and USBDM	-0.5 to +3.8	V
DC Input Voltage – High Impedance Bidirectionals	-0.5 to + (VCC +0.5)	V
DC Input Voltage – All Other Inputs	-0.5 to + (VCC +0.5)	V
DC Output Current – Outputs	24	mA
DC Output Current – Low Impedance Bidirectionals	24	mA
Power Dissipation (VCC = 5.25V)	500	mW

Table 5.1 Absolute Maximum Ratings

* If devices are stored out of the packaging beyond this time limit the devices should be baked before use. The devices should be ramped up to a temperature of +125°C and baked for up to 17 hours.

5.2 DC Characteristics

DC Characteristics (Ambient Temperature = -40°C to +85°C)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCC1	VCC Operating Supply Voltage	4.0	---	5.25	V	Using Internal Oscillator
VCC1	VCC Operating Supply Voltage	3.3	---	5.25	V	Using External Crystal
VCC2	VCCIO Operating Supply Voltage	1.8	---	5.25	V	
Icc1	Operating Supply Current	---	15	---	mA	Normal Operation
Icc2	Operating Supply Current	50	70	100	µA	USB Suspend
3V3	3.3v regulator output	3.0	3.3	3.6	V	

Table 5.2 Operating Voltage and Current

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.3 UART and CBUS I/O Pin Characteristics (VCCIO = +5.0V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.2	2.7	3.2	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.4 UART and CBUS I/O Pin Characteristics (VCCIO = +3.3V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.1	2.6	2.8	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.5 UART and CBUS I/O Pin Characteristics (VCCIO = +2.8V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	1.32	1.62	1.8	V	I source = 0.2mA
Vol	Output Voltage Low	0.06	0.1	0.18	V	I sink = 0.5mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.6 UART and CBUS I/O Pin Characteristics (VCCIO = +1.8V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 6mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 6mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.7 UART and CBUS I/O Pin Characteristics (VCCIO = +5.0V, High Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.2	2.8	3.2	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.8 UART and CBUS I/O Pin Characteristics (VCCIO = +3.3V, High Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.1	2.6	2.8	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.9 UART and CBUS I/O Pin Characteristics (VCCIO = +2.8V, High Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	1.35	1.67	1.8	V	I source = 0.4mA
Vol	Output Voltage Low	0.12	0.18	0.35	V	I sink = 3mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.10 UART and CBUS I/O Pin Characteristics (VCCIO = +1.8V, High Drive Level)

** Only input pins have an internal 200KΩ pull-up resistor to VCCIO

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Vin	Input Switching Threshold	1.3	1.6	1.9	V	
VHys	Input Switching Hysteresis	50	55	60	mV	

Table 5.11 RESET# and TEST Pin Characteristics

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
UVoh	I/O Pins Static Output (High)	2.8		3.6	V	RI = 1.5kΩ to 3V3OUT (D+) RI = 15KΩ to GND (D-)
UVol	I/O Pins Static Output (Low)	0		0.3	V	RI = 1.5kΩ to 3V3OUT (D+) RI = 15kΩ to GND (D-)
UVse	Single Ended Rx Threshold	0.8		2.0	V	
UCom	Differential Common Mode	0.8		2.5	V	
UVDif	Differential Input Sensitivity	0.2			V	
UDrvZ	Driver Output Impedance	26	29	44	Ohms	See Note 1

Table 5.12 USB I/O Pin (USBDP, USBDM) Characteristics

5.3 EEPROM Reliability Characteristics

The internal 1024 Bit EEPROM has the following reliability characteristics:

Parameter	Value	Unit
Data Retention	10	Years
Read / Write Cycle	10,000	Cycles

Table 5.13 EEPROM Characteristics

5.4 Internal Clock Characteristics

The internal Clock Oscillator has the following characteristics:

Parameter	Value			Unit
	Minimum	Typical	Maximum	
Frequency of Operation (see Note 1)	11.98	12.00	12.02	MHz
Clock Period	83.19	83.33	83.47	ns
Duty Cycle	45	50	55	%

Table 5.14 Internal Clock Characteristics

Note 1: Equivalent to +/-1667ppm

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.1	2.8	3.2	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	

Table 5.15 OSCI, OSCO Pin Characteristics – see Note 1

Note1: When supplied, the FT232R is configured to use its internal clock oscillator. These characteristics only apply when an external oscillator or crystal is used.

6 USB Power Configurations

The following sections illustrate possible USB power configurations for the FT232R. The illustrations have omitted pin numbers for ease of understanding since the pins differ between the FT232RL and FT232RQ package options.

All USB power configurations illustrated apply to both package options for the FT232R device. Please refer to Section 3 for the package option pin-out and signal descriptions.

6.1 USB Bus Powered Configuration

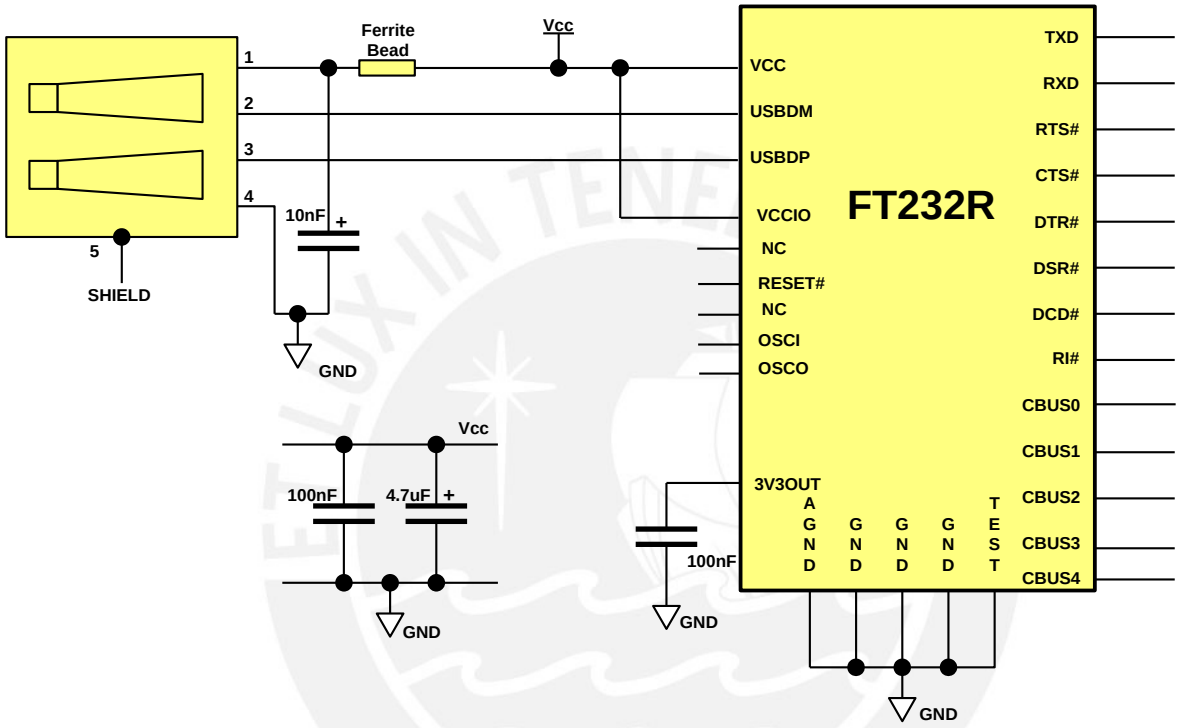


Figure 6.1 Bus Powered Configuration

Figure 6.1 Illustrates the FT232R in a typical USB bus powered design configuration. A USB bus powered device gets its power from the USB bus. Basic rules for USB bus power devices are as follows –

i) On plug-in to USB, the device should draw no more current than 100mA.
ii) In USB Suspend mode the device should draw no more than 2.5mA.
iii) A bus powered high power USB device (one that draws more than 100mA) should use one of the CBUS pins configured as PWREN# and use it to keep the current below 100mA on plug-in and 2.5mA on USB suspend.
iv) A device that consumes more than 100mA cannot be plugged into a USB bus powered hub.
v) No device can draw more than 500mA from the USB bus.

The power descriptors in the internal EEPROM of the FT232R should be programmed to match the current drawn by the device.

A ferrite bead is connected in series with the USB power supply to reduce EMI noise from the FT232R and associated circuitry being radiated down the USB cable to the USB host. The value of the Ferrite Bead depends on the total current drawn by the application. A suitable range of Ferrite Beads is available from Steward (www.steward.com), for example Steward Part # MI0805K400R-10.

Note: If using PWREN# (available using the CBUS) the pin should be pulled to VCCIO using a 10kΩ resistor.

6.2 Self Powered Configuration

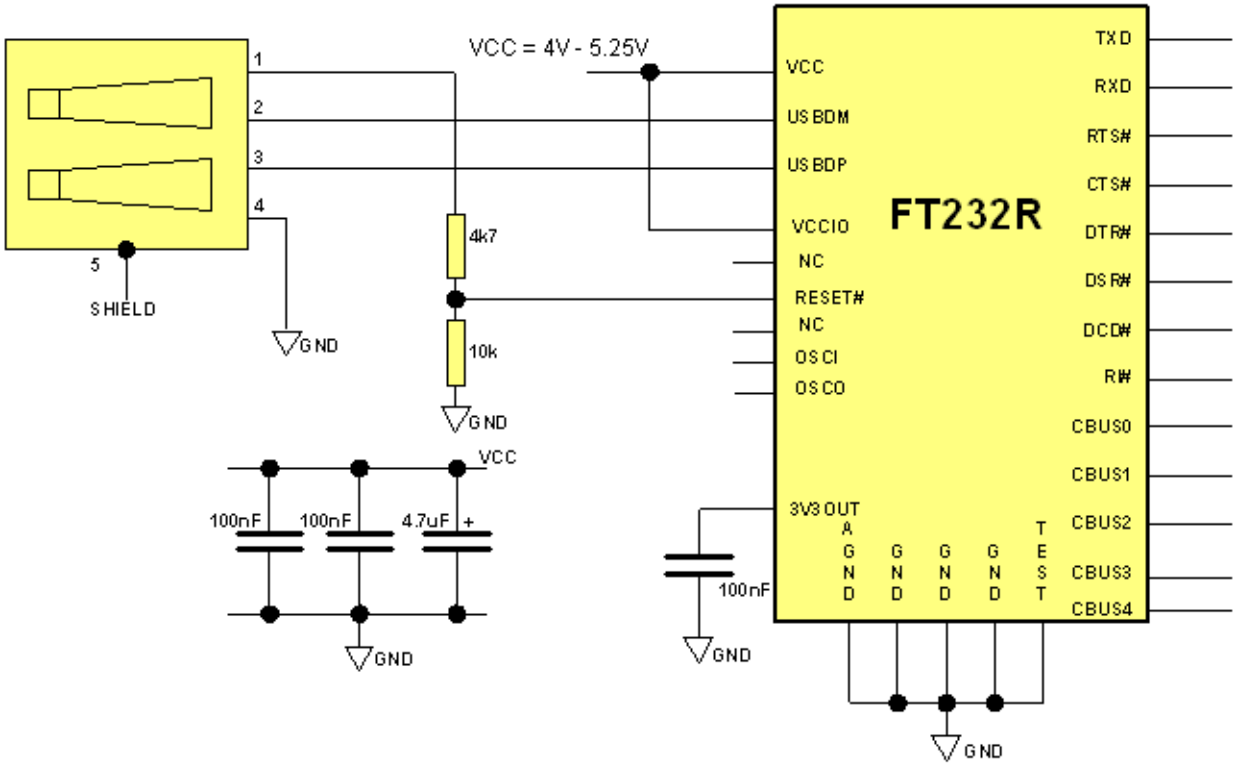


Figure 6.2 Self Powered Configuration

Figure 6.2 illustrates the FT232R in a typical USB self powered configuration. A USB self powered device gets its power from its own power supply, VCC, and does not draw current from the USB bus. The basic rules for USB self powered devices are as follows –

i) A self powered device should not force current down the USB bus when the USB host or hub controller is powered down.
ii) A self powered device can use as much current as it needs during normal operation and USB suspend as it has its own power supply.
iii) A self powered device can be used with any USB host, a bus powered USB hub or a self powered USB hub.

The power descriptor in the internal EEPROM of the FT232R should be programmed to a value of zero (self powered).

In order to comply with the first requirement above, the USB bus power (pin 1) is used to control the RESET# pin of the FT232R device. When the USB host or hub is powered up an internal 1.5kΩ resistor on USBDP is pulled up to +3.3V (generated using the 4K7 and 10k resistor network), thus identifying the device as a full speed device to the USB host or hub. When the USB host or hub is powered off, RESET# will be low and the FT232R is held in reset. Since RESET# is low, the internal 1.5kΩ resistor is not pulled up to any power supply (hub or host is powered down), so no current flows down USBDP via the 1.5kΩ pull-up resistor. Failure to do this may cause some USB host or hub controllers to power up erratically.

Figure 6.2 illustrates a self powered design which has a +4V to +5.25V supply.

Note:

1. When the FT232R is in reset, the UART interface I/O pins are tri-stated. Input pins have internal 200kΩ pull-up resistors to VCCIO, so they will gently pull high unless driven by some external logic.
2. When using internal FT232R oscillator the VCC supply voltage range must be +4.0V to 5.25V.
3. When using external oscillator the VCC supply voltage range must be +3.3V to 5.25V
   Any design which interfaces to +3.3 V or +1.8V would be having a +3.3V or +1.8V supply to VCCIO.

6.3 USB Bus Powered with Power Switching Configuration

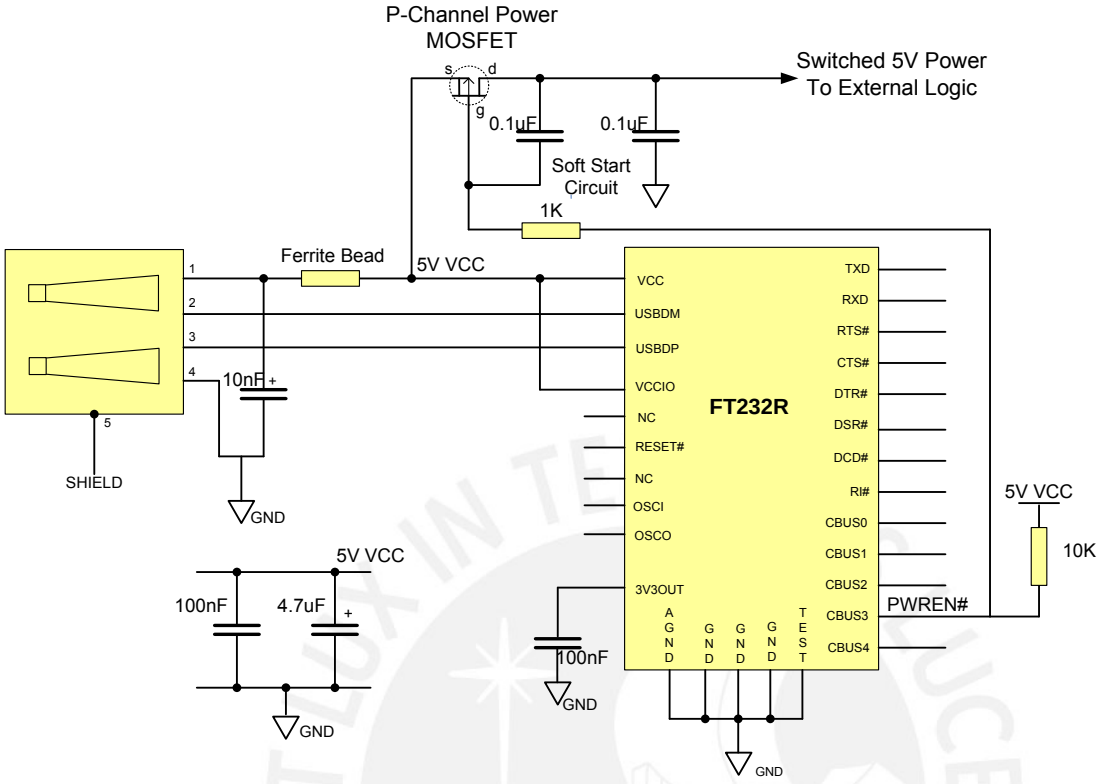


Figure 6.3 Bus Powered with Power Switching Configuration

A requirement of USB bus powered applications, is when in USB suspend mode, the application draws a total current of less than 2.5mA. This requirement includes external logic. Some external logic has the ability to power itself down into a low current state by monitoring the PWREN# signal. For external logic that cannot power itself down in this way, the FT232R provides a simple but effective method of turning off power during the USB suspend mode.

Figure 6.3 shows an example of using a discrete P-Channel MOSFET to control the power to external logic. A suitable device to do this is an International Rectifier (www.irf.com) IRLML6402, or equivalent. It is recommended that a "soft start" circuit consisting of a 1kΩ series resistor and a 0.1µF capacitor is used to limit the current surge when the MOSFET turns on. Without the soft start circuit it is possible that the transient power surge, caused when the MOSFET switches on, will reset the FT232R or the USB host/hub controller. The soft start circuit example shown in Figure 6.3 powers up with a slew rate of approximaely12.5V/ms. Thus supply voltage to external logic transitions from GND to +5V in approximately 400 microseconds.

As an alternative to the MOSFET, a dedicated power switch IC with inbuilt "soft-start" can be used. A suitable power switch IC for such an application is the Micrel (www.micrel.com) MIC2025-2BM or equivalent.

With power switching controlled designs the following should be noted:

i) The external logic to which the power is being switched should have its own reset circuitry to automatically reset the logic when power is re-applied when moving out of suspend mode.

ii) Set the Pull-down on Suspend option in the internal FT232R EEPROM.

iii) One of the CBUS Pins should be configured as PWREN# in the internal FT232R EEPROM, and used to switch the power supply to the external circuitry. This should be pulled high through a 10 kΩ resistor.

iv) For USB high-power bus powered applications (one that consumes greater than 100mA, and up to 500mA of current from the USB bus), the power consumption of the application must be set in the Max Power field in the internal FT232R EEPROM. A high-power bus powered application uses the descriptor in the internal FT232R EEPROM to inform the system of its power requirements.

v) PWREN# gets its VCC from VCCIO. For designs using 3V3 logic, ensure VCCIO is not powered down using the external logic. In this case use the +3V3OUT.

6.4 USB Bus Powered with Selectable External Logic Supply

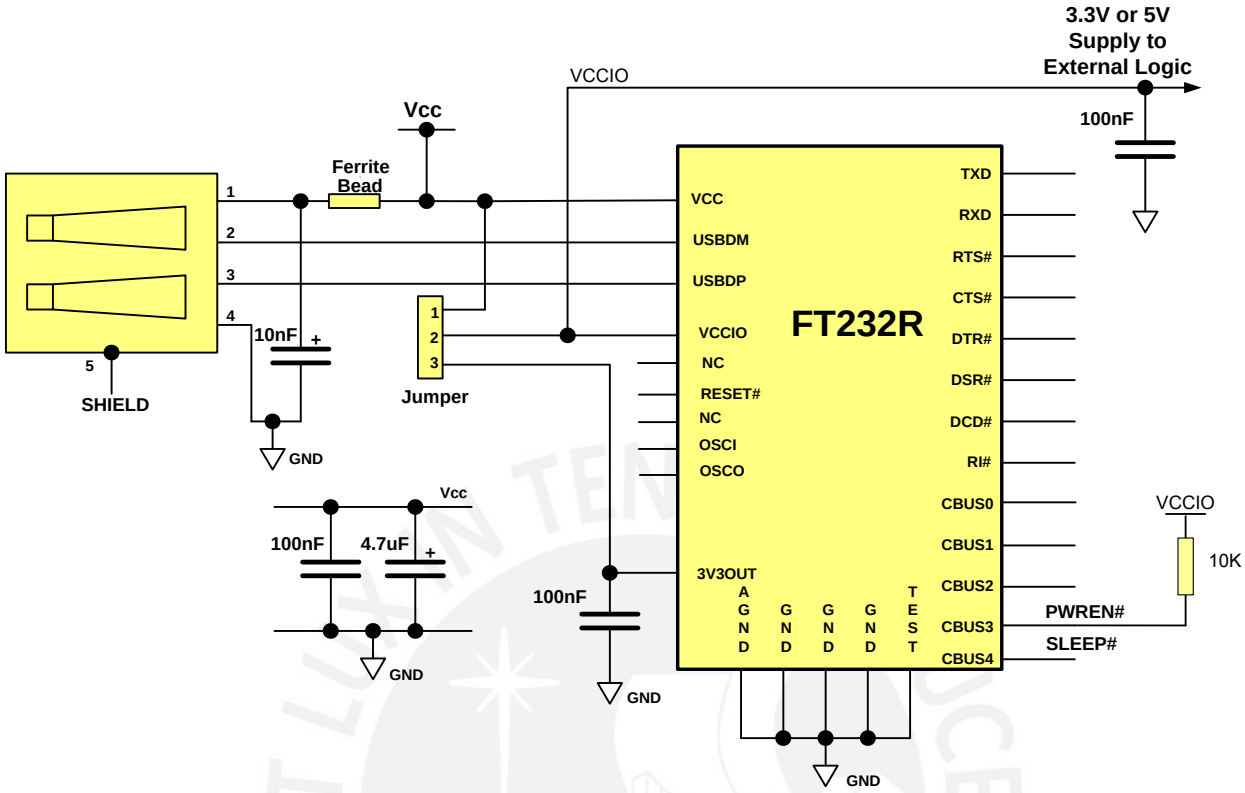


Figure 6.4 USB Bus Powered with +3.3V or +5V External Logic Power Supply

Figure 6.4 illustrates a USB bus power application with selectable external logic supply. The external logic can be selected between +3.3V and +5V using the jumper switch. This jumper is used to allow the FT232R to be interfaced with a +3.3V or +5V logic devices. The VCCIO pin is either supplied with +5V from the USB bus (jumper pins1 and 2 connected), or from the +3.3V output from the FT232R 3V3OUT pin (jumper pins 2 and 3 connected). The supply to VCCIO is also used to supply external logic.

With bus powered applications, the following should be noted:

i) To comply with the 2.5mA current supply limit during USB suspend mode, PWREN# or SLEEP# signals should be used to power down external logic in this mode. If this is not possible, use the configuration shown in Section 6.3.

ii) The maximum current sourced from the USB bus during normal operation should not exceed 100mA, otherwise a bus powered design with power switching (Section 6.3) should be used.

Another possible configuration could use a discrete low dropout (LDO) regulator which is supplied by the 5V on the USB bus to supply between +1.8V and +2.8V to the VCCIO pin and to the external logic. In this case VCC would be supplied with the +5V from the USB bus and the VCCIO would be supplied from the output of the LDO regulator. This results in the FT232R I/O pins driving out at between +1.8V and +2.8V logic levels.

For a USB bus powered application, it is important to consider the following when selecting the regulator:

i) The regulator must be capable of sustaining its output voltage with an input voltage of +4.35V. An Low Drop Out (LDO) regulator should be selected.

ii) The quiescent current of the regulator must be low enough to meet the total current requirement of <= 2.5mA during USB suspend mode.

A suitable series of LDO regulators that meets these requirements is the MicroChip/Telcom (www.microchip.com) TC55 series of devices. These devices can supply up to 250mA current and have a quiescent current of under 1µA.

7 Application Examples

The following sections illustrate possible applications of the FT232R. The illustrations have omitted pin numbers for ease of understanding since the pins differ between the FT232RL and FT232RQ package options.

7.1 USB to RS232 Converter

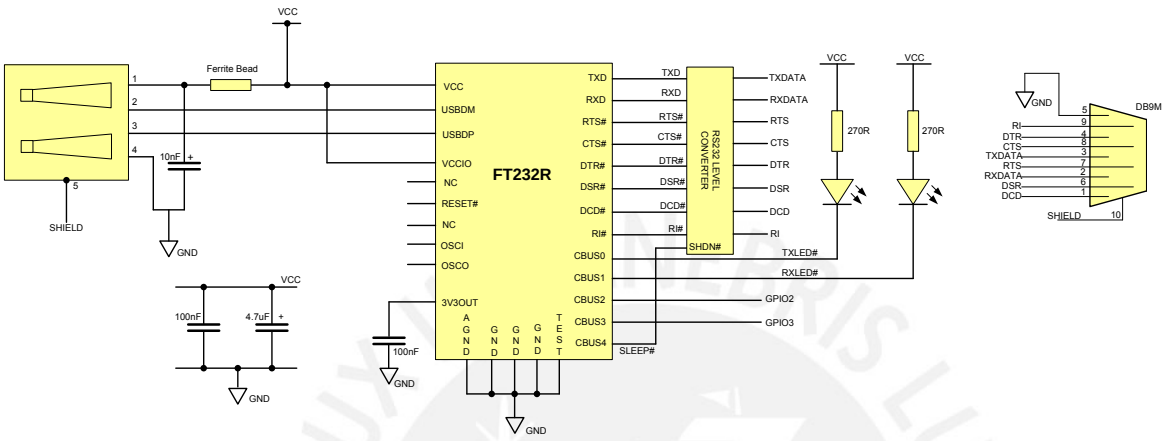


Figure 7.1 Application Example showing USB to RS232 Converter

An example of using the FT232R as a USB to RS232 converter is illustrated in Figure 7.1. In this application, a TTL to RS232 Level Converter IC is used on the serial UART interface of the FT232R to convert the TTL levels of the FT232R to RS232 levels. This level shift can be done using the popular "213" series of TTL to RS232 level converters. These "213" devices typically have 4 transmitters and 5 receivers in a 28-LD SSOP package and feature an in-built voltage converter to convert the +5V (nominal) VCC to the +/- 9 volts required by RS232. A useful feature of these devices is the SHDN# pin which can be used to power down the device to a low quiescent current during USB suspend mode.

A suitable level shifting device is the Sipex SP213EHCA which is capable of RS232 communication at up to 500k baud. If a lower baud rate is acceptable, then several pin compatible alternatives are available such as the Sipex SP213ECA, the Maxim MAX213CAI and the Analogue Devices ADM213E, which are all suitable for communication at up to 115.2k baud. If a higher baud rate is required, the Maxim MAX3245CAI device is capable of RS232 communication rates up to 1Mbaud. Note that the MAX3245 is not pin compatible with the 213 series devices and that the SHDN pin on the MAX device is active high and should be connect to PWREN# pin instead of SLEEP# pin.

In example shown, the CBUS0 and CBUS1 have been configured as TXLED# and RXLED# and are being used to drive two LEDs.

7.2 USB to RS485 Coverter

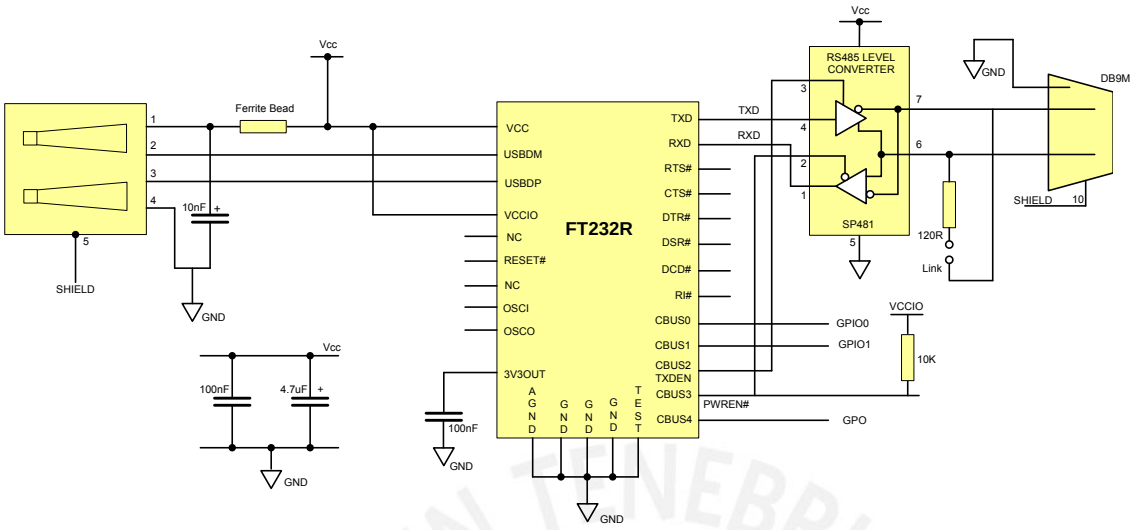


Figure 7.2 Application Example Showing USB to RS485 Converter

An example of using the FT232R as a USB to RS485 converter is shown in Figure 7.2. In this application, a TTL to RS485 level converter IC is used on the serial UART interface of the FT232R to convert the TTL levels of the FT232R to RS485 levels.

This example uses the Sipex SP481 device. Equivalent devices are available from Maxim and Analogue Devices. The SP481 is a RS485 device in a compact 8 pin SOP package. It has separate enables on both the transmitter and receiver. With RS485, the transmitter is only enabled when a character is being transmitted from the UART. The TXDEN signal CBUS pin option on the FT232R is provided for exactly this purpose and so the transmitter enable is wired to CBUS2 which has been configured as TXDEN. Similarly, CBUS3 has been configured as PWREN#. This signal is used to control the SP481's receiver enable. The receiver enable is active low, so it is wired to the PWREN# pin to disable the receiver when in USB suspend mode. CBUS2 = TXDEN and CBUS3 = PWREN# are the default device configurations of the FT232R pins.

RS485 is a multi-drop network; so many devices can communicate with each other over a two wire cable interface. The RS485 cable requires to be terminated at each end of the cable. A link (which provides the 120Ω termination) allows the cable to be terminated if the SP481 is physically positioned at either end of the cable.

In this example the data transmitted by the FT232R is also present on the receive path of the SP481.This is a common feature of RS485 and requires the application software to remove the transmitted data from the received data stream. With the FT232R it is possible to do this entirely in hardware by modifying the example shown in Figure 7.2 by logically OR'ing the FT232R TXDEN and the SP481 receiver output and connecting the output of the OR gate to the RXD of the FT232R.

Note that the TXDEN is activated 1 bit period before the start bit. TXDEN is deactivated at the same time as the stop bit. This is not configurable.

7.3 USB to RS422 Converter

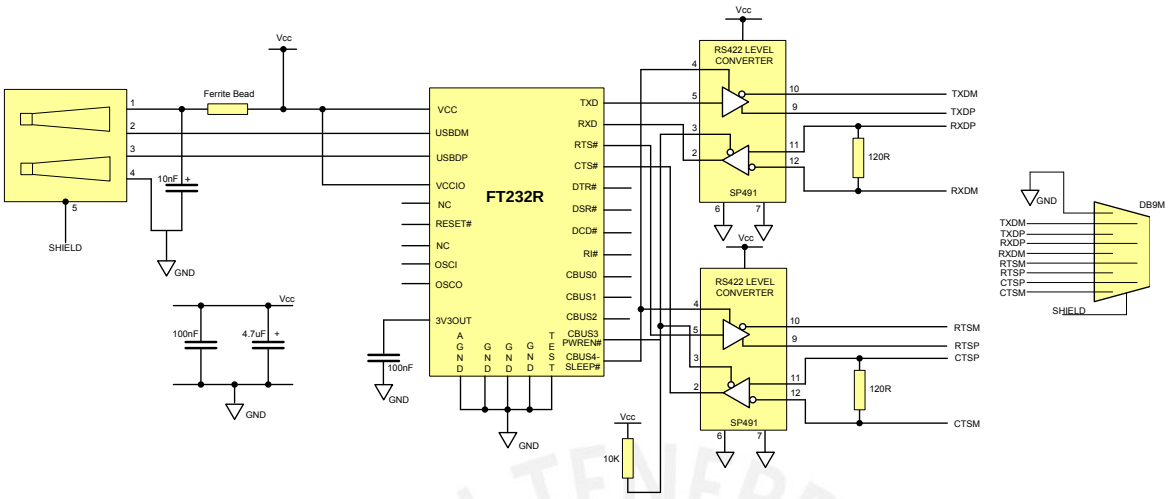


Figure 7.3 USB to RS422 Converter Configuration

An example of using the FT232R as a USB to RS422 converter is shown in Figure 7.3. In this application, two TTL to RS422 Level Converter ICs are used on the serial UART interface of the FT232R to convert the TTL levels of the FT232R to RS422 levels. There are many suitable level converter devices available. This example uses Sipex SP491 devices which have enables on both the transmitter and receiver. Since the SP491 transmitter enable is active high, it is connected to a CBUS pin in SLEEP# configuration. The SP491 receiver enable is active low and is therefore connected to a CBUS pin PWREN# configuration. This ensures that when both the SP491 transmitters and receivers are enabled then the device is active, and when the device is in USB suspend mode, the SP491 transmitters and receivers are disabled. If a similar application is used, but the design is USB BUS powered, it may be necessary to use a P-Channel logic level MOSFET (controlled by PWREN#) in the VCC line of the SP491 devices to ensure that the USB standby current of 2.5mA is met.

The SP491 is specified to transmit and receive data at a rate of up to 5 Mbaud. In this example the maximum data rate is limited to 3 Mbaud by the FT232R.

7.4 USB to MCU UART Interface

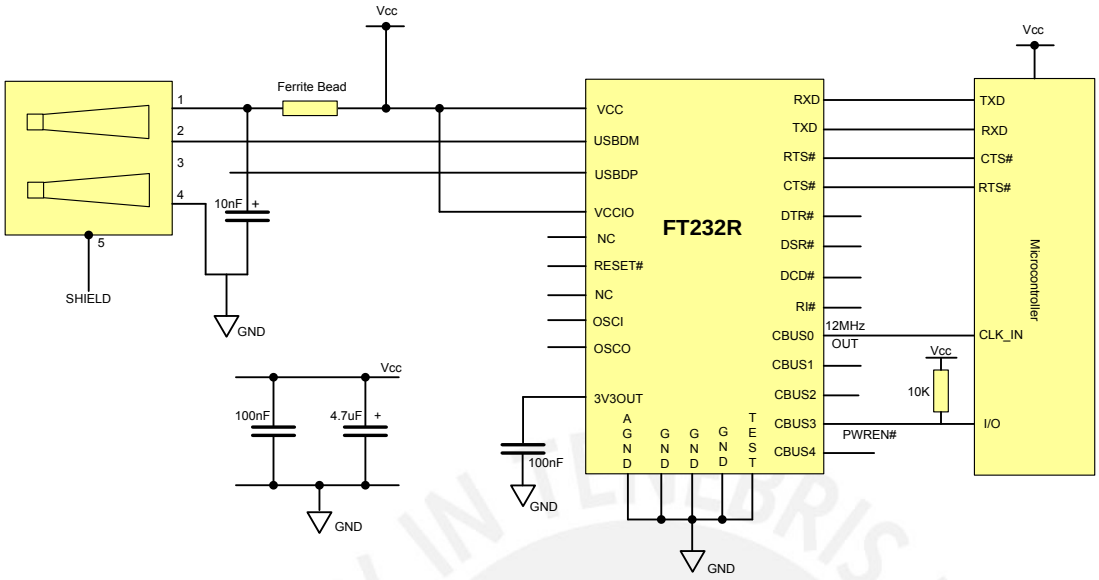


Figure 7.4 USB to MCU UART Interface

An example of using the FT232R as a USB to Microcontroller (MCU) UART interface is shown in Figure 7.4. In this application the FT232R uses TXD and RXD for transmission and reception of data, and RTS# / CTS# signals for hardware handshaking. Also in this example CBUS0 has been configured as a 12MHz output to clock the MCU.

Optionally, RI# could be connected to another I/O pin on the MCU and used to wake up the USB host controller from suspend mode. If the MCU is handling power management functions, then a CBUS pin can be configured as PWREN# and would also be connected to an I/O pin of the MCU.

7.5 LED Interface

Any of the CBUS I/O pins can be configured to drive an LED. The FT232R has 3 configuration options for driving LEDs from the CBUS. These are TXLED#, RXLED#, and TX&RXLED#. Refer to Section 3.5 for configuration options.

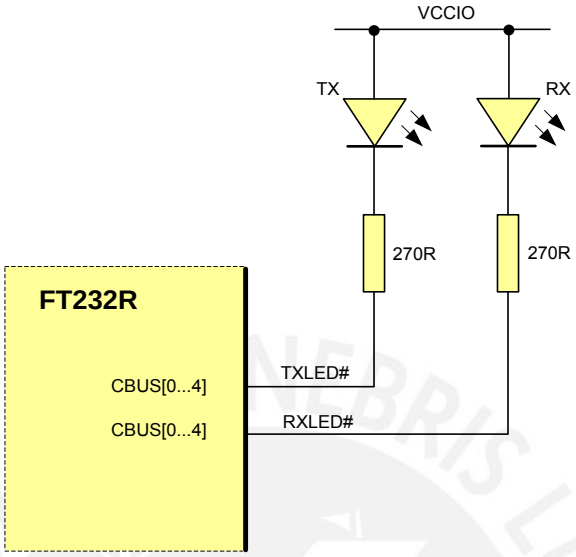


Figure 7.5 Dual LED Configuration

An example of using the FT232R to drive LEDs is shown in Figure 7.5. In this application one of the CBUS pins is used to indicate transmission of data (TXLED#) and another is used to indicate receiving data (RXLED#). When data is being transmitted or received the respective pins will drive from tri-state to low in order to provide indication on the LEDs of data transfer. A digital one-shot is used so that even a small percentage of data transfer is visible to the end user.

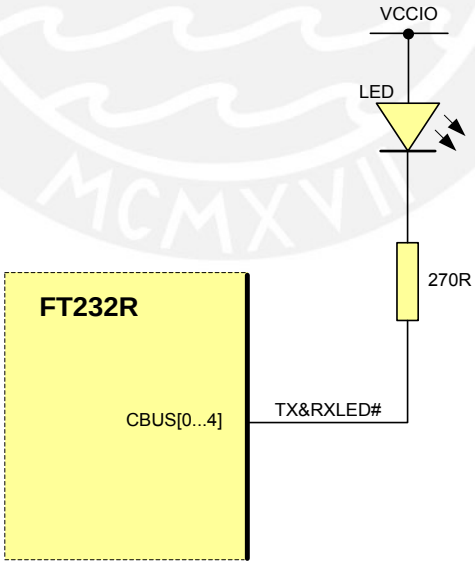


Figure 7.6 Single LED Configuration

Another example of using the FT232R to drive LEDs is shown in Figure 7.6. In this example one of the CBUS pins is used to indicate when data is being transmitted or received by the device (TX&RXLED). In this configuration the FT232R will drive only a single LED.

7.6 Using the External Oscillator

The FT232R defaults to operating using its own internal oscillator. This requires that the device is powered with VCC(min)=+4.0V. This supply voltage can be taken from the USB VBUS. Applications which require using an external oscillator, VCC= +3.3V, must do so in the following order:

1. When device powered for the very first time, it must have VCC > +4.0V. This supply is available from the USB VBUS supply = +5.0V.
2. The EEPROM must then be programmed to enable external oscillator. This EEPROM modification cannot be done using the FTDI programming utility, MPROG. The EEPROM can only be re-configured from a custom application. Please refer to the following applications note on how to do this:

   http://www.ftdichip.com/Documents/AppNotes/AN_100_Using_The_FT232_245R_With_External_Osc(FT_000067).pdf

3. The FT232R can then be powered from VCC=+3.3V and an external oscillator. This can be done using a link to switch the VCC supply.

The FT232R will fail to operate when the internal oscillator has been disabled, but no external oscillator has been connected.

8 Internal EEPROM Configuration

Following a power-on reset or a USB reset the FT232R will scan its internal EEPROM and read the USB configuration descriptors stored there. The default factory programmed values of the internal EEPROM are shown in Table 8.1.

Parameter	Value	Notes
USB Vendor ID (VID)	0403h	FTDI default VID (hex)
USB Product UD (PID)	6001h	FTDI default PID (hex)
Serial Number Enabled?	Yes	
Serial Number	See Note	A unique serial number is generated and programmed into the EEPROM during device final test.
Pull down I/O Pins in USB Suspend	Disabled	Enabling this option will make the device pull down on the UART interface lines when in USB suspend mode (PWREN# is high).
Manufacturer Name	FTDI	
Product Description	FT232R USB UART	
Max Bus Power Current	90mA	
Power Source	Bus Powered	
Device Type	FT232R	
USB Version	0200	Returns USB 2.0 device description to the host. Note: The device is a USB 2.0 Full Speed device (12Mb/s) as opposed to a USB 2.0 High Speed device (480Mb/s).
Remote Wake Up	Enabled	Taking RI# low will wake up the USB host controller from suspend in approximately 20 ms.
High Current I/Os	Disabled	Enables the high drive level on the UART and CBUS I/O pins.
Load VCP Driver	Enabled	Makes the device load the VCP driver interface for the device.
CBUS0	TXLED#	Default configuration of CBUS0 – Transmit LED drive.
CBUS1	RXLED#	Default configuration of CBUS1 – Receive LED drive.
CBUS2	TXDEN	Default configuration of CBUS2 – Transmit data enable for RS485
CBUS3	PWREN#	Default configuration of CBUS3 – Power enable. Low after USB enumeration, high during USB suspend mode.

Parameter	Value	Notes
CBUS4	SLEEP#	Default configuration of CBUS4 – Low during USB suspend mode.
Invert TXD	Disabled	Signal on this pin becomes TXD# if enable.
Invert RXD	Disabled	Signal on this pin becomes RXD# if enable.
Invert RTS#	Disabled	Signal on this pin becomes RTS if enable.
Invert CTS#	Disabled	Signal on this pin becomes CTS if enable.
Invert DTR#	Disabled	Signal on this pin becomes DTR if enable.
Invert DSR#	Disabled	Signal on this pin becomes DSR if enable.
Invert DCD#	Disabled	Signal on this pin becomes DCD if enable.
Invert RI#	Disabled	Signal on this pin becomes RI if enable.

Table 8.1 Default Internal EEPROM Configuration

The internal EEPROM in the FT232R can be programmed over USB using the FTDI utility program MPROG. MPROG can be downloaded from FTDI Utilities on the FTDI website (www.ftdichip.com). Version 2.8a or later is required for the FT232R chip. Users who do not have their own USB Vendor ID but who would like to use a unique Product ID in their design can apply to FTDI for a free block of unique PIDs. Contact FTDI support for this service.

9 Package Parameters

The FT232R is available in two different packages. The FT232RL is the SSOP-28 option and the FT232RQ is the QFN-32 package option. The solder reflow profile for both packages is described in Section 9.5.

9.1 SSOP-28 Package Dimensions

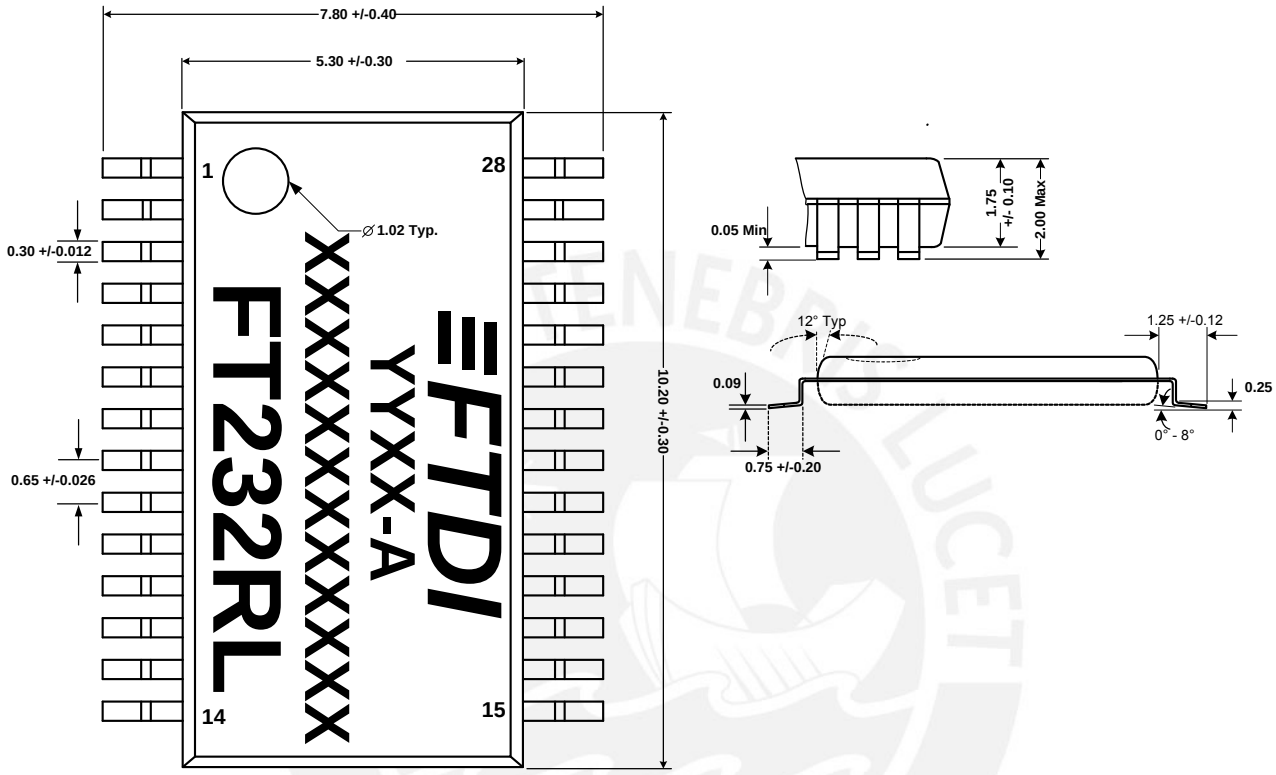


Figure 9.1 SSOP-28 Package Dimensions

The FT232RL is supplied in a RoHS compliant 28 pin SSOP package. The package is lead (Pb) free and uses a 'green' compound. The package is fully compliant with European Union directive 2002/95/EC.

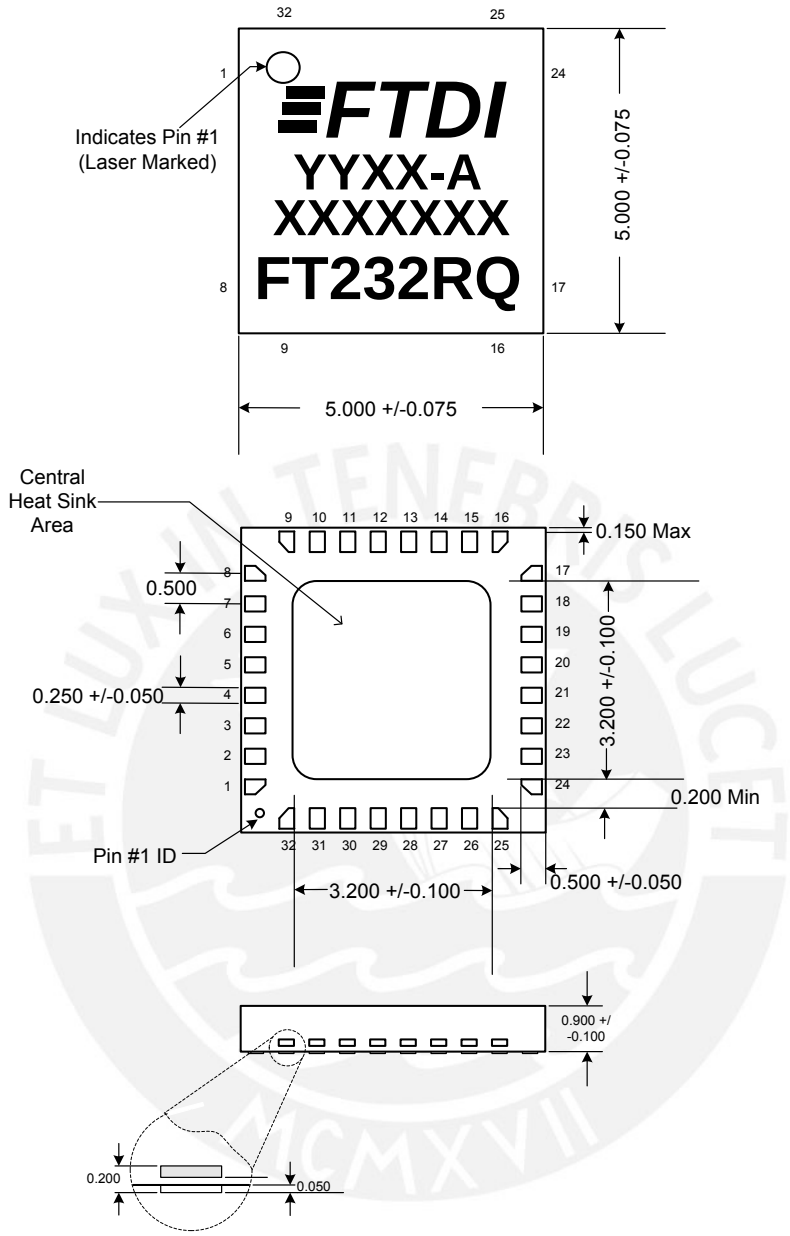
This package is nominally 5.30mm x 10.20mm body (7.80mm x 10.20mm including pins). The pins are on a 0.65 mm pitch. The above mechanical drawing shows the SSOP-28 package.

All dimensions are in millimetres.

The date code format is **YYXX** where XX = 2 digit week number, YY = 2 digit year number. This is followed by the revision number.

The code **XXXXXXXXXXX** is the manufacturing LOT code. This only applies to devices manufactured after April 2009.

9.2 QFN-32 Package Dimensions



Note: The pin #1 ID is connected internally to the device's central heat sink area . It is recommended to ground the central heat sink area of the device.

Dimensions in mm.

Figure 9.2 QFN-32 Package Dimensions

The FT232RQ is supplied in a RoHS compliant leadless QFN-32 package. The package is lead (Pb) free, and uses a 'green' compound. The package is fully compliant with European Union directive 2002/95/EC.

This package is nominally 5.00mm x 5.00mm. The solder pads are on a 0.50mm pitch. The above mechanical drawing shows the QFN-32 package. All dimensions are in millimetres.

The centre pad on the base of the FT232RQ is not internally connected, and can be left unconnected, or connected to ground (recommended).

The date code format is **YYXX** where XX = 2 digit week number, YY = 2 digit year number.

The code **XXXXXXX** is the manufacturing LOT code. This only applies to devices manufactured after April 2009.

9.3 QFN-32 Package Typical Pad Layout

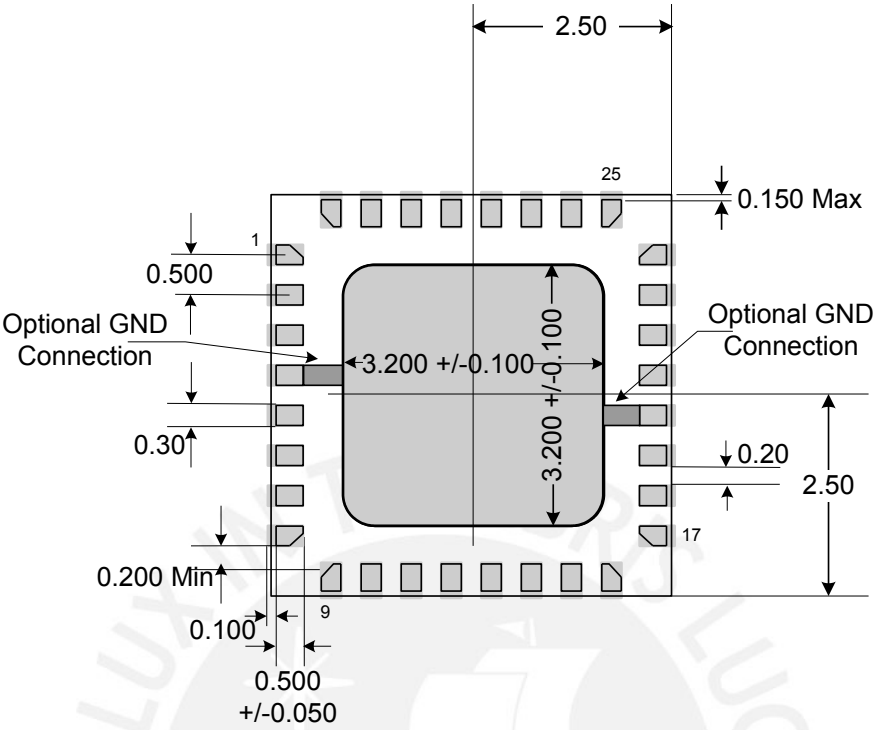


Figure 9.3 Typical Pad Layout for QFN-32 Package

9.4 QFN-32 Package Typical Solder Paste Diagram

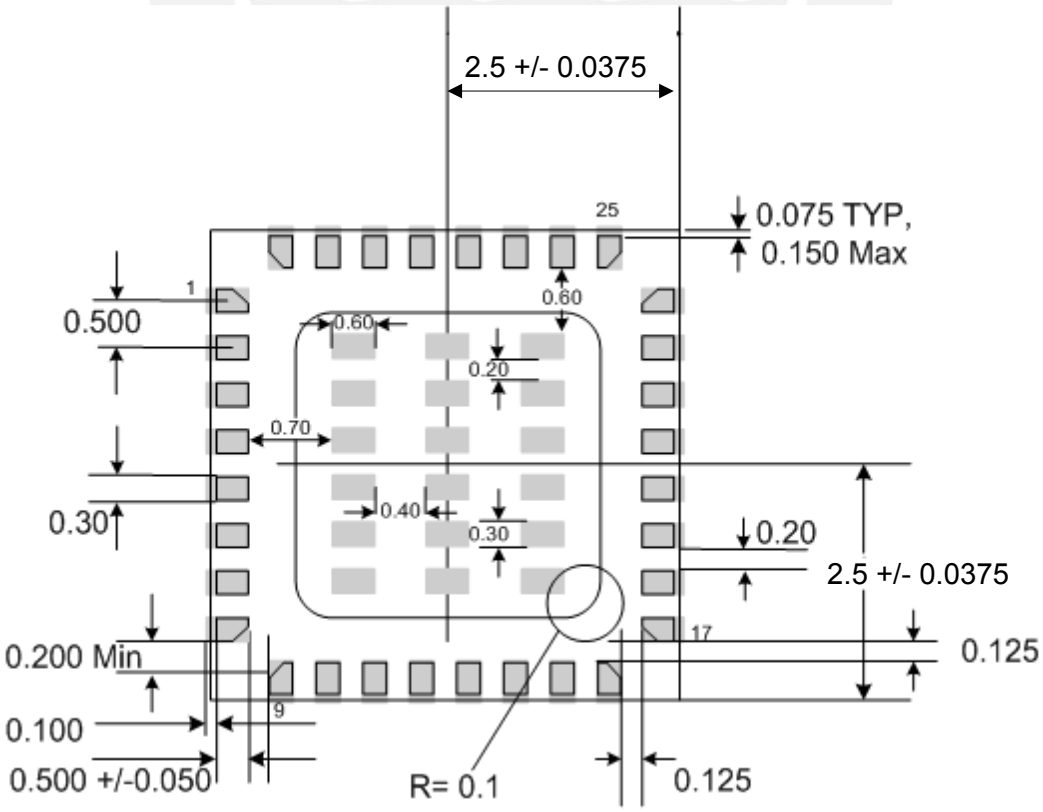


Figure 9.4 Typical Solder Paste Diagram for QFN-32 Package

9.5 Solder Reflow Profile

The FT232R is supplied in Pb free 28 LD SSOP and QFN-32 packages. The recommended solder reflow profile for both package options is shown in Figure 9.5.

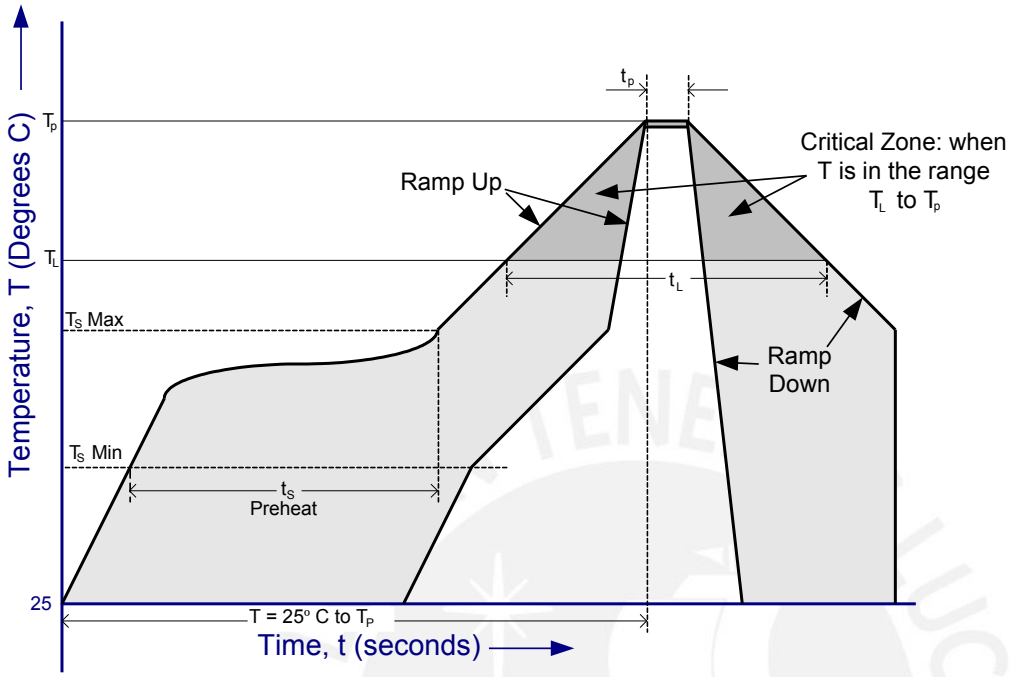


Figure 9.5 FT232R Solder Reflow Profile

The recommended values for the solder reflow profile are detailed in Table 9.1. Values are shown for both a completely Pb free solder process (i.e. the FT232R is used with Pb free solder), and for a non-Pb free solder process (i.e. the FT232R is used with non-Pb free solder).

Profile Feature	Pb Free Solder Process	Non-Pb Free Solder Process
Average Ramp Up Rate (T_s to T_p)	3°C / second Max.	3°C / Second Max.
Preheat - Temperature Min (T_s Min.) - Temperature Max (T_s Max.) - Time (t_s Min to t_s Max)	 150°C 200°C 60 to 120 seconds	 100°C 150°C 60 to 120 seconds
Time Maintained Above Critical Temperature T_L: - Temperature (T_L) - Time (t_L)	 217°C 60 to 150 seconds	 183°C 60 to 150 seconds
Peak Temperature (T_p)	260°C	240°C
Time within 5°C of actual Peak Temperature (t_p)	20 to 40 seconds	20 to 40 seconds
Ramp Down Rate	6°C / second Max.	6°C / second Max.
Time for T= 25°C to Peak Temperature, T_p	8 minutes Max.	6 minutes Max.

Table 9.1 Reflow Profile Parameter Values

10 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place
Centurion Business Park
Glasgow, G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com
Web Site URL http://www.ftdichip.com
Web Shop URL http://www.ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No 516, Sec. 1 NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Hillsboro, Oregon, USA

Future Technology Devices International Limited (USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 408, 317 Xianxia Road,
ChangNing District,
ShangHai, China

Tel: +86 (21) 62351596
Fax: +86(21) 62351595

E-Mail (Sales): cn.sales@ftdichip.com
E-Mail (Support): cn.support@ftdichip.com
E-Mail (General Enquiries): cn.admin1@ftdichip.com

Web Site URL: http://www.ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

Useful Application Notes

http://www.ftdichip.com/Documents/AppNotes/AN232R-01_FT232RBitBangModes.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_107_AdvancedDriverOptions_AN_000073.pdf

http://www.ftdichip.com/Documents/AppNotes/AN232R-02_FT232RChipID.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_121_FTDI_Device_EEPROM_User_Area_Usage.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_120_Aliasing_VCP_Baud_Rates.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_100_Using_The_FT232_245R_With_External_Osc(FT_000067).pdf

http://www.ftdichip.com/Resources/Utilities/AN_126_User_Guide_For_FT232_Factory%20test%20utility.pdf

http://www.ftdichip.com/Documents/AppNotes/AN232B-05_BaudRates.pdf

http://www.ftdichip.com/Documents/InstallGuides.htm

Appendix B - List of Figures and Tables

List of Figures

List of Tables

Appendix C - Revision History

Document Title:	USB UART IC FT232R
Document Reference No.:	FT_000053
Clearance No.:	FTDI# 38
Product Page:	http://www.ftdichip.com/FTProducts.htm
Document Feedback:	Send Feedback

Version 0.90	Initial Datasheet Created	August 2005
Version 0.96	Revised Pre-release datasheet	October 2005
Version 1.00	Full datasheet released	December 2005
Version 1.02	Minor revisions to datasheet	December 2005
Version 1.03	Manufacturer ID added to default EEPROM configuration; Buffer sizes added	January 2006
Version 1.04	QFN-32 Pad layout and solder paste diagrams added	January 2006
Version 2.00	Reformatted, updated package info, added notes for 3.3V operation; Part numbers, TID; added UART and CBUS characteristics for +1.8V; Corrected RESET#; Added MTTF data; Corrected the input switching threshold and input hysteresis values for VCCIO=5V	June 2008
Version 2.01	Corrected pin-out number in table3.2 for GND pin18. Improved graphics on some Figures. Add packing details. Changed USB suspend current spec from 500uA to 2.5mA Corrected Figure 9.2 QFN dimensions.	August 2008
Version 2.02	Corrected Tape and Reel quantities. Added comment "PWREN# should be used with a 10kΩ resistor pull up". Replaced TXDEN# with TXDEN since it is active high in various places. Added lot number to the device markings. Added 3V3 regulator output tolerance. Clarified VCC operation and added section headed "Using an external Oscillator" Updated company contact information.	April 2009
Version 2.03	Corrected the RX/TX buffer definitions to be relative to the USB interface	June 2009
Version 2.04	Additional dimensions added to QFN solder profile	June 2009
Version 2.05	Modified package dimensions to 5.0 x 5.0 +/-0.075mm. and Solder paste diagram to 2.50 x 2.50 +/-0.0375mm Added Windows 7 32, 64 bit driver support Added FT_PROG utility references Added Appendix A-references.Figure 2.1 updated. Updated USB-IF TID for Rev B	December 2009
Version 2.06	Updated section 6.2, Figure 6.2 and the note, Updated section 5.3, Table 5.13, EEPROM data retention time	May 2010
Version 2.07	Added USB Certification Logos	July 2010
Version 2.08	Updated USB-IF TID for Rev C	April 2011
Version 2.09	Corrected Rev C TID number	April 2011
Version 2.10	Table 3.9, added clock output frequency within ±0.7% Edited Table 3.9, TXLED# and TXLED# Description Added feedback links	March 2012

TECHNICAL INFORMATION

SICK

DT20 Hi

Distance Sensor

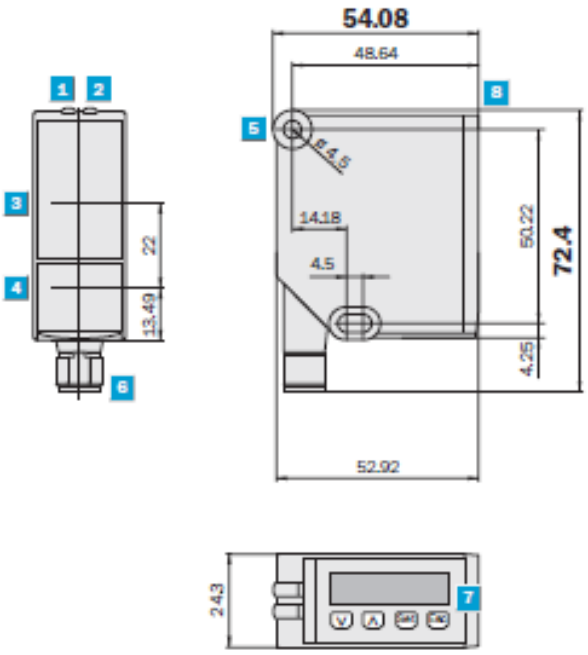
Measurement range
100 ... 600/100 ... 300/
50 ... 150 mm

Distance sensor, scanner mode

- Analogue output 4 ... 20 mA
- High measuring accuracy
- Power-on LED
- Display
- Not sensitive to reflective objects

CE

Dimensional drawing

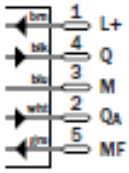


1 Power
2 Status indicator
3 Optical axis – receiver
4 Optical axis – sender
5 Fixing hole
6 Plug M12, 5-pin
7 Display and operating panel
8 Reference edge

Connection type

DT20 Hi

5-pin, M12



AUDIN - 8, avenue de la malle - 51370 Saint Brice Courcelles - France
Tel : 03.26.04.20.21 - Fax : 03.26.04.28.20 - Web : http: www.audin.fr - Email : info@audin.fr

DT20 Hi Distance sensor, scanner mode

Technical data		DT20 Hi	P2148	P2448	P2548	N2148	N2448	N2548
Measurement range								
Object with 6 % ... 90 % remission	100 ... 600 mm		■			■		
Object with 6 % ... 90 % remission	100 ... 300 mm			■			■	
Object with 6 % ... 90 % remission	50 ... 150 mm				■			■
Light source [1]	Laser class 2		■	■	■	■	■	■
Light type	Red light		■	■	■	■	■	■
Light spot diameter	3 mm at 100 ... 600 mm		■	■	■	■	■	■
Supply voltage V_S [2]	10 ... 30 V DC		■	■	■	■	■	■
Power consumption [3]	1.8 W		■	■	■	■	■	■
Residual ripple [4]	≤ 5 V_{pp}		■	■	■	■	■	■
Analogue output [5]	4 ... 20 mA		■	■	■	■	■	■
Accuracy [6]	≙ 2 mm		■			■		
	≙ 1 mm			■			■	
	≙ 0.5 mm				■			■
Reproducibility [7]	≙ 1 mm		■			■		
	≙ 0.5 mm			■			■	
	≙ 0.2 mm				■			■
Resolution	< 0.5 mm		■			■		
	< 0.2 mm			■			■	
	< 0.1 mm				■			■
Response time [8]	< 15 ms		■	■	■	■	■	■
Output rate	< 2.8 ms		■	■	■	■	■	■
Temperature drift	0.25 mm/K		■	■	■	■	■	■
Switching outputs	PNP, Q/$\overline{Q}$		■	■	■			
	NPN, Q/$\overline{Q}$					■	■	■
Signal voltage PNP	HIGH = V_S – (< 2 V)/LOW = 0 V		■	■	■	■	■	■
Signal voltage NPN	HIGH = V_S/LOW ≤ 2 V		■	■	■	■	■	■
Connection type	M12 plug, 5-pin		■	■	■	■	■	■
VDE protection class	⧈		■	■	■	■	■	■
Enclosure rating	IP 65		■	■	■	■	■	■
Ambient temperature T_A	Operation –20 ... +55 °C [9]		■	■	■	■	■	■
	Storage –40 ... +60 °C		■	■	■	■	■	■
Weight	135 g		■	■	■	■	■	■
Housing material	Metal		■	■	■	■	■	■

1) Average service life 50,000 h at T_A = +25 °C
2) Limit values, reverse-polarity protected. Operation in short-circuit protected network max. 8 A
3) Without load
4) May not exceed or fall short of V_S tolerances
5) Freely scalable
6) As 6 ... 90 % remission
7) Switch-on/off medium
8) Lateral entry of object into measurement range
9) As 24 V
10) Continuous change of distance in measurement area

Reproducibility and response time dependent on set mode

Mode	Response time [10]	Reproducibility		
		600 mm	300 mm	150 mm
Fast	2.5 ms	≙ 2 mm	≙ 1 mm	≙ 0.5 mm
Medium	10 ms	≙ 1 mm	≙ 0.5 mm	≙ 0.25 mm
Slow	40 ms	≙ 0.5 mm	≙ 0.25 mm	≙ 0.13 mm

Ordering information

Type	Order no.
DT20-P2148	1040012
DT20-N2148	1040140
DT20-P2448	1040406
DT20-N2448	1040713
DT20-P2548	1041278
DT20-N2548	1041279

Great Britain
Phone +44 (0)1727 83 11 21
E-Mail: info@sick.co.uk

USA
Phone +1(952) 941-6780
tollfree +1-800-325-7425
E-Mail: info@sickusa.com

Australia
Phone +61 3 9497 4100
tollfree +1800 33 48 02
E-Mail: sales@sick.com.au

More representatives and agencies in all major industrial nations at www.sick.com

SICK
Sensor Intelligence.

SICK AG | Waldkirch | Germany | www.sick.com

Future Technology Devices International Ltd.

FT232R USB UART I.C.

Incorporating Clock Generator Output and FTDIChip-ID™ Security Dongle

The ***FT232R*** *is the latest device to be added to FTDI's range of USB UART interface Integrated Circuit Devices. The FT232R is a USB to serial UART interface with optional clock generator output, and the new FTDIChip-ID™ security dongle feature. In addition, asynchronous and synchronous bit bang interface modes are available. USB to serial designs using the FT232R have been further simplified by fully integrating the external EEPROM, clock circuit and USB resistors onto the device.*

The FT232R adds two new functions compared with its predecessors, effectively making it a "3-in-1" chip for some application areas. The internally generated clock (6MHz, 12MHz, 24MHz, and 48MHz) can be brought out of the device and used to drive a microcontroller or external logic. A unique number (the FTDIChip-ID™) is burnt into the device during manufacture and is readable over USB, thus forming the basis of a security dongle which can be used to protect customer application software from being copied.

The FT232R is available in Pb-free (RoHS compliant) compact 28-Lead SSOP and QFN-32 packages.

1. Features

1.1 Hardware Features

- Single chip USB to asynchronous serial data transfer interface.
- Entire USB protocol handled on the chip - No USB-specific firmware programming required.
- UART interface support for 7 or 8 data bits, 1 or 2 stop bits and odd / even / mark / space / no parity.
- Fully assisted hardware or X-On / X-Off software handshaking.
- Data transfer rates from 300 baud to 3 Megabaud (RS422 / RS485 and at TTL levels) and 300 baud to 1 Megabaud (RS232).
- 256 byte receive buffer and 128 byte transmit buffer utilising buffer smoothing technology to allow for high data throughput.
- FTDI's royalty-free VCP and D2XX drivers eliminate the requirement for USB driver development in most cases.
- In-built support for event characters and line break condition.
- New USB FTDIChip-ID™ feature.
- New configurable CBUS I/O pins.
- Auto transmit buffer control for RS485 applications.
- Transmit and receive LED drive signals.
- New 48MHz, 24MHz,12MHz, and 6MHz clock output signal Options for driving external MCU or FPGA.
- FIFO receive and transmit buffers for high data throughput.
- Adjustable receive buffer timeout.
- Synchronous and asynchronous bit bang mode interface options with RD# and WR# strobes.
- New CBUS bit bang mode option.
- Integrated 1024 Bit internal EEPROM for storing USB VID, PID, serial number and product description strings, and CBUS I/O configuration.
- Device supplied preprogrammed with unique USB serial number.
- Support for USB suspend and resume.
- Support for bus powered, self powered, and high-power bus powered USB configurations.
- Integrated 3.3V level converter for USB I/O .
- Integrated level converter on UART and CBUS for interfacing to 5V - 1.8V Logic.
- True 5V / 3.3V / 2.8V / 1.8V CMOS drive output and TTL input.
- High I/O pin output drive option.
- Integrated USB resistors.
- Integrated power-on-reset circuit.
- Fully integrated clock - no external crystal, oscillator, or resonator required.
- Fully integrated AVCC supply filtering - No separate AVCC pin and no external R-C filter required.
- UART signal inversion option.
- USB bulk transfer mode.
- 3.3V to 5.25V Single Supply Operation.
- Low operating and USB suspend current.
- Low USB bandwidth consumption.
- UHCI / OHCI / EHCI host controller compatible
- USB 2.0 Full Speed compatible.
- -40°C to 85°C extended operating temperature range.
- Available in compact Pb-free 28 Pin SSOP and QFN-32 packages (both RoHS compliant).

1.2 Driver Support

Royalty-Free VIRTUAL COM PORT (VCP) DRIVERS for...

- Windows 98, 98SE, ME, 2000, Server 2003, XP.
- Windows Vista / Longhorn*
- Windows XP 64-bit.*
- Windows XP Embedded.
- Windows CE.NET 4.2 & 5.0
- MAC OS 8 / 9, OS-X
- Linux 2.4 and greater

Royalty-Free D2XX *Direct* Drivers (USB Drivers + DLL S/W Interface)

- Windows 98, 98SE, ME, 2000, Server 2003, XP.
- Windows Vista / Longhorn*
- Windows XP 64-bit.*
- Windows XP Embedded.
- Windows CE.NET 4.2 & 5.0
- Linux 2.4 and greater

The drivers listed above are all available to download for free from the FTDI website. Various 3rd Party Drivers are also available for various other operating systems - see the FTDI website for details.

* Currently Under Development. Contact FTDI for availability.

1.3 Typical Applications

- USB to RS232 / RS422 / RS485 Converters
- Upgrading Legacy Peripherals to USB
- Cellular and Cordless Phone USB data transfer cables and interfaces
- Interfacing MCU / PLD / FPGA based designs to USB
- USB Audio and Low Bandwidth Video data transfer
- PDA to USB data transfer
- USB Smart Card Readers
- USB Instrumentation
- USB Industrial Control
- USB MP3 Player Interface
- USB FLASH Card Reader / Writers
- Set Top Box PC - USB interface
- USB Digital Camera Interface
- USB Hardware Modems
- USB Wireless Modems
- USB Bar Code Readers
- USB Software / Hardware Encryption Dongles

2. Enhancements

2.1 Device Enhancements and Key Features

This section summarises the enhancements and the key features of the FT232R device. For further details, consult the device pin-out description and functional description sections.

Integrated Clock Circuit - Previous generations of FTDI's USB UART devices required an external crystal or ceramic resonator. The clock circuit has now been integrated onto the device meaning that no crystal or ceramic resonator is required. However, if required, an external 12MHz crystal can be used as the clock source.

Integrated EEPROM - Previous generations of FTDI's USB UART devices required an external EEPROM if the device were to use USB Vendor ID (VID), Product ID (PID), serial number and product description strings other than the default values in the device itself. This external EEPROM has now been integrated onto the FT232R chip meaning that all designs have the option to change the product description strings. A user area of the internal EEPROM is available for storing additional data. The internal EEPROM is programmable in circuit, over USB without any additional voltage requirement.

Preprogrammed EEPROM - The FT232R is supplied with its internal EEPROM preprogrammed with a serial number which is unique to each individual device. This, in most cases, will remove the need to program the device EEPROM.

Integrated USB Resistors - Previous generations of FTDI's USB UART devices required two external series resistors on the USBDP and USBDM lines, and a 1.5 kΩ pull up resistor on USBDP. These three resistors have now been integrated onto the device.

Integrated AVCC Filtering - Previous generations of FTDI's USB UART devices had a separate AVCC pin - the supply to the internal PLL. This pin required an external R-C filter. The separate AVCC pin is now connected internally to VCC, and the filter has now been integrated onto the chip.

Less External Components - Integration of the crystal, EEPROM, USB resistors, and AVCC filter will substantially reduce the bill of materials cost for USB interface designs using the FT232R compared to its FT232BM predecessor.

Transmit and Receive Buffer Smoothing - The FT232R's 256 byte receive buffer and 128 byte transmit buffer utilise new buffer smoothing technology to allow for high data throughput.

Configurable CBUS I/O Pin Options - There are now 5 configurable Control Bus (CBUS) lines. Options are **TXDEN** - transmit enable for RS485 designs, **PWREN#** - Power control for high power, bus powered designs, **TXLED#** - for pulsing an LED upon transmission of data, **RXLED# -** for pulsing an LED upon receiving data, **TX&RXLED# -** which will pulse an LED upon transmission OR reception of data, **SLEEP# -** indicates that the device going into USB suspend mode, **CLK48 / CLK24 / CLK12 / CLK6** - 48MHz, 24MHz,12MHz, and 6MHz clock output signal options. There is also the option to bring out bit bang mode read and write strobes (see below). The CBUS lines can be configured with any one of these output options by setting bits in the internal EEPROM. The device is supplied with the most commonly used pin definitions preprogrammed - see Section 10 for details.

Enhanced Asynchronous Bit Bang Mode with RD# and WR# Strobes - The FT232R supports FTDI's BM chip bit bang mode. In bit bang mode, the eight UART lines can be switched from the regular interface mode to an 8-bit general purpose I/O port. Data packets can be sent to the device and they will be sequentially sent to the interface at a rate controlled by an internal timer (equivalent to the baud rate prescaler). With the FT232R device this mode has been enhanced so that the internal RD# and WR# strobes are now brought out of the device which can be used to allow external logic to be clocked by accesses to the bit bang I/O bus. This option will be described more fully in a separate application note.

Synchronous Bit Bang Mode - Synchronous bit bang mode differs from asynchronous bit bang mode in that the interface pins are only read when the device is written to. Thus making it easier for the controlling program to measure the response to an output stimulus as the data returned is synchronous to the output data. The feature was previously seen in FTDI's FT2232C device. This option will be described more fully in a separate application note.

CBUS Bit Bang Mode - This mode allows four of the CBUS pins to be individually configured as GPIO pins, similar to Asynchronous bit bang mode. It is possible to use this mode while the UART interface is being used, thus providing up to four general purpose I/O pins which are available during normal operation. An application note describing this feature is available separately from the FTDI website.

Lower Supply Voltage - Previous generations of the chip required 5V supply on the VCC pin. The FT232R will work with a Vcc supply in the range 3.3V - 5.25V. Bus powered designs would still take their supply from the 5V on the USB bus, but for self powered designs where only 3.3V is available and there is no 5V supply there is no longer any need for an additional external regulator.

Integrated Level Converter on UART Interface and Control Signals - VCCIO pin supply can be from 1.8V to 5V. Connecting the VCCIO pin to 1.8V, 2.8V, or 3.3V allows the device to directly interface to 1.8V, 2.8V or 3.3V and other logic families without the need for external level converter I.C. devices.

5V / 3.3V / 2.8V / 1.8V Logic Interface - The FT232R provides *true* CMOS Drive Outputs and TTL level Inputs.

Integrated Power-On-Reset (POR) Circuit- The device incorporates an internal POR function. A RESET# pin is available in order to allow external logic to reset the FT232R where required. However, for many applications the RESET# pin can be left unconnected, or pulled up to VCCIO.

Lower Operating and Suspend Current - The device operating supply current has been further reduced to 15mA, and the suspend current has been reduced to around 70µA. This allows greater margin for peripheral designs to meet the USB suspend current limit of 500µA.

Low USB Bandwidth Consumption - The operation of the USB interface to the FT232R has been designed to use as little as possible of the total USB bandwidth available from the USB host controller.

High Output Drive Option - The UART interface and CBUS I/O pins can be made to drive out at three times the standard signal drive level thus allowing multiple devices to be driven, or devices that require a greater signal drive strength to be interfaced to the FT232R. This option is enabled in the internal EEPROM.

Power Management Control for USB Bus Powered, High Current Designs- The PWREN# signal can be used to directly drive a transistor or P-Channel MOSFET in applications where power switching of external circuitry is required. An option in the internal EEPROM makes the device gently pull down on its UART interface lines when the power is shut off (PWREN# is high). In this mode any residual voltage on external circuitry is bled to GND when power is removed, thus ensuring that external circuitry controlled by PWREN# resets reliably when power is restored.

UART Pin Signal Inversion - The sense of each of the eight UART signals can be individually inverted by setting options in the internal EEPROM. Thus, CTS# (active low) can be changed to CTS (active high), or TXD can be changed to TXD#.

FTDIChip-ID™ - Each FT232R is assigned a unique number which is burnt into the device at manufacture. This ID number cannot be reprogrammed by product manufacturers or end-users. This allows the possibility of using FT232R based dongles for software licensing. Further to this, a renewable license scheme can be implemented based on the FTDIChip-ID™ number when encrypted with other information. This encrypted number can be stored in the user area of the FT232R internal EEPROM, and can be decrypted, then compared with the protected FTDIChip-ID™ to verify that a license is valid. Web based applications can be used to maintain product licensing this way. An application note describing this feature is available separately from the FTDI website.

Improved EMI Performance - The reduced operating current and improved on-chip VCC decoupling significantly improves the ease of PCB design requirements in order to meet FCC, CE and other EMI related specifications.

Programmable Receive Buffer Timeout - The receive buffer timeout is used to flush remaining data from the receive buffer. This time defaults to 16ms, but is programmable over USB in 1ms increments from 1ms to 255ms, thus allowing the device to be optimised for protocols that require fast response times from short data packets.

Extended Operating Temperature Range - The FT232R operates over an extended temperature range of -40° to +85° C thus allowing the device to be used in automotive and industrial applications.

New Package Options - The FT232R is available in two packages - a compact 28 pin SSOP (**FT232RL**) and an ultra-compact 5mm x 5mm pinless QFN-32 package (**FT232RQ**). Both packages are lead (Pb) free, and use a 'green' compound. Both packages are fully compliant with European Union directive 2002/95/EC.

3. Block Diagram

3.1 Block Diagram (Simplified)

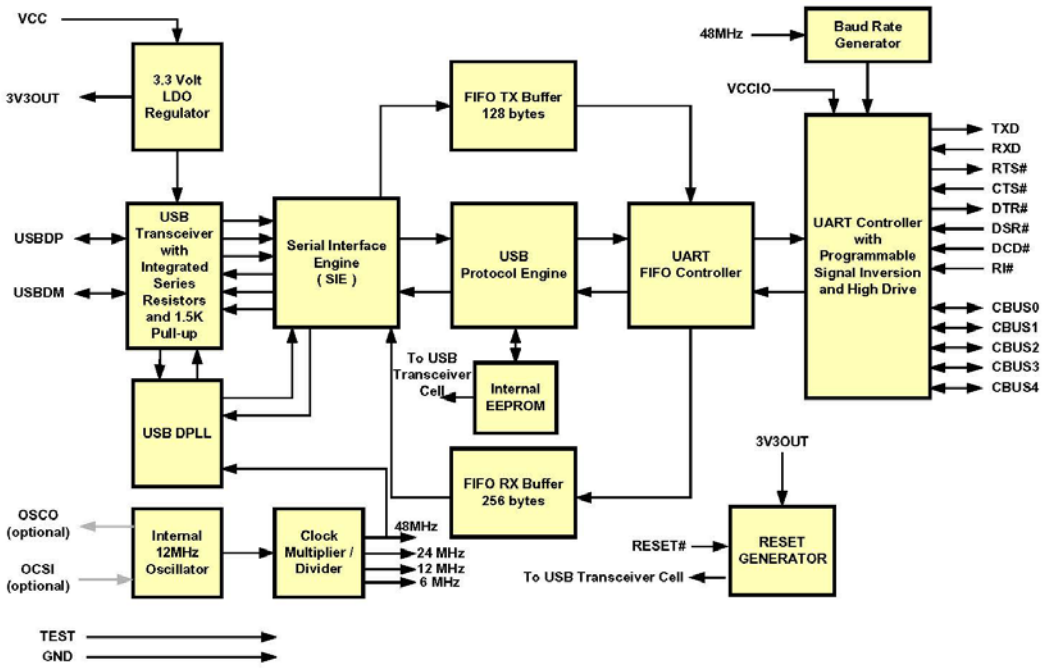


Figure 1 - FT232R Block Diagram

3.2 Functional Block Descriptions

3.3V LDO Regulator - The 3.3V LDO Regulator generates the 3.3V reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It also provides 3.3V power to the 1.5kΩ internal pull up resistor on USBDP. The main function of this block is to power the USB Transceiver and the Reset Generator Cells rather than to power external logic. However, external circuitry requiring a 3.3V nominal supply at a current of around than 50mA could also draw its power from the 3V3OUT pin, if required.

USB Transceiver - The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB cable. The output drivers provide 3.3V level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB data in, SEO and USB Reset condition detection. This Cell also incorporates internal USB series resistors on the USB data lines, and a 1.5kΩ pull up resistor on USBDP.

USB DPLL - The USB DPLL cell locks on to the incoming NRZI USB data and provides separate recovered clock and data signals to the SIE block.

Internal 12MHz Oscillator - The Internal 12MHz Oscillator cell generates a 12MHz reference clock input to the x4 Clock multiplier. The 12MHz Oscillator is also used as the reference clock for the SIE, USB Protocol Engine and UART FIFO controller blocks

Clock Multiplier / Divider - The Clock Multiplier / Divider takes the 12MHz input from the Oscillator Cell and generates the 48MHz, 24MHz, 12MHz, and 6MHz reference clock signals. The 48Mz clock reference is used for the USB DPLL and the Baud Rate Generator blocks.

Serial Interface Engine (SIE) - The Serial Interface Engine (SIE) block performs the Parallel to Serial and Serial to Parallel conversion of the USB data. In accordance to the USB 2.0 specification, it performs bit stuffing / un-stuffing and CRC5 / CRC16 generation / checking on the USB data stream.

USB Protocol Engine - The USB Protocol Engine manages the data stream from the device USB control endpoint. It handles the low level USB protocol (Chapter 9) requests generated by the USB host controller and the commands for controlling the functional parameters of the UART.

FIFO TX Buffer (128 bytes) - Data from the USB data out endpoint is stored in the FIFO TX buffer and removed from the buffer to the UART transmit register under control of the UART FIFO controller.

FIFO RX Buffer (256 bytes) - Data from the UART receive register is stored in the FIFO RX buffer prior to being removed by the SIE on a USB request for data from the device data in endpoint.

UART FIFO Controller - The UART FIFO controller handles the transfer of data between the FIFO RX and TX buffers and the UART transmit and receive registers.

UART Controller with Programmable Signal Inversion and High Drive - Together with the UART FIFO Controller the UART Controller handles the transfer of data between the FIFO RX and FIFO TX buffers and the UART transmit and receive registers. It performs asynchronous 7 / 8 bit Parallel to Serial and Serial to Parallel conversion of the data on the RS232 (RS422 and RS485) interface. Control signals supported by UART mode include RTS, CTS, DSR , DTR, DCD and RI. The UART Controller also provides a transmitter enable control signal pin option (TXDEN) to assist with interfacing to RS485 transceivers. RTS / CTS, DSR / DTR and X-On / X-Off handshaking options are also supported. Handshaking, where required, is handled in hardware to ensure fast response times. The UART also supports the RS232 BREAK setting and detection conditions. A new feature, programmable in the internal EEPROM allows the UART signals to each be individually inverted. Another new EEPROM programmable feature allows a high signal drive strength to be enabled on the UART interface and CBUS pins.

Baud Rate Generator - The Baud Rate Generator provides a x16 clock input to the UART Controller from the 48MHz reference clock and consists of a 14 bit prescaler and 3 register bits which provide fine tuning of the baud rate (used to divide by a number plus a fraction or "sub-integer"). This determines the Baud Rate of the UART, which is programmable from 183 baud to 3 million baud.

The FT232R supports all standard baud rates and non-standard baud rates from 300 Baud up to 3 Megabaud. Achievable non-standard baud rates are calculated as follows -

Baud Rate = 3000000 / (n + *x*)

where n can be any integer between 2 and 16,384 (= 2^{14}) and *x* can be a sub-integer of the value 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, or 0.875. When n = 1, *x* = 0, i.e. baud rate divisors with values between 1 and 2 are not possible.

This gives achievable baud rates in the range 183.1 baud to 3,000,000 baud. When a non-standard baud rate is required simply pass the required baud rate value to the driver as normal, and the FTDI driver will calculate the required divisor, and set the baud rate. See FTDI application note AN232B-05 for more details.

RESET Generator - The integrated Reset Generator Cell provides a reliable power-on reset to the device internal circuitry on power up. A RESET# input pin is provided to allow other devices to reset the FT232R. RESET# can be tied to VCCIO or left unconnected, unless it is a requirement to reset the device from external logic or an external reset generator I.C.

Internal EEPROM - The internal EEPROM in the FT232R can be used to store USB Vendor ID (VID), Product ID (PID), device serial number, product description string, and various other USB configuration descriptors. The internal EEPROM is also used to configure the CBUS pin functions. The device is supplied with the internal EEPROM settings preprogrammed as described in Section 10.

4. Device Pin Out and Signal Descriptions

4.1 28-LD SSOP Package

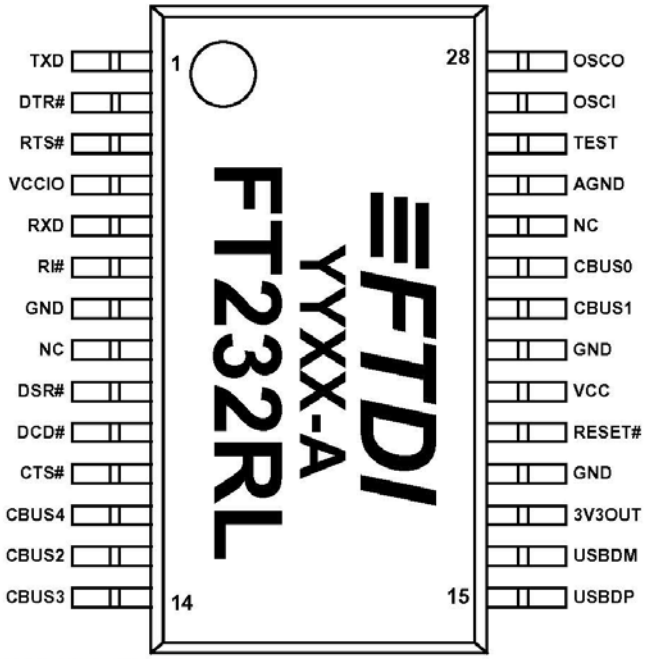


Figure 2 - 28 Pin SSOP Package Pin Out

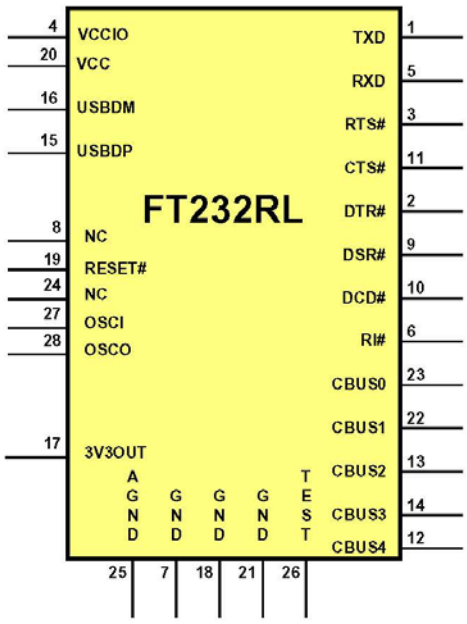


Figure 3 - 28 Pin SSOP Package Pin Out (Schematic Symbol)

4.2 SSOP-28 Package Signal Descriptions

Table 1 - SSOP Package Pin Out Description

Pin No.	Name	Type	Description
USB Interface Group			
15	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to 3.3V
16	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.
Power and Ground Group			
4	VCCIO	PWR	+1.8V to +5.25V supply to the UART Interface and CBUS group pins (1...3, 5, 6, 9...14, 22, 23). In USB bus powered designs connect to 3V3OUT to drive out at 3.3V levels, or connect to VCC to drive out at 5V CMOS level. This pin can also be supplied with an external 1.8V - 2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to Vcc. This means that in bus powered designs a regulator which is supplied by the 5V on the USB bus should be used.
7, 18, 21	GND	PWR	Device ground supply pins
17	3V3OUT	Output	3.3V output from integrated L.D.O. regulator. This pin should be decoupled to ground using a 100nF capacitor. The prime purpose of this pin is to provide the internal 3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the FT232R's VCCIO pin.
20	VCC	PWR	3.3V to 5.25V supply to the device core.
25	AGND	PWR	Device analog ground supply for internal clock multiplier
Miscellaneous Signal Group			
8, 24	NC	NC	No internal connection.
19	RESET#	Input	Can be used by an external device to reset the FT232R. If not required can be left unconnected, or pulled up to VCCIO.
26	TEST	Input	Puts the device into I.C. test mode. Must be tied to GND for normal operation.
27	OSCI	Input	Input to 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation. *
28	OSCO	Output	Output from 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation if internal oscillator is used. *
UART Interface and CBUS Group **			
1	TXD	Output	Transmit Asynchronous Data Output.
2	DTR#	Output	Data Terminal Ready Control Output / Handshake signal.
3	RTS#	Output	Request To Send Control Output / Handshake signal.
5	RXD	Input	Receive Asynchronous Data Input.
6	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low can be used to resume the PC USB host controller from suspend.
9	DSR#	Input	Data Set Ready Control Input / Handshake signal.
10	DCD#	Input	Data Carrier Detect Control input.
11	CTS#	Input	Clear to Send Control input / Handshake signal.
12	CBUS4	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is SLEEP#. See CBUS Signal Options, Table 3.
13	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXDEN. See CBUS Signal Options, Table 3.
14	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is PWREN#. See CBUS Signal Options, Table 3.
22	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is RXLED#. See CBUS Signal Options, Table 3.
23	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXLED#. See CBUS Signal Options, Table 3.

* Contact FTDI technical support for details on how to use an external crystal, ceramic resonator, or oscillator with the FT232R.

** When used in Input Mode, these pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

4.3 QFN-32 Package

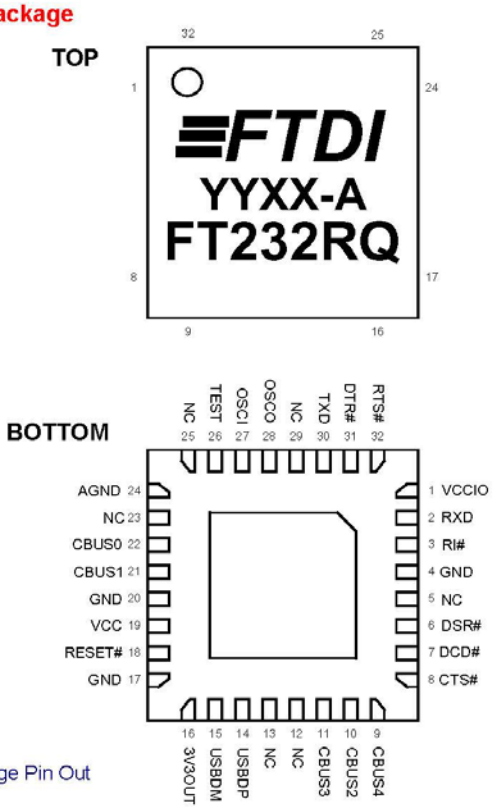


Figure 4 - QFN-32 Package Pin Out

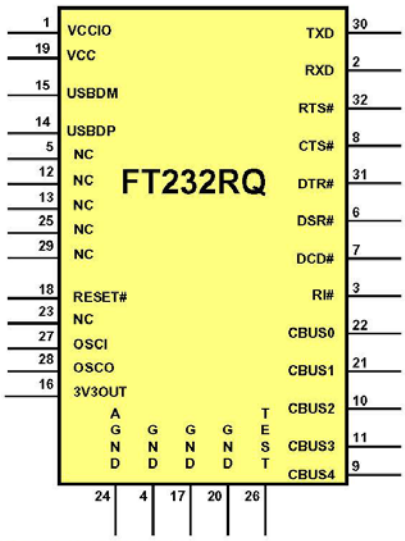


Figure 5 - QFN-32 Package Pin Out (Schematic Symbol)

4.4 QFN-32 Package Signal Descriptions

Table 2 - QFN Package Pin Out Description

Pin No.	Name	Type	Description
USB Interface Group			
14	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to 3.3V
15	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.
Power and Ground Group			
1	VCCIO	PWR	+1.8V to +5.25V supply to UART Interface and CBUS group pins (2,3, 6, ...,11, 21, 22, 30,..32). In USB bus powered designs connect to 3V3OUT to drive out at 3.3V levels, or connect to VCC to drive out at 5V CMOS level. This pin can also be supplied with an external 1.8V - 2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to Vcc. This means that in bus powered designs a regulator which is supplied by the 5V on the USB bus should be used.
4, 17, 20	GND	PWR	Device ground supply pins
16	3V3OUT	Output	3.3V output from integrated L.D.O. regulator. This pin should be decoupled to ground using a 100nF capacitor. The prime purpose of this pin is to provide the internal 3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the FT232R's VCCIO pin.
19	VCC	PWR	3.3V to 5.25V supply to the device core.
24	AGND	PWR	Device analog ground supply for internal clock multiplier
Miscellaneous Signal Group			
5, 12, 13, 23, 25, 29	NC	NC	No internal connection.
18	RESET#	Input	Can be used by an external device to reset the FT232R. If not required can be left unconnected or pulled up to VCCIO.
26	TEST	Input	Puts the device into I.C. test mode. Must be tied to GND for normal operation.
27	OSCI	Input	Input to 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation. *
28	OSCO	Output	Output from 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation if internal oscillator is used. *
UART Interface and CBUS Group **			
30	TXD	Output	Transmit Asynchronous Data Output.
31	DTR#	Output	Data Terminal Ready Control Output / Handshake signal.
32	RTS#	Output	Request To Send Control Output / Handshake signal.
2	RXD	Input	Receive Asynchronous Data Input.
3	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low can be used to resume the PC USB host controller from suspend.
6	DSR#	Input	Data Set Ready Control Input / Handshake signal.
7	DCD#	Input	Data Carrier Detect Control input.
8	CTS#	Input	Clear to Send Control input / Handshake signal.
9	CBUS4	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is SLEEP#. See CBUS Signal Options, Table 3.
10	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXDEN. See CBUS Signal Options, Table 3.
11	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is PWREN#. See CBUS Signal Options, Table 3.
21	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is RXLED#. See CBUS Signal Options, Table 3.
22	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXLED#. See CBUS Signal Options, Table 3.

* Contact FTDI technical support for details on how to use an external crystal, ceramic resonator, or oscillator with the FT232R.

** When used in Input Mode, these pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

4.5 CBUS Signal Options

The following options can be configured on the CBUS I/O pins. CBUS signal options are common to both package versions of the FT232R. These options are all configured in the internal EEPROM using the utility software MPROG, which can be downloaded from the FTDI website. The default configuration is described in Section 10.

Table 3 - CBUS Signal Options

CBUS Signal Option	Available On CBUS Pin...	Description
TXDEN	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Enable transmit data for RS485
PWREN#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Goes low after the device is configured by USB, then high during USB suspend. Can be used to control power to external logic P-Channel logic level MOSFET switch. Enable the interface pull-down option when using the PWREN# pin in this way.
TXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Transmit data LED drive - pulses low when transmitting data via USB. See Section 9 for more details.
RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Receive data LED drive - pulses low when receiving data via USB. See Section 9 for more details.
TX&RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	LED drive - pulses low when transmitting or receiving data via USB. See Section 9 for more details.
SLEEP#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Goes low during USB suspend mode. Typically used to power down an external TTL to RS232 level converter I.C. in USB to RS232 converter designs.
CLK48	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	48MHz Clock output.
CLK24	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	24MHz Clock output.
CLK12	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	12MHz Clock output.
CLK6	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	6MHz Clock output.
CBitBangI/O	CBUS0, CBUS1, CBUS2, CBUS3	CBUS bit bang mode option. Allows up to 4 of the CBUS pins to be used as general purpose I/O. Configured individually for CBUS0, CBUS1, CBUS2 and CBUS3 in the internal EEPROM. A separate application note will describe in more detail how to use CBUS bit bang mode.
BitBangWRn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode WR# strobe Output
BitBangRDn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode RD# strobe Output

5. Package Parameters

The FT232R is supplied in two different packages. The FT232RL is the SSOP-28 option and the FT232RQ is the QFN-32 package option. The solder reflow profile for both packages is described in Section 5.3.

5.1 SSOP-28 Package Dimensions

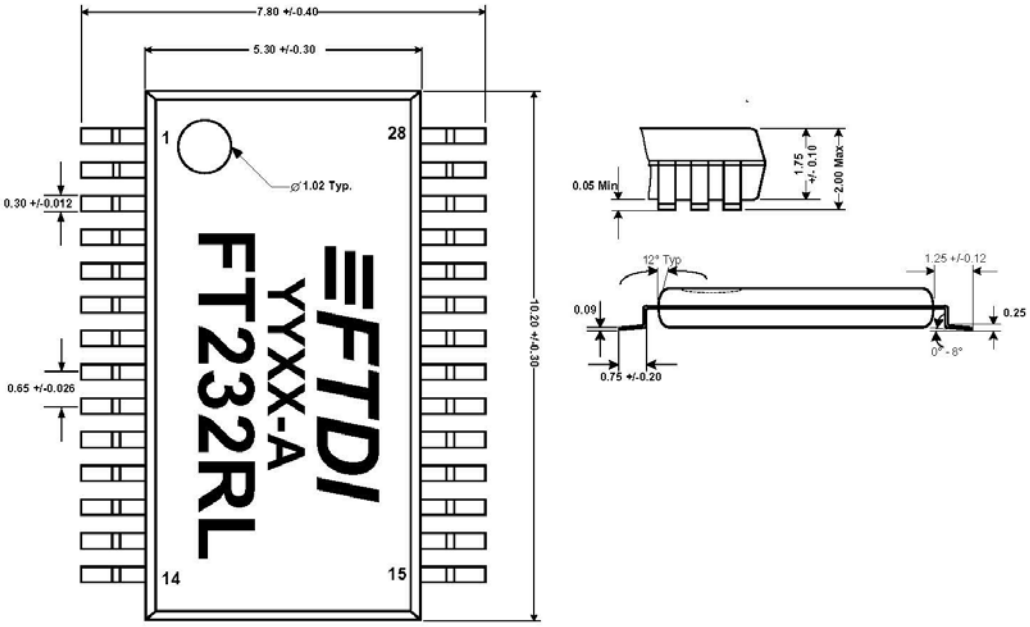


Figure 6 - SSOP-28 Package Dimensions

The FT232RL is supplied in a RoHS compliant 28 pin SSOP package. The package is lead (Pb) free and uses a 'green' compound. The package is fully compliant with European Union directive 2002/95/EC.

This package has a 5.30mm x 10.20mm body (7.80mm x 10.20mm including pins). The pins are on a 0.65 mm pitch. The above mechanical drawing shows the SSOP-28 package – all dimensions are in millimetres.

The date code format is **YYXX** where XX = 2 digit week number, YY = 2 digit year number.

5.2 QFN-32 Package Dimensions

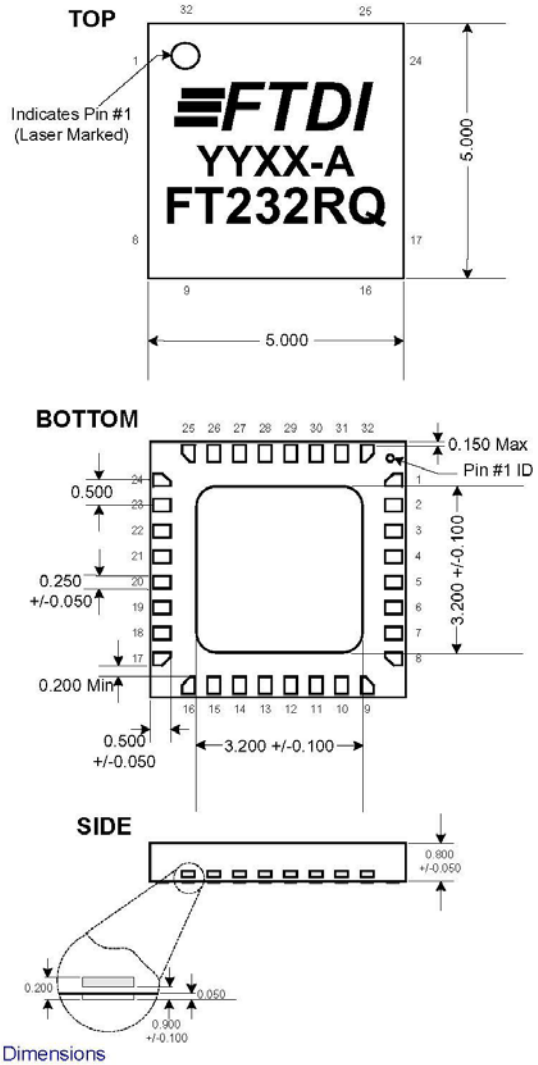


Figure 7 - QFN-32 Package Dimensions

The FT232RQ is supplied in a RoHS compliant leadless QFN-32 package. The package is lead (Pb) free, and uses a 'green' compound. The package is fully compliant with European Union directive 2002/95/EC.

This package has a compact 5.00mm x 5.00mm body. The solder pads are on a 0.50mm pitch. The above mechanical drawing shows the QFN-32 package – all dimensions are in millimetres.

The centre pad on the base of the FT232RQ is not internally connected, and can be left unconnected, or connected to ground (recommended).

The date code format is **YYXX** where XX = 2 digit week number, YY = 2 digit year number.

5.3 QFN-32 Package Typical Pad Layout

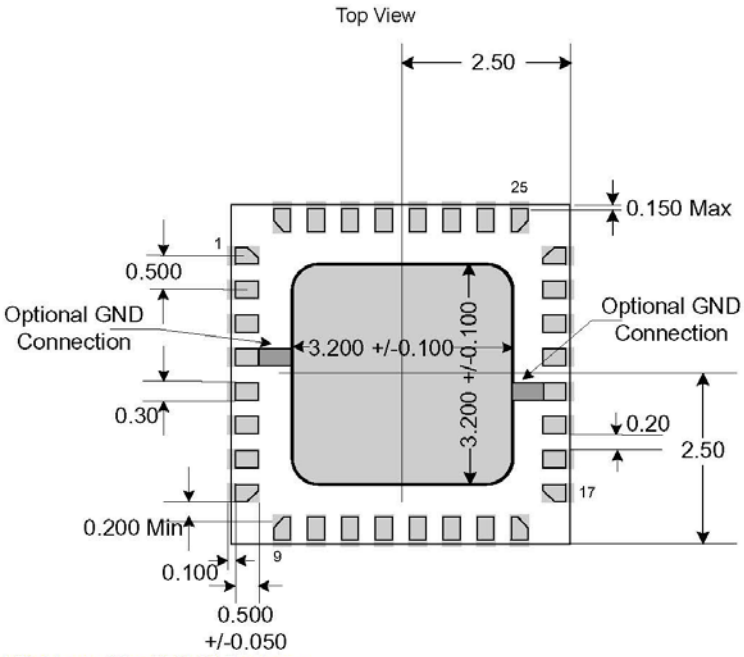


Figure 8 - Typical Pad Layout for QFN-32 Package

5.4 QFN-32 Package Typical Solder Paste Diagram

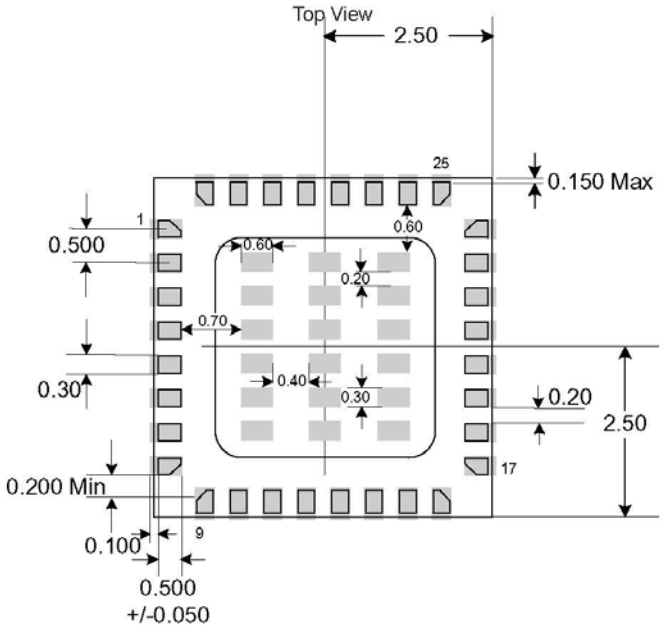


Figure 9 - Typical Solder Paste Diagram for QFN-32 Package

5.5 Solder Reflow Profile

The FT232R is supplied in Pb free 28 LD SSOP and QFN-32 packages. The recommended solder reflow profile for both package options is shown in Figure 10.

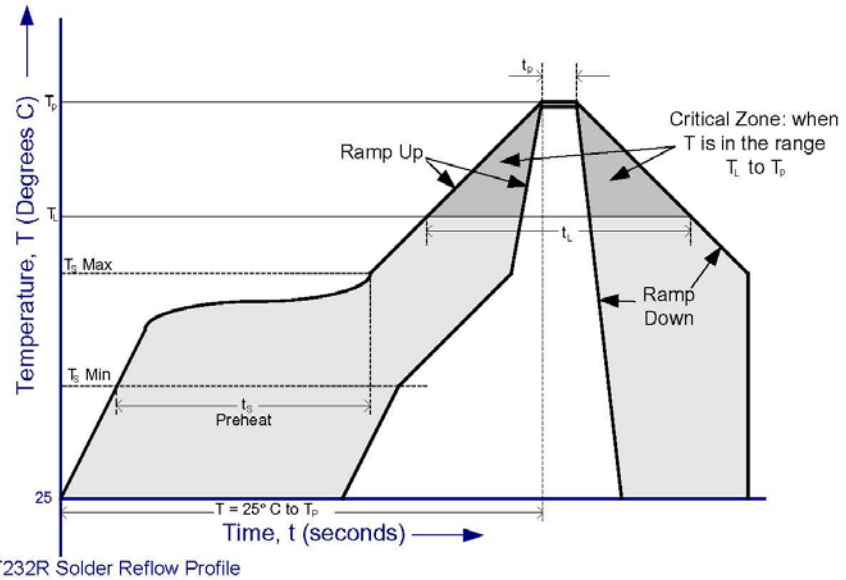


Figure 10 - FT232R Solder Reflow Profile

The recommended values for the solder reflow profile are detailed in Table 4. Values are shown for both a completely Pb free solder process (i.e. the FT232R is used with Pb free solder), and for a non-Pb free solder process (i.e. the FT232R is used with non-Pb free solder).

Table 4 - Reflow Profile Parameter Values

Profile Feature	Pb Free Solder Process	Non-Pb Free Solder Process
Average Ramp Up Rate (T_s to T_p)	3°C / second Max.	3°C / Second Max.
Preheat - Temperature Min (T_S Min.) - Temperature Max (T_S Max.) - Time (t_S Min to t_S Max)	 150°C 200°C 60 to 120 seconds	 100°C 150°C 60 to 120 seconds
Time Maintained Above Critical Temperature T_L: - Temperature (T_L) - Time (t_L)	 217°C 60 to 150 seconds	 183°C 60 to 150 seconds
Peak Temperature (T_P)	260°C	240°C
Time within 5°C of actual Peak Temperature (t_p)	20 to 40 seconds	20 to 40 seconds
Ramp Down Rate	6°C / second Max.	6°C / second Max.
Time for T= 25°C to Peak Temperature, T_p	8 minutes Max.	6 minutes Max.

6. Device Characteristics and Ratings

6.1 Absolute Maximum Ratings

The absolute maximum ratings for the FT232R devices are as follows. These are in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

Table 5 - Absolute Maximum Ratings

Parameter	Value	Unit
Storage Temperature	-65°C to 150°C	Degrees C
Floor Life (Out of Bag) At Factory Ambient (30°C / 60% Relative Humidity)	168 Hours (IPC/JEDEC J-STD-033A MSL Level 3 Compliant)*	Hours
Ambient Temperature (Power Applied)	-40°C to 85°C	Degrees C.
Vcc Supply Voltage	-0.5 to +6.00	V
D.C. Input Voltage - USBDP and USBDM	-0.5 to +3.8	V
D.C. Input Voltage - High Impedance Bidirectionals	-0.5 to +(Vcc +0.5)	V
D.C. Input Voltage - All other Inputs	-0.5 to +(Vcc +0.5)	V
D.C. Output Current - Outputs	24	mA
DC Output Current - Low Impedance Bidirectionals	24	mA
Power Dissipation (Vcc = 5.25V)	500	mW

* If devices are stored out of the packaging beyond this time limit the devices should be baked before use. The devices should be ramped up to a temperature of 125°C and baked for up to 17 hours.

6.2 DC Characteristics

DC Characteristics (Ambient Temperature = -40°C to +85°C)

Table 6 - Operating Voltage and Current

Parameter	Description	Min	Typ	Max	Units	Conditions
Vcc1	VCC Operating Supply Voltage	3.3	-	5.25	V	
Vcc2	VCCIO Operating Supply Voltage	1.8	-	5.25	V	
Icc1	Operating Supply Current	-	15	-	mA	Normal Operation
Icc2	Operating Supply Current	50	70	100	µA	USB Suspend

Table 7 - UART and CBUS I/O Pin Characteristics (VCCIO = 5.0V, Standard Drive Level)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 2mA
Vin	Input Switching Threshold	1.3	1.6	1.9	V	**
VHys	Input Switching Hysteresis	50	55	60	mV	**

Table 8 - UART and CBUS I/O Pin Characteristics (VCCIO = 3.3V, Standard Drive Level)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	2.2	2.7	3.2	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 9 - UART and CBUS I/O Pin Characteristics (VCCIO = 2.8V, Standard Drive Level)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	2.1	2.6	3.1	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 10 - UART and CBUS I/O Pin Characteristics (VCCIO = 5.0V, High Drive Level)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 6mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 6mA
Vin	Input Switching Threshold	1.3	1.6	1.9	V	**
VHys	Input Switching Hysteresis	50	55	60	mV	**

Table 11 - UART and CBUS I/O Pin Characteristics (VCCIO = 3.3V, High Drive Level)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	2.2	2.8	3.2	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 12 - UART and CBUS I/O Pin Characteristics (VCCIO = 2.8V, High Drive Level)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	2.1	2.8	3.2	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

**Inputs have an internal 200kΩ pull-up resistor to VCCIO.

Table 13 - RESET# and TEST Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
Vin	Input Switching Threshold	1.3	1.6	1.9	V	
VHys	Input Switching Hysteresis	50	55	60	mV	

Table 14 - USB I/O Pin (USBDP, USBDM) Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
UVoh	I/O Pins Static Output (High)	2.8		3.6	V	RI = 1.5kΩ to 3V3Out (D+) RI = 15kΩ to GND (D-)
UVol	I/O Pins Static Output (Low)	0		0.3	V	RI = 1.5kΩ to 3V3Out (D+) RI = 15kΩ to GND (D-)
UVse	Single Ended Rx Threshold	0.8		2.0	V	
UCom	Differential Common Mode	0.8		2.5	V	
UVDif	Differential Input Sensitivity	0.2			V	
UDrvZ	Driver Output Impedance	26	29	44	Ohms	***

***Driver Output Impedance includes the internal USB series resistors on USBDP and USBDM pins.

6.3 EEPROM Reliability Characteristics

The internal 1024 Bit EEPROM has the following reliability characteristics-

Table 15 - EEPROM Characteristics

Parameter Description	Value	Unit
Data Retention	15	Years
Read / Write Cycles	100,000	Cycles

6.4 Internal Clock Characteristics

The internal Clock Oscillator has the following characteristics.

Table 16 - Internal Clock Characteristics

Parameter	Value			Unit
	Min	Typical	Max	
Frequency of Operation	11.98	12.00	12.02	MHz****
Clock Period	83.19	83.33	83.47	ns
Duty Cycle	45	50	55	%

****Equivalent to +/-1667ppm.

Table 17 - OSCI, OSCO Pin Characteristics (Optional - Only applies if external Oscillator is used*****)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	2.8	-	3.6	V	Fosc = 12MHz
Vol	Output Voltage Low	0.1	-	1.0	V	Fosc = 12MHz
Vin	Input Switching Threshold	1.8	2.5	3.2	V	

*****When supplied the device is configured to use its internal clock oscillator. Users who wish to use an external oscillator or crystal should contact FTDI technical support.

7. Device Configurations

Please note that pin numbers on the FT232R chip in this section have deliberately been left out as they vary between the FT232RL and FT232RQ versions of the device. All of these configurations apply to both package options for the FT232R device. Please refer to Section 4 for the package option pin-out and signal descriptions.

7.1 Bus Powered Configuration

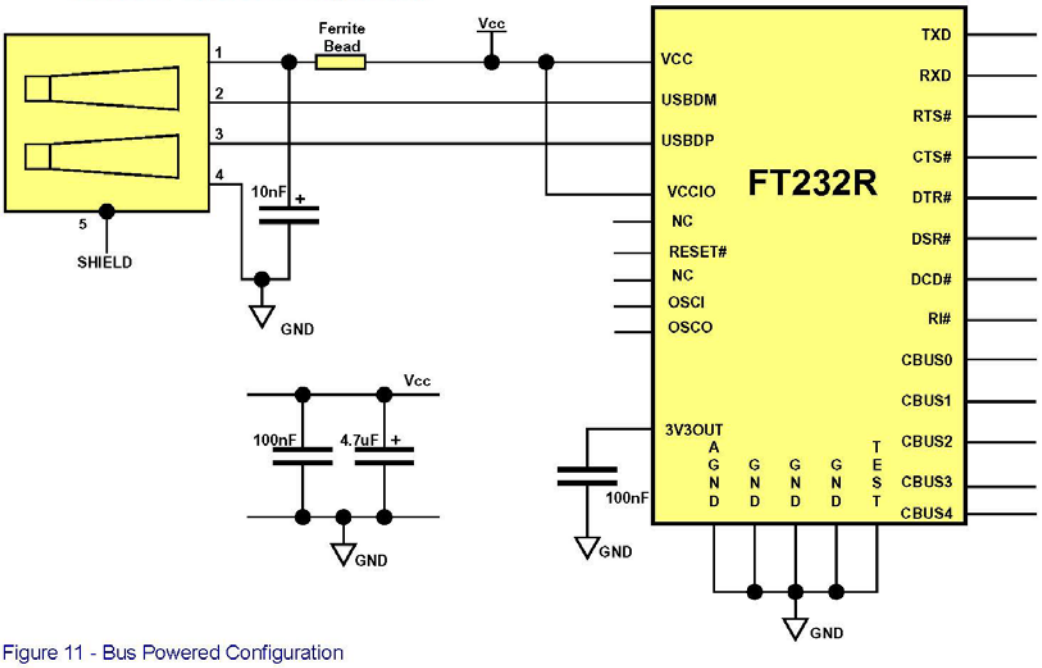


Figure 11 - Bus Powered Configuration

Figure 11 illustrates the FT232R in a typical USB bus powered design configuration. A USB Bus Powered device gets its power from the USB bus. Basic rules for USB Bus power devices are as follows –

i) On plug-in to USB, the device must draw no more than 100mA.

ii) On USB Suspend the device must draw no more than 500µA.

iii) A Bus Powered High Power USB Device (one that draws more than 100mA) should use one of the CBUS pins configured as PWREN# and use it to keep the current below 100mA on plug-in and 500µA on USB suspend.

iv) A device that consumes more than 100mA can not be plugged into a USB Bus Powered Hub.

v) No device can draw more that 500mA from the USB Bus.

The power descriptor in the internal EEPROM should be programmed to match the current draw of the device.
A Ferrite Bead is connected in series with USB power to prevent noise from the device and associated circuitry (EMI) being radiated down the USB cable to the Host. The value of the Ferrite Bead depends on the total current required by the circuit – a suitable range of Ferrite Beads is available from Steward (www.steward.com) for example Steward Part # MI0805K400R-00.

7.2 Self Powered Configuration

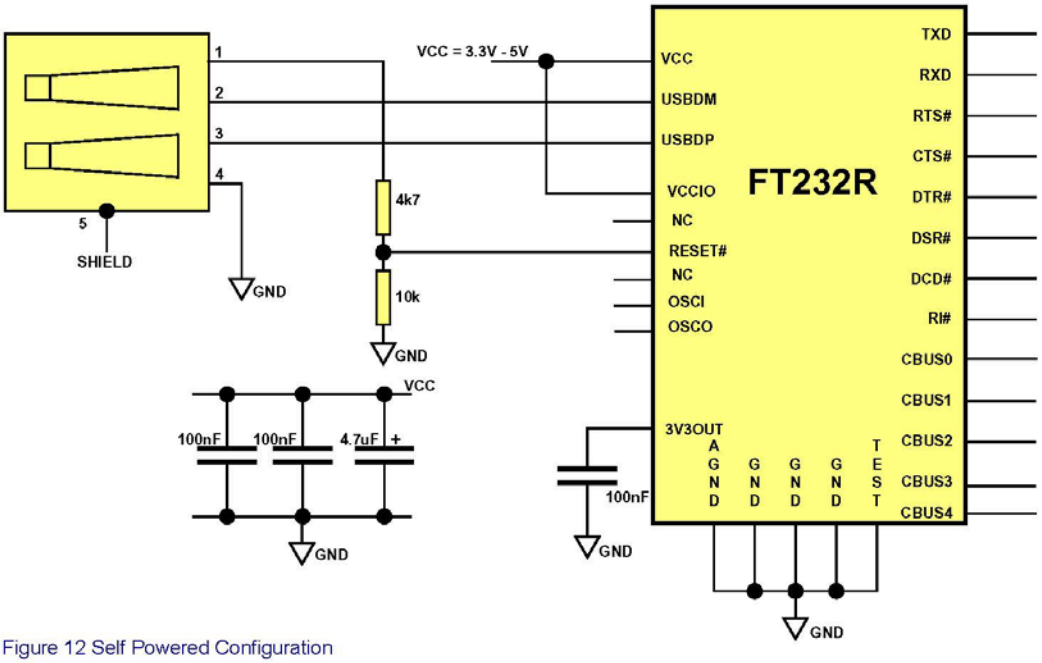


Figure 12 Self Powered Configuration

Figure 12 illustrates the FT232R in a typical USB self powered configuration. A USB Self Powered device gets its power from its own power supply and does not draw current from the USB bus. The basic rules for USB Self powered devices are as follows –

i) A Self Powered device should not force current down the USB bus when the USB Host or Hub Controller is powered down.

ii) A Self Powered Device can use as much current as it likes during normal operation and USB suspend as it has its own power supply.

iii) A Self Powered Device can be used with any USB Host and both Bus and Self Powered USB Hubs

The power descriptor in the internal EEPROM should be programmed to a value of zero (self powered).

In order to meet requirement (i) the USB Bus Power is used to control the RESET# Pin of the FT232R device. When the USB Host or Hub is powered up the internal 1.5kΩ resistor on USBDP is pulled up to 3.3V, thus identifying the device as a full speed device to USB. When the USB Host or Hub power is off, RESET# will go low and the device will be held in reset. As RESET# is low, the internal 1.5kΩ resistor will not be pulled up to 3.3V, so no current will be forced down USBDP via the 1.5kΩ pull-up resistor when the host or hub is powered down. Failure to do this may cause some USB host or hub controllers to power up erratically.

Figure 10 illustrates a self powered design which has a 3.3V - 5V supply. A design which is interfacing to 2.8V - 1.8V logic would have a 2.8V - 1.8V supply to VCCIO, and a 3.3V - 5V supply to VCC

Note : When the FT232R is in reset, the UART interface pins all go tri-state. These pins have internal 200kΩ pull-up resistors to VCCIO, so they will gently pull high unless driven by some external logic.

7.3 USB Bus Powered with Power Switching Configuration

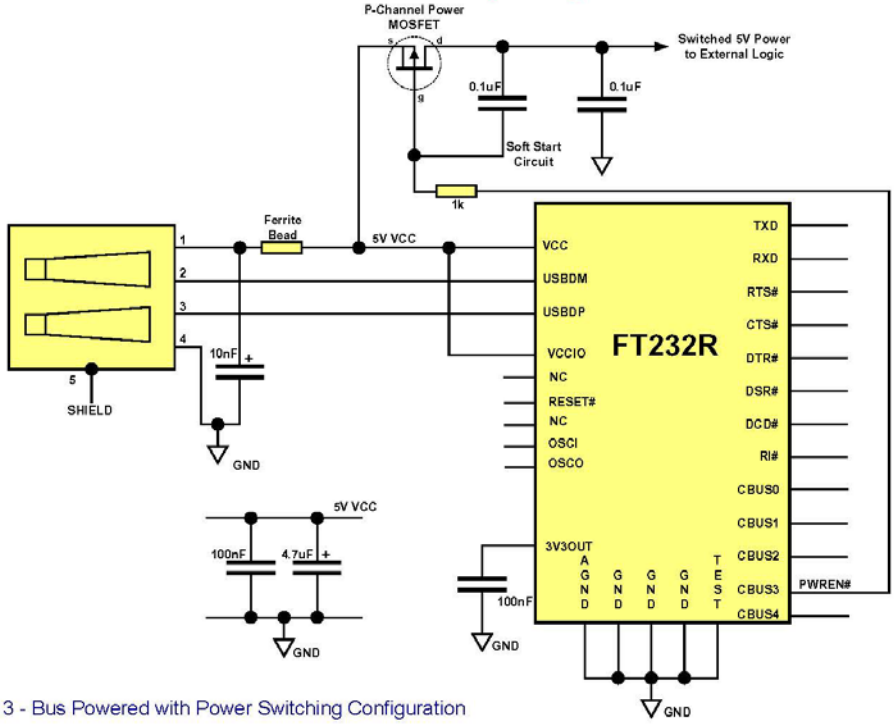


Figure 13 - Bus Powered with Power Switching Configuration

USB Bus powered circuits need to be able to power down in USB suspend mode in order to meet the <= 500µA total USB suspend current requirement (including external logic). Some external logic can power itself down into a low current state by monitoring the PWREN# signal. For external logic that cannot power itself down in this way, the FT232R provides a simple but effective way of turning off power to external circuitry during USB suspend.

Figure 13 shows how to use a discrete P-Channel Logic Level MOSFET to control the power to external logic circuits. A suitable device would be an International Rectifier (www.irf.com) IRLML6402, or equivalent. It is recommended that a "soft start" circuit consisting of a 1kΩ series resistor and a 0.1µF capacitor are used to limit the current surge when the MOSFET turns on. Without the soft start circuit there is a danger that the transient power surge of the MOSFET turning on will reset the FT232R, or the USB host / hub controller. The values used here allow attached circuitry to power up with a slew rate of ~12.5V per millisecond, in other words the output voltage will transition from GND to 5V in approximately 400 microseconds.

Alternatively, a dedicated power switch I.C. with inbuilt "soft-start" can be used instead of a MOSFET. A suitable power switch I.C. for such an application would be a Micrel (www.micrel.com) MIC2025-2BM or equivalent.

Please note the following points in connection with power controlled designs –

i) The logic to be controlled must have its own reset circuitry so that it will automatically reset itself when power is re-applied on coming out of suspend.
ii) Set the Pull-down on Suspend option in the internal EEPROM.
iii) One of the CBUS Pins should be configured as PWREN# in the internal EEPROM, and should be used to switch the power supply to the external circuitry..
iv) For USB high-power bus powered device (one that consumes greater than 100mA, and up to 500mA of current from the USB bus), the power consumption of the device should be set in the max power field in the internal EEPROM. A high-power bus powered device must use this descriptor in the internal EEPROM to inform the system of its power requirements.
v) For 3.3V power controlled circuits the VCCIO pin must not be powered down with the external circuitry (the PWREN# signal gets its VCC supply from VCCIO). Either connect the power switch between the output of the 3.3V regulator and the external 3.3V logic or power VCCIO from the 3V3OUT pin of the FT232R.

7.4 USB Bus Powered with 3.3V / 5V Supply and Logic Drive / IO Supply Voltage

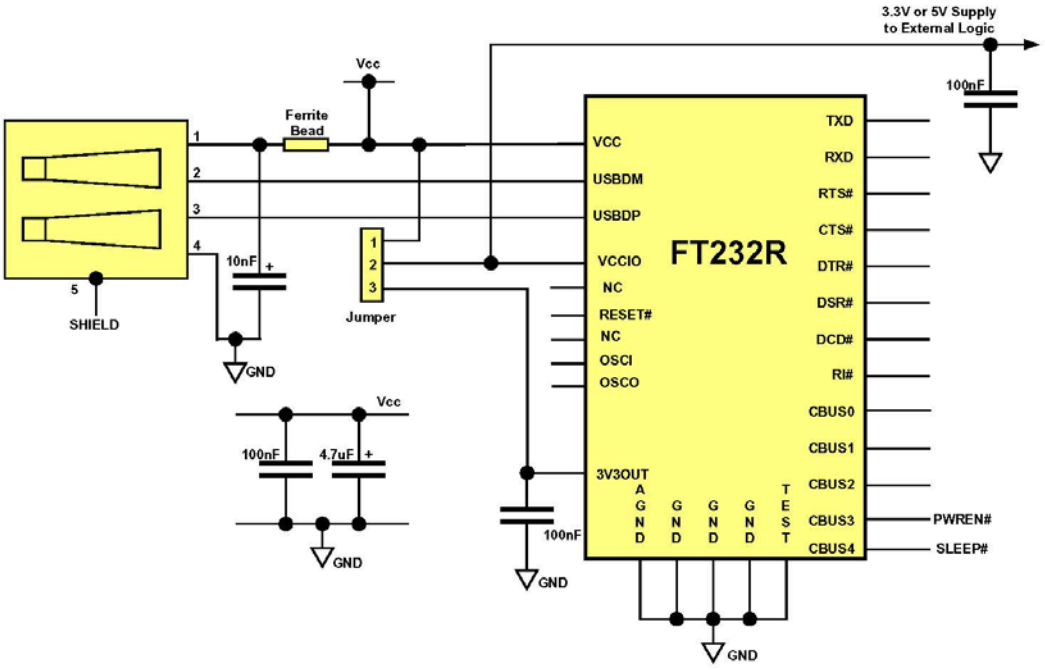


Figure 14 - Bus Powered with 3.3V / 5V Supply and Logic Drive

Figure 14 shows a configuration where a jumper switch is used to allow the FT232R to be interfaced with a 3.3V or 5V logic devices. The VCCIO pin is either supplied with 5V from the USB bus, or with 3.3V from the 3V3OUT pin. The supply to VCCIO is also used to supply external logic.

Please note the following in relation to bus powered designs of this type -

i) PWREN# or SLEEP# signals should be used to power down external logic during USB suspend mode, in order to comply with the limit of 500µA. If this is not possible, use the configuration shown in Section 7.3.

ii) The maximum current source from USB Bus during normal operation should not exceed 100mA, otherwise a bus powered design with power switching (Section 7.3) should be used.

Another possible configuration would be to use a discrete low dropout regulator which is supplied by the 5V on the USB bus to supply 2.8V - 1.8V to the VCCIO pin and to the external logic. VCC would be supplied with the 5V from the USB bus. With VCCIO connected to the output of the low dropout regulator, would in turn will cause the FT232R I/O pins to drive out at 2.8V - 1.8V logic levels.

For USB bus powered circuits some considerations have to be taken into account when selecting the regulator –

iii) The regulator must be capable of sustaining its output voltage with an input voltage of 4.35V. A Low Drop Out (L.D.O.) regulator must be selected.

iv) The quiescent current of the regulator must be low in order to meet the USB suspend total current requirement of <= 500µA during USB suspend.

An example of a regulator family that meets these requirements is the MicroChip / Telcom TC55 Series of devices (www.microchip.com). These devices can supply up to 250mA current and have a quiescent current of under 1µA.

8. Example Interface Configurations

As in the Device Configurations section, please note that pin numbers on the FT232R chip in this section have deliberately been left out as they vary between the FT232RL and FT232RQ versions of the device. All of these configurations apply to both package options for the FT232R device. Please refer to Section 4 for the package option pin-out and signal descriptions.

8.1 USB to RS232 Converter Configuration

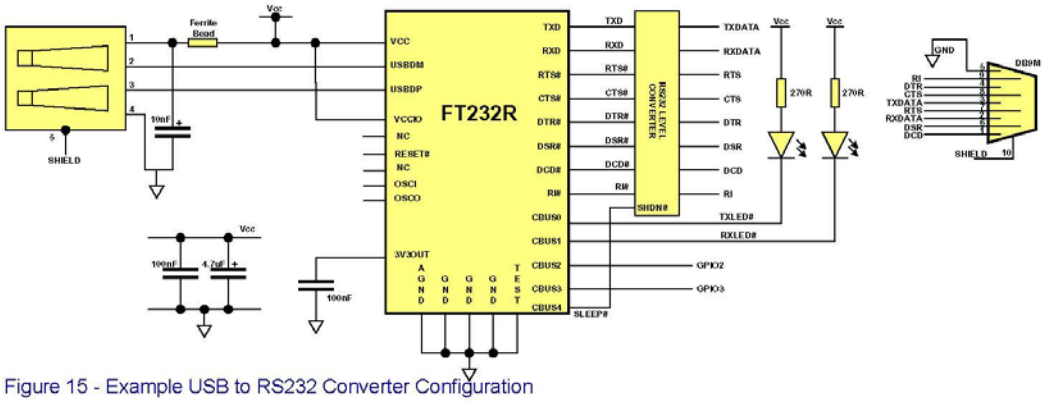


Figure 15 - Example USB to RS232 Converter Configuration

Figure 15 illustrates how to connect an FT232R as a USB to RS232 converter. A TTL – RS232 Level Converter I.C. is used on the serial UART of the FT232R to make the RS232 level conversion. This, for example can be done using the popular "213" series of TTL to RS232 level converters. These devices have 4 transmitters and 5 receivers in a 28-LD SSOP package and feature an in-built voltage converter to convert the 5V (nominal) VCC to the +/- 9 volts required by RS232. An important feature of these devices is the SHDN# pin which can power down the device to a low quiescent current during USB suspend mode.

An example of a device which can be used for this is a Sipex SP213EHCA which is capable of RS232 communication at up to 500kΩ baud. If a lower baud rate is acceptable, then several pin compatible alternatives are available such as the Sipex SP213ECA , the Maxim MAX213CAI and the Analog Devices ADM213E, which are all good for communication at up to 115,200 baud. If a higher baud rate is desired, use a Maxim MAX3245CAI part which is capable of RS232 communication at rates of up to 1M baud. The MAX3245 is not pin compatible with the 213 series devices, also its SHDN pin is active high, so connect it to PWREN# instead of SLEEP#.

In the above example CBUS0 and CBUS1 have been configured as TXLED# and RXLED#, and are being used to drive two LEDs.

8.2 USB to RS485 Converter Configuration

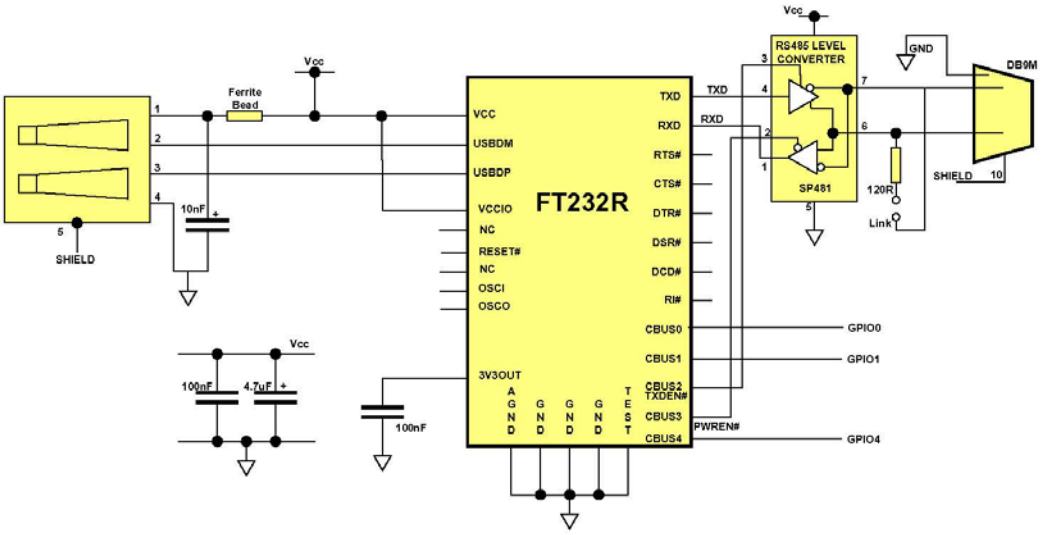


Figure 16 - Example USB to RS485 Converter Configuration

Figure 16 illustrates how to connect the FT232R's UART interface to a TTL – RS485 Level Converter I.C. to make a USB to RS485 converter. This example uses the Sipex SP481 device but there are similar parts available from Maxim and Analog Devices amongst others. The SP481 is a RS485 device in a compact 8 pin SOP package. It has separate enables on both the transmitter and receiver. With RS485, the transmitter is only enabled when a character is being transmitted from the UART. The TXDEN signal CBUS pin option on the FT232R is provided for exactly this purpose and so the transmitter enable is wired to CBUS2 which has been configured as TXDEN. Similarly, CBUS3 has been configured as PWREN#. This signal is used to control the SP481's receiver enable. The receiver enable is active low, so it is wired to the PWREN# pin to disable the receiver when in USB suspend mode. CBUS2 = TXDEN and CBUS3 = PWREN# are the default device configurations of these pins. See Section 10.

RS485 is a multi-drop network – i.e. many devices can communicate with each other over a single two wire cable connection. The RS485 cable requires to be terminated at each end of the cable. A link is provided to allow the cable to be terminated if the device is physically positioned at either end of the cable.

In this example the data transmitted by the FT232R is also received by the device that is transmitting. This is a common feature of RS485 and requires the application software to remove the transmitted data from the received data stream. With the FT232R it is possible to do this entirely in hardware – simply modify the schematic so that RXD of the FT232R is the logical OR of the SP481 receiver output with TXDEN using an HC32 or similar logic gate.

8.3 USB to RS422 Converter Configuration

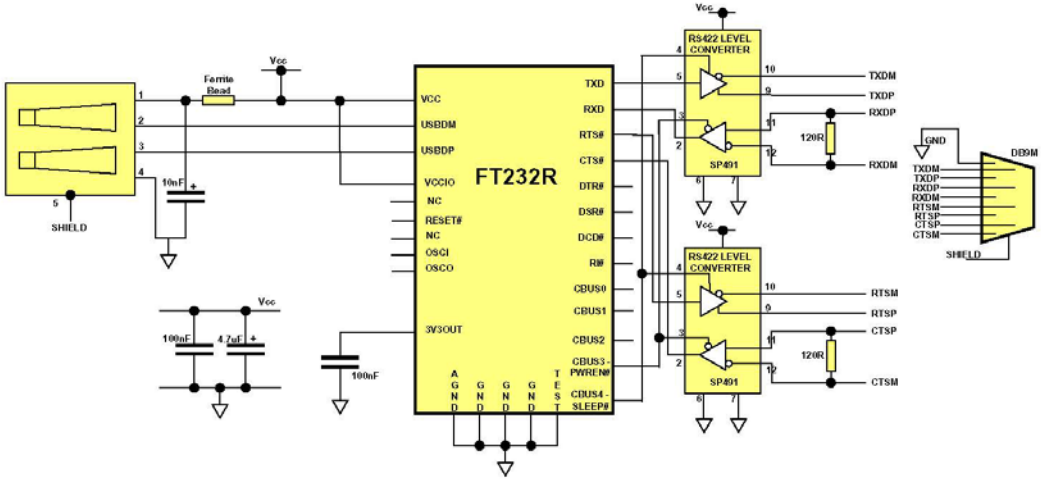


Figure 17 -Example USB to RS422 Converter Configuration

Figure 17 illustrates how to connect the UART interface of the FT232R to a TTL – RS422 Level Converter I.C. to make a USB to RS422 converter. There are many such level converter devices available – this example uses Sipex SP491 devices which have enables on both the transmitter and receiver. Because the transmitter enable is active high, it is connected to a CBUS pin in SLEEP# configuration. The receiver enable is active low and so is connected to a CBUS pin PWREN# configuration. This ensures that both the transmitters and receivers are enabled when the device is active, and disabled when the device is in USB suspend mode. If the design is USB BUS powered, it may be necessary to use a P-Channel logic level MOSFET (controlled by PWREN#) in the VCC line of the SP491 devices to ensure that the USB stand-by current of 500µA is met.

The SP491 is good for sending and receiving data at a rate of up to 5 Megbaud – in this case the maximum rate is limited to 3 Megabaud by the FT232R.

8.4 USB to MCU UART Interface

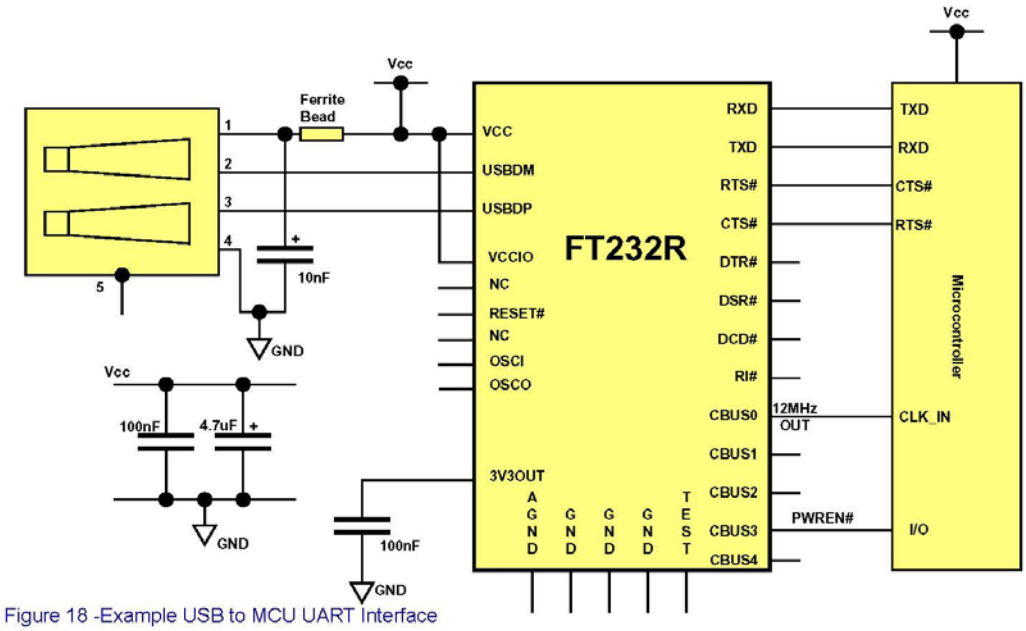


Figure 18 -Example USB to MCU UART Interface

Figure 18 is an example of interfacing the FT232R to a Microcontroller (MCU) UART interface. This example uses TXD and RXD for transmission and reception of data, and RTS# / CTS# hardware handshaking. Also in this example CBUS0 has been configured as a 12MHz output which is being used to clock the MCU.

Optionally, RI# can be connected to another I/O pin on the MCU and could be used to wake up the USB host controller from suspend mode. If the MCU is handling power management functions, then a CBUS pin can be configured as PWREN# and should also be connected to an I/O pin of the MCU.

9. LED Interface

Any of the 5 CBUS I/O pins can be configured to drive an LED. The FT232R has 3 options for driving an LED - these are TXLED#, RXLED#, and TX&RXLED#.

Figure 19 -Dual LED Configuration

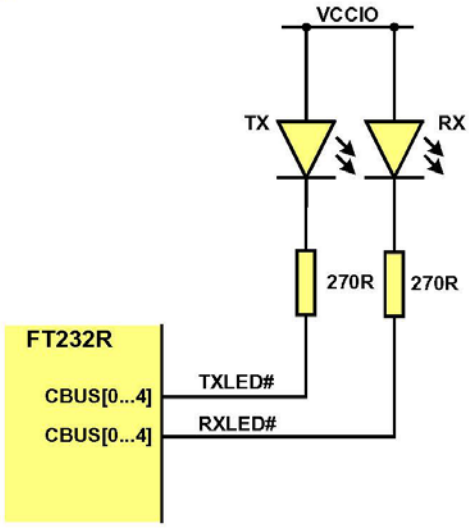


Figure 19 illustrates the configuration where one pin is used to indicate transmission of data (TXLED#) and another is used to indicate receiving data (RXLED#). When data is being transmitted or received the respective pins will drive from tri-state to low in order to provide indication on the LEDs of data transfer. A digital one-shot time is used so that even a small percentage of data transfer is visible to the end user.

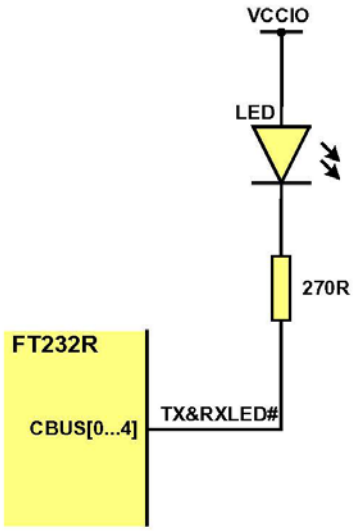


Figure 20 -Single LED Configuration

In figure 20 the TX&RXLED CBUS option is used. This option will cause the pin to drive a single LED when data is being transmitted or received by the device.

10. Internal EEPROM Configuration

Following a power-on reset or a USB reset the FT232R will scan its internal EEPROM and read the USB configuration descriptors stored there. The default values programmed into the internal EEPROM in a brand new device are defined in Table 18.

Table 18 - Default Internal EEPROM Configuration

Parameter	Value	Notes
USB Vendor ID (VID)	0403h	FTDI default VID (hex)
USB Product ID (PID)	6001h	FTDI default PID (hex)
Serial Number Enabled?	Yes	
Serial Number	*See Note*	A unique serial number is generated and programmed into the EEPROM during device final test.
Pull Down I/O Pins in USB Suspend	Disabled	Enabling this option will make the device pull down on the UART interface lines when the power is shut off (PWREN# is high)
Manufacturer Name	FTDI	
Manufacturer ID	FT	Serial number prefix
Product Description	FT232R USB UART	
Max Bus Power Current	90mA	
Power Source	Bus Powered	
Device Type	FT232R	
USB Version	0200	Returns USB 2.0 device descriptor to the host. Note: The device is be a USB 2.0 Full Speed device (12Mb/s) as opposed to a USB 2.0 High Speed device (480Mb/s).
Remote Wake up	Enabled	Taking RI# low will wake up the USB host controller from suspend.
High Current I/Os	Disabled	Enables the high drive level on the UART and CBUS I/O pins
Load VCP Driver	Enabled	Makes the device load the VCP driver interface for the device.
CBUS0	TXLED#	Default configuration of CBUS0 - Transmit LED drive
CBUS1	RXLED#	Default configuration of CBUS1 - Receive LED drive
CBUS2	TXDEN	Default configuration of CBUS2 - Transmit data enable for RS485
CBUS3	PWREN#	Default configuration of CBUS3 - Power enable. Low after USB enumeration, high during USB suspend.
CBUS4	SLEEP#	Default configuration of CBUS4 - Low during USB suspend.
Invert TXD	Disabled	Signal on this pin becomes TXD# if enabled.
Invert RXD	Disabled	Signal on this pin becomes RXD# if enabled.
Invert RTS#	Disabled	Signal on this pin becomes RTS if enabled.
Invert CTS#	Disabled	Signal on this pin becomes CTS if enabled.
Invert DTR#	Disabled	Signal on this pin becomes DTR if enabled.
Invert DSR#	Disabled	Signal on this pin becomes DSR if enabled.
Invert DCD#	Disabled	Signal on this pin becomes DCD if enabled.
Invert RI#	Disabled	Signal on this pin becomes RI if enabled.

The internal EEPROM in the FT232R can be programmed over USB using the utility program MPROG. MPROG can be downloaded from the FTDI website. Version 2.8a or later is required for the FT232R chip. Users who do not have their own USB Vendor ID but who would like to use a unique Product ID in their design can apply to FTDI for a free block of unique PIDs. Contact FTDI support for this service.

Disclaimer

Copyright © Future Technology Devices International Limited , 2005.

Version 0.90 - Initial Datasheet Created August 2005
Version 0.96 - Revised Pre-release datasheet October 2005
Version 1.00 - Full datasheet released December 2005
Version 1.02 - Minor revisions to datasheet 7th December 2005
Version 1.03 - 9th January 2006 - Manufacturer ID added to default EEPROM configuration; Buffer sizes added.
Version 1.04 - 30th January 2006 - QFN-32 Pad layout and solder paste diagrams added.

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied.

Future Technology Devices International Ltd. will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected.

This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury.

This document provides preliminary information that may be subject to change without notice.

Contact FTDI

Head Office -
Future Technology Devices International Ltd.
373 Scotland Street,
Glasgow G5 8QB,
United Kingdom

Tel. : +(44) 141 429 2777
Fax. : +(44) 141 429 2758

E-Mail (Sales) : **sales1@ftdichip.com**
E-Mail (Support) : **support1@ftdichip.com**
E-Mail (General Enquiries) : **admin1@ftdichip.com**

Regional Sales Offices -

Future Technology Devices International Ltd. (Taiwan)
4F, No 16-1,
Sec. 6 Mincyuan East Road,
Neihu District,
Taipei 114,
Taiwan, R.o.C.

Tel.: +886 2 8791 3570
Fax: +886 2 8791 3576

E-Mail (Sales): **tw.sales@ftdichip.com**
E-Mail (Support): **tw.support@ftdichip.com**
E-Mail (General Enquiries): **tw.admin@ftdichip.com**

Future Technology Devices International Ltd. (USA)
5285 NE Elam Young
Parkway, Suite B800
Hillsboro,
OR 97124-6499
USA

Tel.: +1 (503) 547-0988
Fax: +1 (503) 547-0987

E-Mail (Sales): **us.sales@ftdichip.com**
E-Mail (Support): **us.support@ftdichip.com**
E-Mail (General Enquiries): **us.admin@ftdichip.com**

Website URL : **http://www.ftdichip.com**

N-channel enhancement mode TrenchMOS™ transistor

IRFZ44N

GENERAL DESCRIPTION

N-channel enhancement mode standard level field-effect power transistor in a plastic envelope using '**trench**' technology. The device features very low on-state resistance and has integral zener diodes giving ESD protection up to 2kV. It is intended for use in switched mode power supplies and general purpose switching applications.

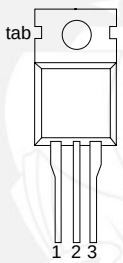
QUICK REFERENCE DATA

SYMBOL	PARAMETER	MAX.	UNIT
V_{DS}	Drain-source voltage	55	V
I_D	Drain current (DC)	49	A
P_{tot}	Total power dissipation	110	W
T_j	Junction temperature	175	˚C
$R_{DS(ON)}$	Drain-source on-state resistance V_{GS} = 10 V	22	mΩ

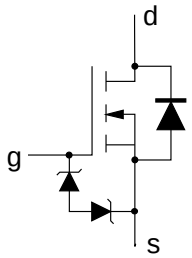
PINNING - TO220AB

PIN	DESCRIPTION
1	gate
2	drain
3	source
tab	drain

PIN CONFIGURATION



SYMBOL



LIMITING VALUES

Limiting values in accordance with the Absolute Maximum System (IEC 134)

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{DS}	Drain-source voltage	-	-	55	V
V_{DGR}	Drain-gate voltage	R_{GS} = 20 kΩ	-	55	V
$\pm V_{GS}$	Gate-source voltage	-	-	20	V
I_D	Drain current (DC)	T_{mb} = 25 ˚C	-	49	A
I_D	Drain current (DC)	T_{mb} = 100 ˚C	-	35	A
I_{DM}	Drain current (pulse peak value)	T_{mb} = 25 ˚C	-	160	A
P_{tot}	Total power dissipation	T_{mb} = 25 ˚C	-	110	W
T_{stg}, T_j	Storage & operating temperature	-	- 55	175	˚C

ESD LIMITING VALUE

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_C	Electrostatic discharge capacitor voltage, all pins	Human body model (100 pF, 1.5 kΩ)	-	2	kV

THERMAL RESISTANCES

SYMBOL	PARAMETER	CONDITIONS	TYP.	MAX.	UNIT
$R_{th\ j\text{-}mb}$	Thermal resistance junction to mounting base	-	-	1.4	K/W
$R_{th\ j\text{-}a}$	Thermal resistance junction to ambient	in free air	60	-	K/W

STATIC CHARACTERISTICS

T_j= 25˚C unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$V_{(BR)DSS}$	Drain-source breakdown voltage	V_{GS} = 0 V; I_D = 0.25 mA;	55	-	-	V
		T_j = -55˚C	50	-	-	V
$V_{GS(TO)}$	Gate threshold voltage	V_{DS} = V_{GS}; I_D = 1 mA	2.0	3.0	4.0	V
		T_j = 175˚C	1.0	-	-	V
		T_j = -55˚C	-	-	4.4	
I_{DSS}	Zero gate voltage drain current	V_{DS} = 55 V; V_{GS} = 0 V;	-	0.05	10	µA
		T_j = 175˚C	-	-	500	µA
I_{GSS}	Gate source leakage current	V_{GS} = ±10 V; V_{DS} = 0 V	-	0.04	1	µA
		T_j = 175˚C	-	-	20	µA
$±V_{(BR)GSS}$	Gate source breakdown voltage	I_G = ±1 mA;	16	-	-	V
$R_{DS(ON)}$	Drain-source on-state resistance	V_{GS} = 10 V; I_D = 25 A	-	15	22	mΩ
		T_j = 175˚C	-	-	42	mΩ

DYNAMIC CHARACTERISTICS

T_{mb} = 25˚C unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
g_{fs}	Forward transconductance	V_{DS} = 25 V; I_D = 25 A	6	-	-	S
C_{iss}	Input capacitance	V_{GS} = 0 V; V_{DS} = 25 V; f = 1 MHz	-	1350	1800	pF
C_{oss}	Output capacitance		-	330	400	pF
C_{rss}	Feedback capacitance		-	155	215	pF
Q_g	Total gate charge	V_{DD} = 44 V; I_D = 50 A; V_{GS} = 10 V	-	-	62	nC
Q_{gs}	Gate-cource charge		-	-	15	nC
Q_{gd}	Gate-drain (miller) charge		-	-	26	nC
$t_{d\ on}$	Turn-on delay time	V_{DD} = 30 V; I_D = 25 A;	-	18	26	ns
t_r	Turn-on rise time	V_{GS} = 10 V; R_G = 10 Ω	-	50	75	ns
$t_{d\ off}$	Turn-off delay time	Resistive load	-	40	50	ns
t_f	Turn-off fall time		-	30	40	ns
L_d	Internal drain inductance	Measured from contact screw on tab to centre of die	-	3.5	-	nH
L_d	Internal drain inductance	Measured from drain lead 6 mm from package to centre of die	-	4.5	-	nH
L_s	Internal source inductance	Measured from source lead 6 mm from package to source bond pad	-	7.5	-	nH

REVERSE DIODE LIMITING VALUES AND CHARACTERISTICS

T_j = 25˚C unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I_{DR}	Continuous reverse drain current		-	-	49	A
I_{DRM}	Pulsed reverse drain current		-	-	160	A
V_{SD}	Diode forward voltage	I_F = 25 A; V_{GS} = 0 V	-	0.95	1.2	V
		I_F = 40 A; V_{GS} = 0 V	-	1.0	-	
t_{rr}	Reverse recovery time	I_F = 40 A; -dI_F/dt = 100 A/µs;	-	47	-	ns
Q_{rr}	Reverse recovery charge	V_{GS} = -10 V; V_R = 30 V	-	0.15	-	µC

N-channel enhancement mode TrenchMOS™ transistor

IRFZ44N

AVALANCHE LIMITING VALUE

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
W_{DSS}	Drain-source non-repetitive unclamped inductive turn-off energy	I_D = 45 A; $V_{DD} \leq$ 25 V; V_{GS} = 10 V; R_{GS} = 50 Ω; T_{mb} = 25 ˚C	-	-	110	mJ

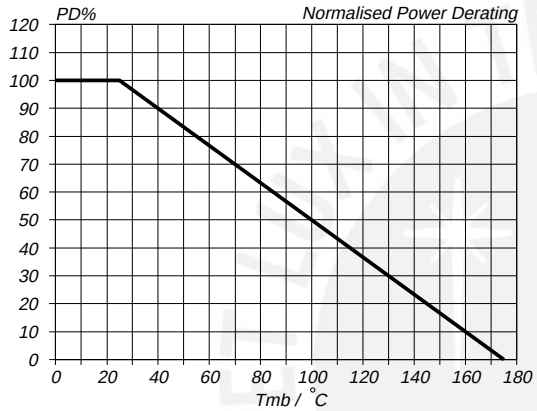


Fig.1. Normalised power dissipation.
PD% = 100·P_D/$P_{D\ 25\ ˚C}$ = f(T_{mb})

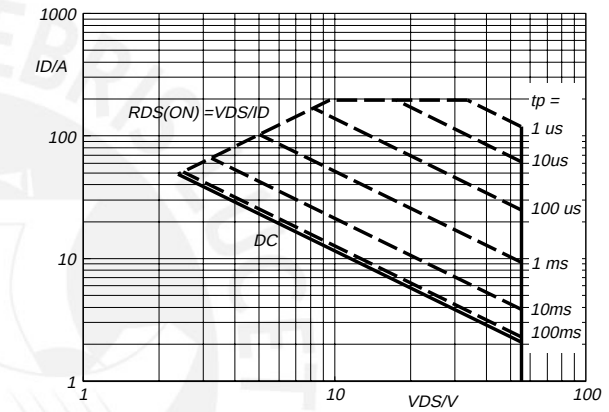


Fig.3. Safe operating area. T_{mb} = 25 ˚C
I_D & I_{DM} = f(V_{DS}); I_{DM} single pulse; parameter t_p

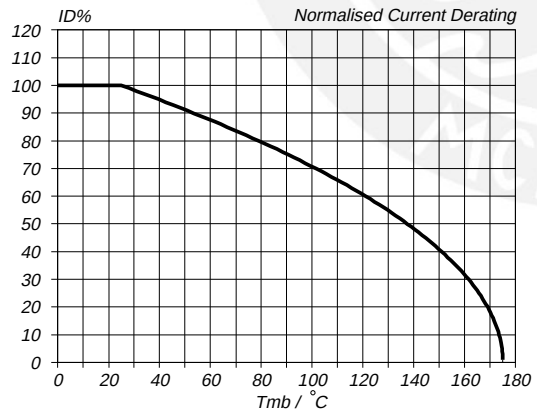


Fig.2. Normalised continuous drain current.
ID% = 100·I_D/$I_{D\ 25\ ˚C}$ = f(T_{mb}); conditions: $V_{GS} \geq$ 10 V

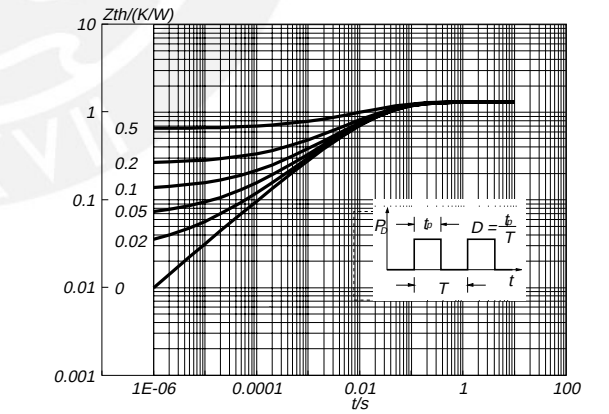


Fig.4. Transient thermal impedance.
$Z_{th\ j\text{-}mb}$ = f(t); parameter D = t_p/T

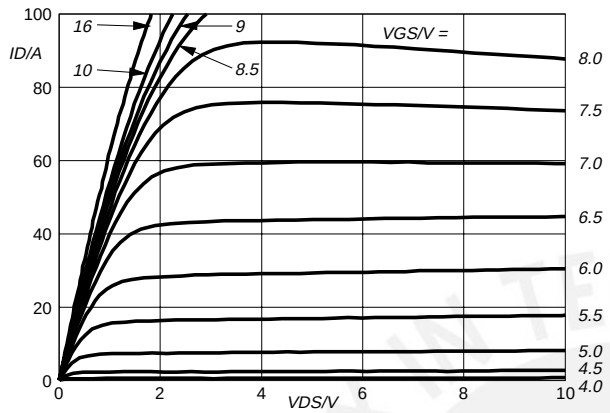


Fig.5. Typical output characteristics, T_j = 25 ˚C. $I_D = f(V_{DS})$; parameter V_{GS}

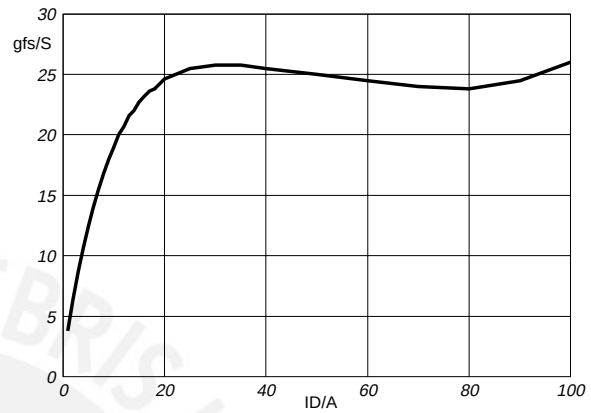


Fig.8. Typical transconductance, T_j = 25 ˚C. $g_{fs} = f(I_D)$; conditions: V_{DS} = 25 V

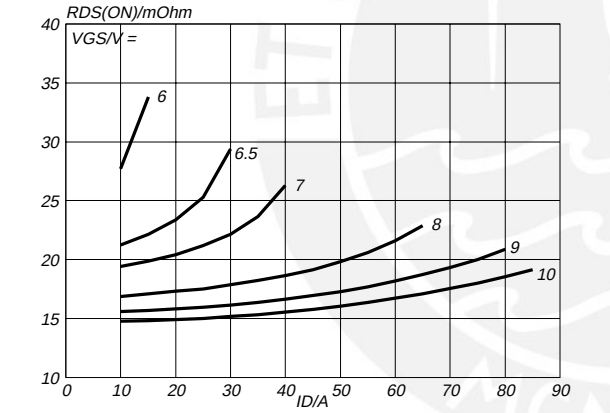


Fig.6. Typical on-state resistance, T_j = 25 ˚C. $R_{DS(ON)} = f(I_D)$; parameter V_{GS}

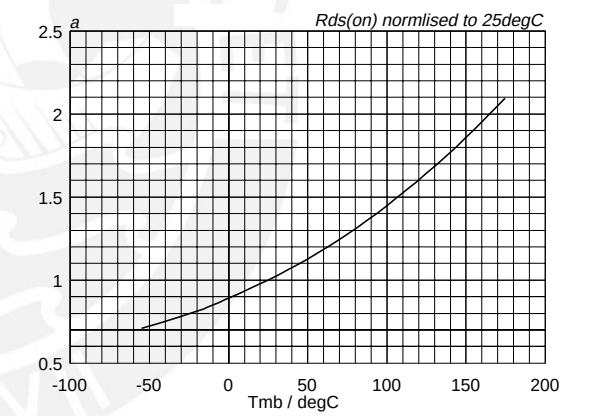


Fig.9. Normalised drain-source on-state resistance. $a = R_{DS(ON)}/R_{DS(ON)25\ ˚C} = f(T_j)$; I_D = 25 A; V_{GS} = 10 V

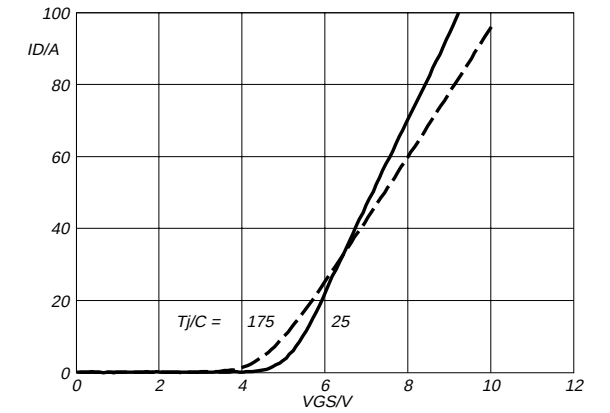


Fig.7. Typical transfer characteristics. $I_D = f(V_{GS})$; conditions: V_{DS} = 25 V; parameter T_j

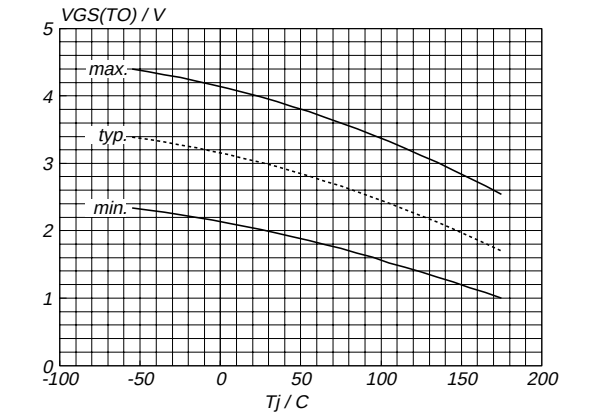


Fig.10. Gate threshold voltage. $V_{GS(TO)} = f(T_j)$; conditions: I_D = 1 mA; $V_{DS} = V_{GS}$

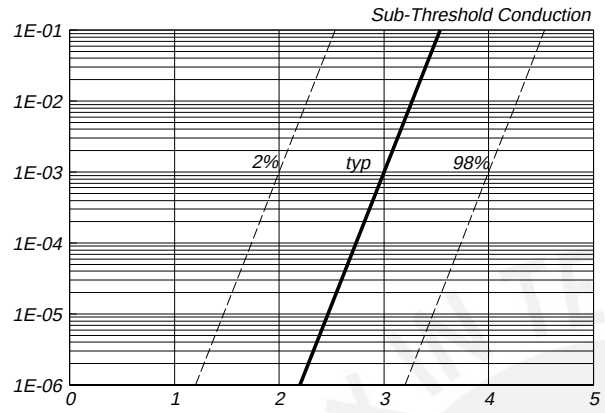


Fig.11. Sub-threshold drain current.
$I_D = f(V_{GS})$; conditions: T_j = 25 ˚C; $V_{DS} = V_{GS}$

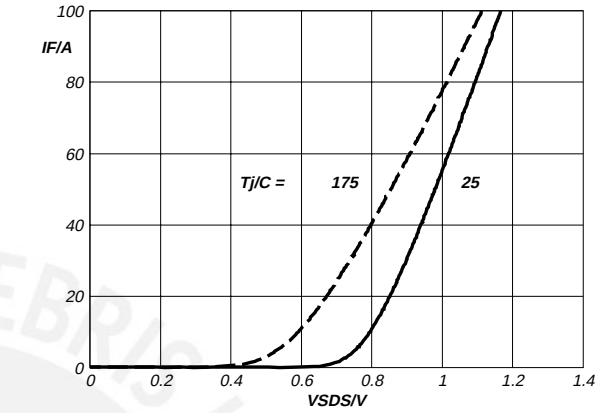


Fig.14. Typical reverse diode current.
$I_F = f(V_{SDS})$; conditions: V_{GS} = 0 V; parameter T_j

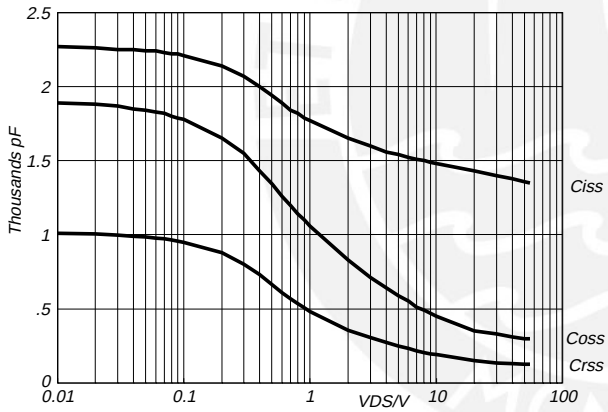


Fig.12. Typical capacitances, C_{iss}, C_{oss}, C_{rss}.
C = $f(V_{DS})$; conditions: V_{GS} = 0 V; f = 1 MHz

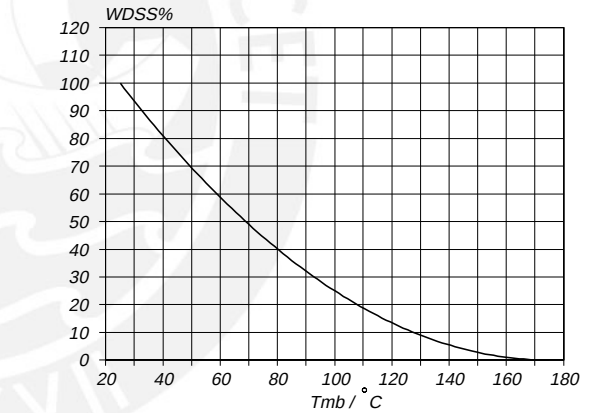


Fig.15. Normalised avalanche energy rating.
W_{DSS}% = $f(T_{mb})$; conditions: I_D = 49 A

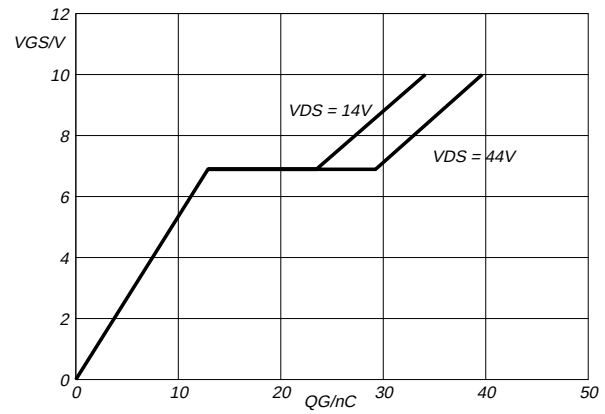


Fig.13. Typical turn-on gate-charge characteristics.
V_{GS} = $f(Q_G)$; conditions: I_D = 50 A; parameter V_{DS}

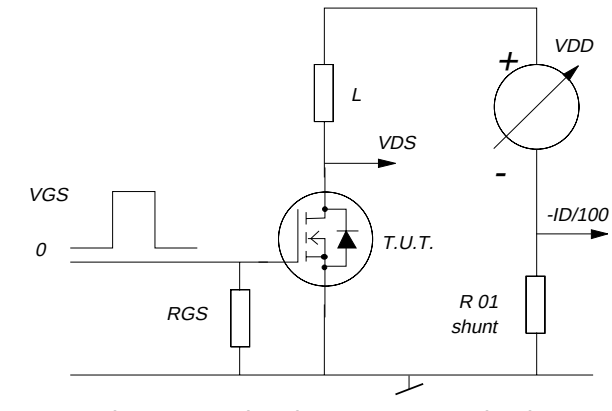


Fig.16. Avalanche energy test circuit.

$$W_{DSS} = 0.5 \cdot LI_D^2 \cdot BV_{DSS}/(BV_{DSS} - V_{DD})$$

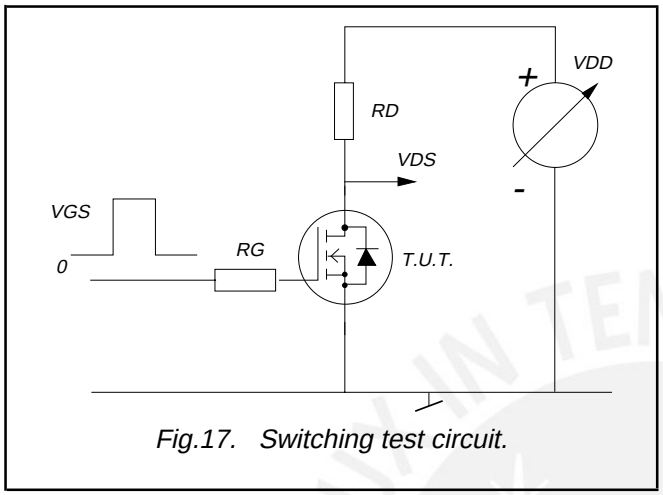


Fig.17. Switching test circuit.

MECHANICAL DATA

Dimensions in mm

Net Mass: 2 g

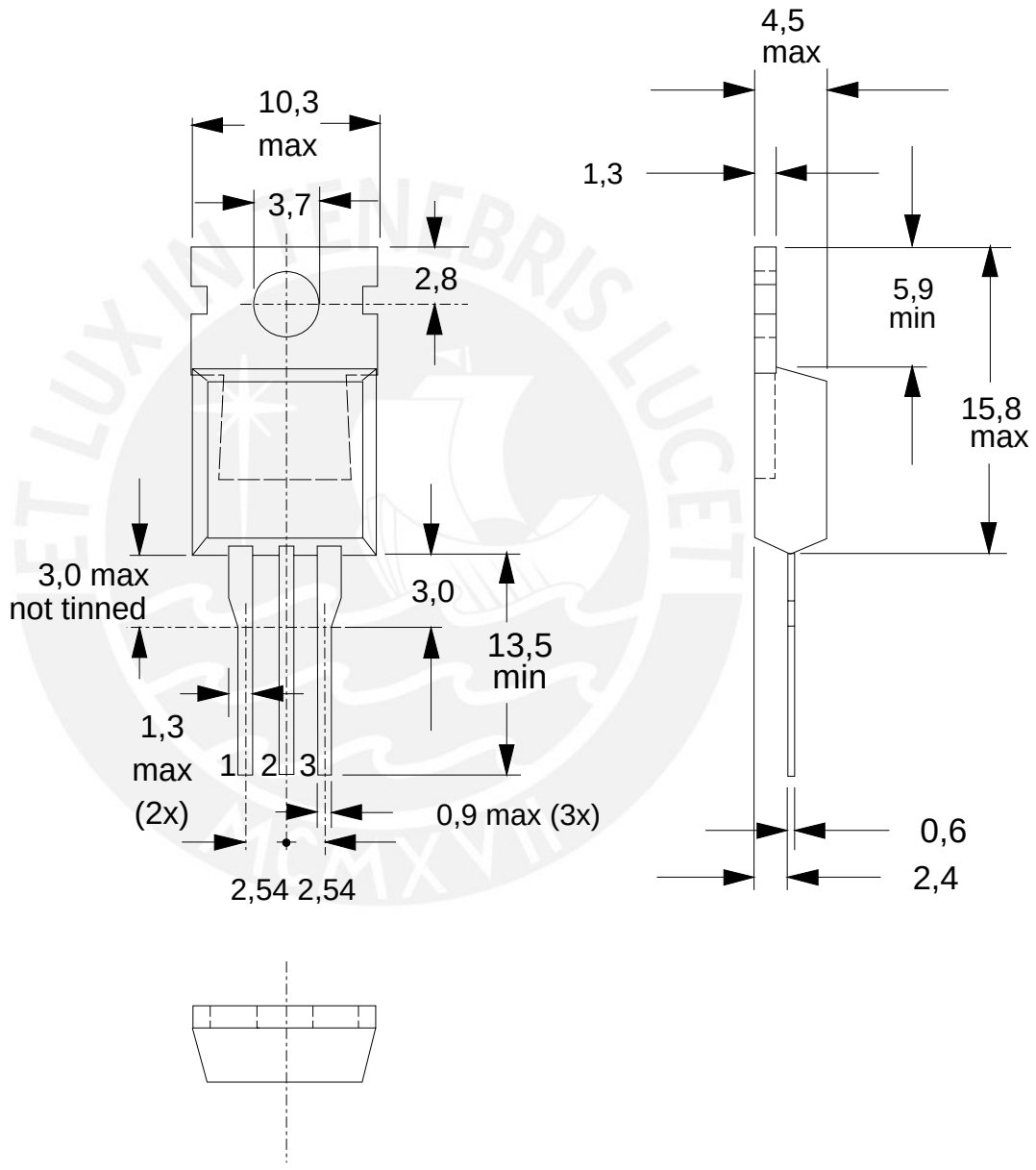


Fig.18. SOT78 (TO220AB); pin 2 connected to mounting base.

Notes

1. Observe the general handling precautions for electrostatic-discharge sensitive devices (ESDs) to prevent damage to MOS gate oxide.
2. Refer to mounting instructions for SOT78 (TO220) envelopes.
3. Epoxy meets UL94 V0 at 1/8".

DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values are given in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of this specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	
The information presented in this document does not form part of any quotation or contract, it is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights.	

LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices or systems where malfunction of these products can be reasonably expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

User's Guide

GDM12864HLCM

(Liquid Crystal Display Module)

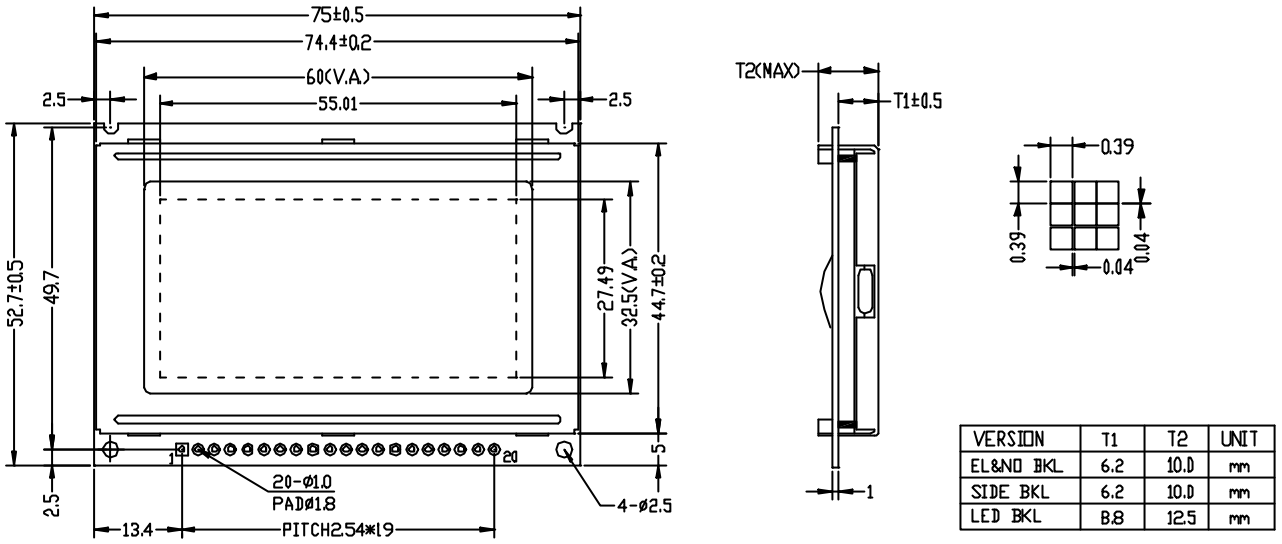
For product support, contact

XIAMEN OCULAR OPTICS CO.,LTD.
厦门高卓立光电有限公司
South2/F.,Guangxia Building Torch Hi-tech
Industrial Development Area Xiamen, China
36100 **中国厦门火炬高技术产业开发区光厦楼南二楼**
Tel: 86-592-5650516 Fax: 86-592-5650695

CONTENTS

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

➢ Mechanical diagram



VERSION	T1	T2	UNIT
EL&NO BKL	6.2	10.0	mm
SIDE BKL	6.2	10.0	mm
LED BKL	8.8	12.5	mm

➢ Absolute maximum ratings

Item	Symbol	Min.	Max.	Unit
Supply voltage for logic	Vdd - Vss	0	6.5	V
Input voltage	Vin	0	Vdd	
Operating temperature range	T0p	-20	70	℃
Storage temperature range	Tst	-25	75	

➢ Interface pin connections

Pin No.	Symbol	Level	Description
1	Vdd	5.0V	Supply voltage for logic and LCD (+)
2	Vss	0V	Ground
3	V0	-	Operating voltage for LCD (variable)
4~11	DB0~DB7	H/L	Data bit 0~7
12	CS2	L	Chip select signal for IC2
13	CS1	L	Chip select signal for IC1
14	/RES	L	Reset signal
15	R/W	H/L	H: read (MUP<－module),L: write (MPU－>module)
16	D/I	H/L	H: data, L: instruction code
17	E	H, H　L	Chip enable signal
18	VEE	-	Operating voltage for LCD (variable)
19	A	4.2V	Backlight power supply
20	K	0V	Backlight power supply

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA　TEL: 86-592-5650516　FAX: 86-592-5650695

➢ Optical characteristics

STN Type display module (Ta=25℃, Vdd=5.0V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Viewing angle	θ	Cr≥2	-60	-	35	deg
	ϕ		-40	-	40	
Contrast ratio (rise)	Cr		-	6	-	
Response time (fall)	Tr	-	-	150	250	ms
	Tr	-	-	150	250	ms

➢ Electrical characteristics

Item		Symbol	Condition	Standard value			Unit
				Min.	Typ.	Max.	
Supply voltage for	Logic	Vdd - Vss	-	4.75	5.0	5.25	V
	LCD	Vdd-V0	-	-	9.5	-	
Supply current for	Logic	Idd	-	-	2.5	-	mA
	LCD	Iee	-	-	1.0	-	
Operating voltage for LCD (Recommended)		Vdd-v0	-	-	-	-	V
			25℃	-	9.5	-	
			-	-	-	-	
Input voltage	H: level	Vih	High level	0.7Vdd	-	Vdd	
	L: Level	Vil	Low level	0	-	0.3Vdd	

Electrical Absolute Maximum Ratings (KS0107B)

Parameter	Symbol	Rating	Unit	Note
Operating voltage	V_{DD}	-0.3 ~ +7.0	V	*1
Supply voltage	V_{EE}	V_{DD}-19.0 ~ V_{DD}+0.3	V	*4
Driver supply voltage	V_B	-0.3 ~ V_{DD}+0.3	V	*1,2
	V_{LCD}	V_{EE}-0.3 ~ V_{DD}+0.3	V	*3,4

***Notes:**

*1. Based on V_{SS} = 0V

*2. Applies to input terminals and I/O terminals at high impedance. (Except V0L, V1L, V4L, and V5L)

*3. Applies to V0L, V1L, V4L, and V5L.

*4. Voltage level: $V_{DD} \geq V0 \geq V1 \geq V2 \geq V3 \geq V4 \geq V5 \geq V_{EE}$

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

DC Electrical Characteristics (KS0107B)

(VDD= 4.5 to 5.5V, VSS=0V,VDD-VEE=8~17V,Ta= -30 to +85℃)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Operating voltage	V_{DD}	-	4.5	-	5.5	V	
Input voltage	V_{IH}	-	0.7_{VDD}	-	V_{DD}		*1
	V_{IL}	-	V_{SS}	-	$0.3V_{DD}$		
output voltage	V_{OH}	I_{OH}= -0.4mA	V_{DD}-0.4	-	-		*2
	V_{OL}	I_{OL}= 0.4mA	-	-	0.4		
Input leakage current	I_{LKG}	V_{IN}= V_{DD} ~ V_{SS}	-1.0	-	+1.0	μA	*1
OSC Frequency	fosc	Rf=47kΩ±2% Cf=20pF±5%	315	450	585	kHz	
On Resistance (Vdiv-Ci)	R_{ONS}	V_{DD}-V_{EE}=17V Load current±150μA	-	-	1.5	kΩ	
Operating current	I_{DD1}	Master mode 1/128 Duty	-	-	1.0	mA	*3
	I_{DD2}	Master mode 1/128 Duty	-	-	0.2		*4
Supply Current	I_{EE}	Master mode 1/128 Duty	-	-	0.1		*5
Operating	f_{op1}	Master mode External Duty	50	-	600	kHz	
Frequency	f_{op2}	Slave mode	0.5	-	1500		

Notes

*1. Applies to input terminals FS, DS1, DS2, CR, SHL, MS and PCLK2 and I/O terminals DIO1, DIO2, M, and CL2 in the input state.

*2. Applies to output terminals CLK1, CLK2 and FRM and I/O terminals DIO1, DIO2, M, and CL2 in the output state.

*3. This value is specified about current flowing through V_{SS}.
Internal oscillation circuit: Rf=47kΩ, cf=20pF
Each terminals of DS1, DS2, FS, SHL, and MS is connected to V_{DD} and out is no load.

*4. This value is specified about current flowing through V_{SS}.
Each terminals is DS1, DS2, FS, SHL, PCLK2 and CR is connected to V_{DD},MS is connected to V_{SS} and CL2, M, DIO1 is external clock.

*5. This value is specified about current flowing through V_{EE}, Don' t connect to V_{LCD} (V1~V5).

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

Electrical Absolute Maximum Ratings (KS0108B)

Parameter	Symbol	Rating	Unit	Note
Operating voltage	V_{DD}	-0.3 ~ +7.0	V	*1
Supply voltage	V_{EE}	V_{DD}-19.0 ~ V_{DD}+0.3	V	*4
Driver supply voltage	V_B	-0.3 ~ V_{DD}+0.3	V	*1,3
	V_{LCD}	V_{EE}-0.3 ~ V_{DD}+0.3	V	*2

***Notes:**

*1. Based on V_{SS} = 0V
*2. Applies the same supply voltage to V_{EE}. $V_{LCD}=V_{DD}-V_{EE}$.
*3. Applies to M, FRM, CLK1, CLK2, CL, RESETB, ADC, CS1B, CS2B, CS3, E, R/W, RS and DB0~DB7.
*4. Applies V0L, V2L, V3L and V5L.

Voltage level: $V_{DD} \geq V0 \geq V1 \geq V2 \geq V3 \geq V4 \geq V5 \geq V_{EE}$

DC Electrical Characteristics (KS0108B)

(VDD= 4.5 to 5.5V, VSS=0V,VDD-VEE=8~17V,Ta= -30 to +85℃)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Operating voltage	V_{DD}	-	4.5	-	5.5	V	
Input High voltage	V_{IH1}	-	0.7_{VDD}	-	V_{DD}		*1
	V_{IH2}	-	2.0	-	V_{DD}		*2
Input Low voltage	V_{IL1}	-	0	-	$0.3V_{DD}$		*1
	V_{IL2}	-	0	-	0.8		*2
Output High Voltage	V_{OH}	I_{OH}= -0.2mA	2.4	-	-		*3
Output Low Voltage	V_{OL}	I_{OL}= 1.6mA	-	-	0.4		*3
Input leakage current	I_{LKG}	$V_{IN}= V_{SS} \sim V_{DD}$	-1.0	-	+1.0	μA	*4
Three-state (OFF) Input Current	I_{TSL}	$V_{IN}= V_{SS} \sim V_{DD}$	-5.0	-	5.0		*5
Driver Input leakage current	I_{DIL}	$V_{IN}= V_{EE} \sim V_{DD}$	-2.0		2.0		*6
On Resistance (Vdiv-Ci)	R_{ONS}	$V_{DD}-V_{EE}$=15V Load current±100μA	-	-	7.5	kΩ	*8
Operating current	I_{DD1}	During Display	-	-	0.1	mA	*7
	I_{DD2}	During Access Access Cycle=1MHz	-	-	0.5		*7

Notes

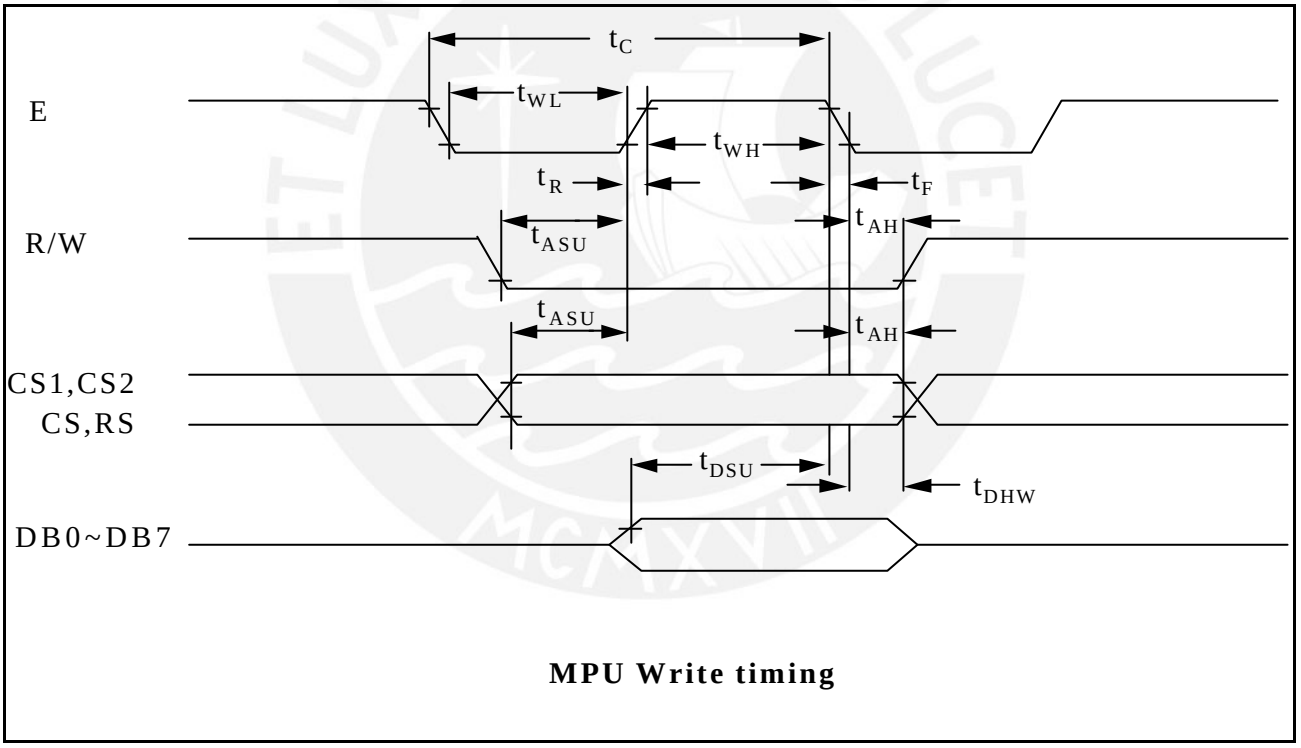
*1. CL, FRM, M, RSTB, CLK1, CLK2
*2. CS1B, CS2B, CS3, E, R/W, RS, DB0~DB7
*3. DB0~DB7
*4. Except DB0~DB7
*5. DB0~DB7 at high impedance
*6. V0, V1, V3, V3, V4, V5
*7. 1/64 duty, FCLK=250KHZ, Frame Frequency=70HKZ, Output: No Load
*8. $V_{DD}-V_{EE}$=15.5V
V0L>V2L>= V_{DD}-2/7($V_{DD}-V_{EE}$)>V3L= V_{EE}+2/7($V_{DD}-V_{EE}$)>V5L

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

➢ Write or read cycle

Characteristic	Symbol	Min.	Typ.	Max.	Unit
E cycle	Tc	1000	-	-	ns
E high level width	Twh	450	-	-	ns
E low level width	Twl	450	-	-	ns
E rise time	Tr	-	-	25	ns
E fall time	Tf	-	-	25	ns
Address set-up time	Tasu	140	-	-	ns
Address hold time	Tah	10	-	-	ns
Data set-up time	Tdsu	200	-	-	ns
Data delay time	Td	-	-	320	ns
Data hold time (write)	Tdhw	10	-	-	ns
Data hold time (read)	Tdhr	20	-	-	ns

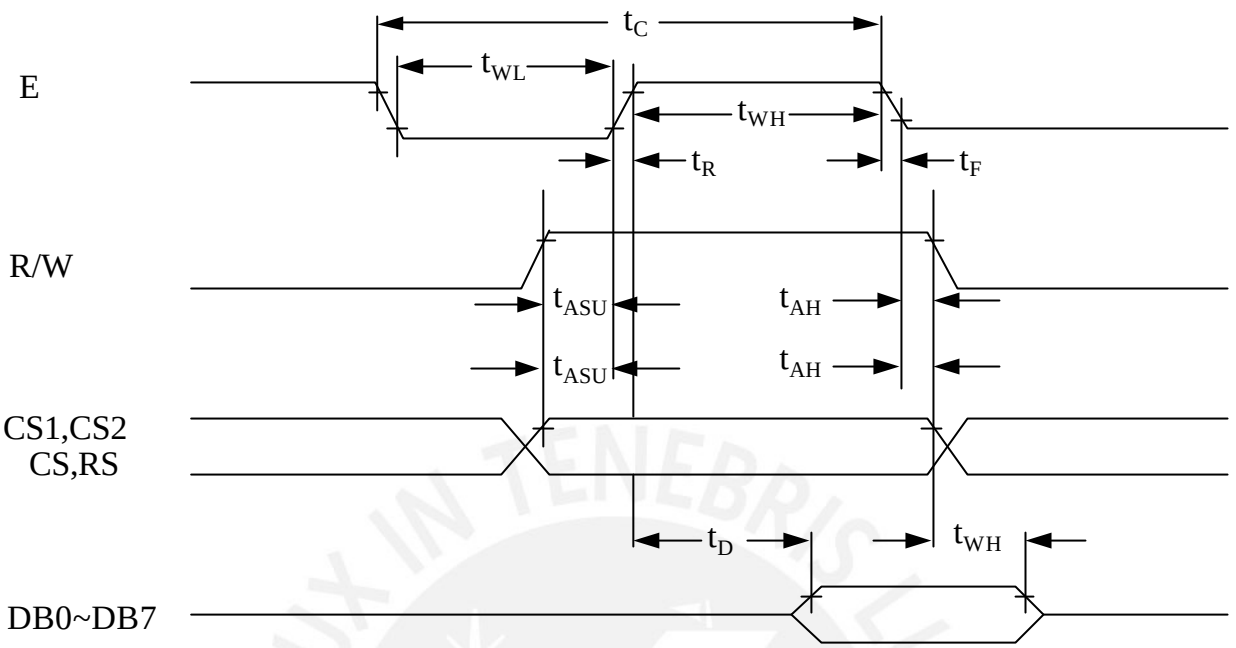
✧ Write timing



MPU Write timing

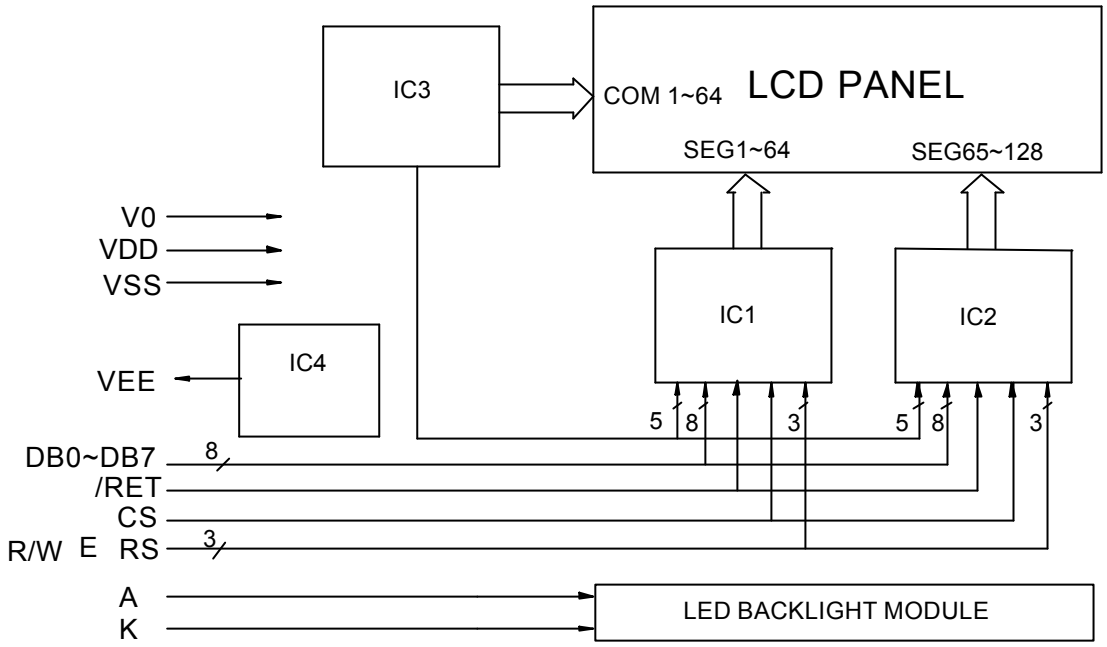
SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

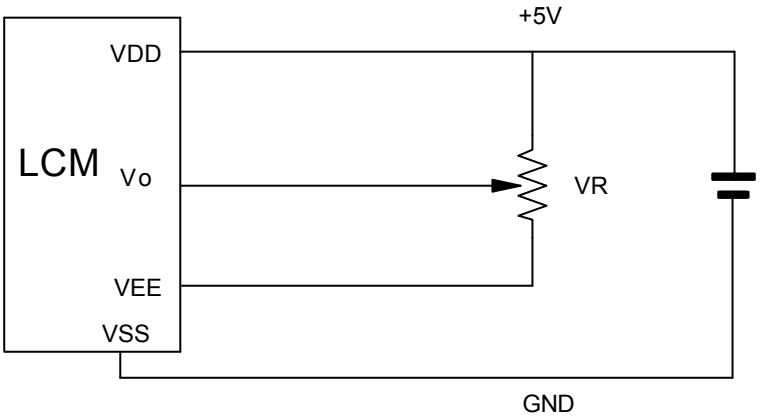
✧ Read timing



MPU Read timing

✧ Block diagram





VDD-Vo:LCD DRIVING VOLTAGE
VR:10K~20K

***Note**

1/64 duty, 1/9 bias

$V_{DD} \geq V1 \geq V2 \geq V3 \geq V4 \geq V5 \geq V_{EE}$

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

✧ Display Control Instruction

The display control instructions control the internal state of the KS0108B. Instruction is received from MPU to KS0108B for the display control. The following table shows various instructions.

Instruction	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Read Display Data	1	1	Read data								Reads data (DB[7:0])from display data RAM to the data bus.
Write Display Data	1	0	Write data								Writes data (DB[7:0]) into display data RAM. After writing instruction, Y address is incriminated by 1 automatically
Status Read	0	1	Busy	0	ON/ OFF	Re-set	0	0	0	0	Reads the internal status BUSY 0: Ready 1: In operation ON/OFF 0: Display ON 1: Display OFF RESET 0: Normal 1: Reset
Set Address (Y address)	0	0	0	1	Y address (0~63)						Sets the Y address in the Y address counter
Set Display Start Line	0	0	1	1	Display start line (0~63)						Indicates the display data RAM displayed at the top of the screen.
Set Address (X address)	0	0	1	0	1	1	1	Page (0~7)			Sets the X address at the X address register.
Display On/off	0	0	0	0	1	1	1	1	1	0/1	Controls the display ON or OFF. The internal status and the DDRAM data is not affected. 0: OFF, 1: ON

1. Display On/Off

The display data appears when D is 1 and disappears when D is 0.
Though the data is not on the screen with D=0, it remains in the display data RAM.
Therefore, you can make it appear by changing D=0 into D=1.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	D

2. Set Address (Y Address)

Y address (AC0~AC5) of the display data RAM is set in the Y address counter.
An address is set by instruction and increased by 1 automatically by read or write operations of display data.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

3. Set Page (X Address)

X address (AC0~AC2) of the display data RAM is set in the X address register.
Writing or reading to or from MPU is executed in this specified page until the next page is set.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	1	1	AC2	AC1	AC0

4. Display Start Line (Z Address)

Z address (AC0~AC5) of the display data RAM is set in the display start line register and displayed at the top of the screen.
When the display duty cycle is 1/64 or others (1/32~1/64), the data of total line number of LCD screen, from the line specified by display start line instruction, is displayed.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	AC5	AC4	AC3	AC2	AC1	AC0

5. Status Read

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	BUSY	0	ON/OFF	RESET	0	0	0	0

- BUSY
  When BUSY is 1, the Chip is executing internal operation and no instructions are accepted.
  When BUSY is 0, the Chip is ready to accept any instructions.
- ON/OFF
  When ON/OFF is 1, the display is on.
  When ON/OFF is 0, the display is off.
- RESET
  When RESET is 1, the system is being initialized.
  In this condition, no instructions except status read can be accepted.
  When RESET is 0, initializing has finished and the system is in the usual operation condition.

6. Write Display Data

Writes data (D0~D7) into the display data RAM.
After writing instruction, Y address is increased by 1 automatically.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

7. Read Display Data

Reads data (D0~D7) from the display data RAM.
After reading instruction, Y address is increased by 1 automatically.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

✧ Operating principles & methods

1. I/O Buffer

Input buffer controls the status between the enable and disable of chip. Unless the CS1B to CS3 is in active mode, Input or output of data and instruction does not execute. Therefore internal state is not change. But RSTB and ADC can operate regardless CS!B-CS3.

2. Input register

Input register is provided to interface with MPU which is different operating frequency. Input register stores the data temporarily before writing it into display RAM.
When CS1B to CS3 are in the active mode, R/W and RS select the input register. The data from MPU is written into input register. Then writing it into display RAM. Data latched for falling of the E signal and write automatically into the display data RAM by internal operation.

3. Output register

Output register stores the data temporarily from display data RAM when CS1B, CS2B and CS3 are in active mode and R/W and RS=H, stored data in display data RAM is latched in output register. When CS1B to CS3 is in active mode and R/W=H , RS=L, status data (busy check) can read out.
To read the contents of display data RAM, twice access of read instruction is needed. In first access, data in display data RAM is latched into output register. In second access, MPU can read data which is latched. That is to read the data in display data RAM, it needs dummy read. But status read is not needed dummy read.

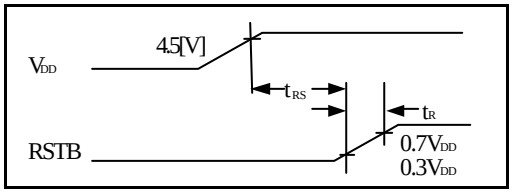
RS	R/W	Function
L	L	Instruction
	H	Status read (busy check)
H	L	Data write (from input register to display data RAM)
	H	Data read (from display data RAM to output register)

4. Reset

The system can be initialized by setting RSTB terminal at low level when turning power on, receiving instruction from MPU. When RSTB becomes low, following procedure is occurred.
1. Display off
2. Display start line register become set by 0. (Z-address 0)
While RSTB is low, No instruction except status read can by accepted. Therefore, execute other instructions after making sure that DB4= (clear RSTB) and DB7=0 (ready) by status read instruction.
The conditions of power supply at initial power up are shown in table 1.

Table 1. Power Supply Initial Conditions

Item	Symbol	Min	Typ	Max	Unit
Reset Time	t_{RS}	1.0	-	-	us
Rise Time	t_R	-	-	200	ns



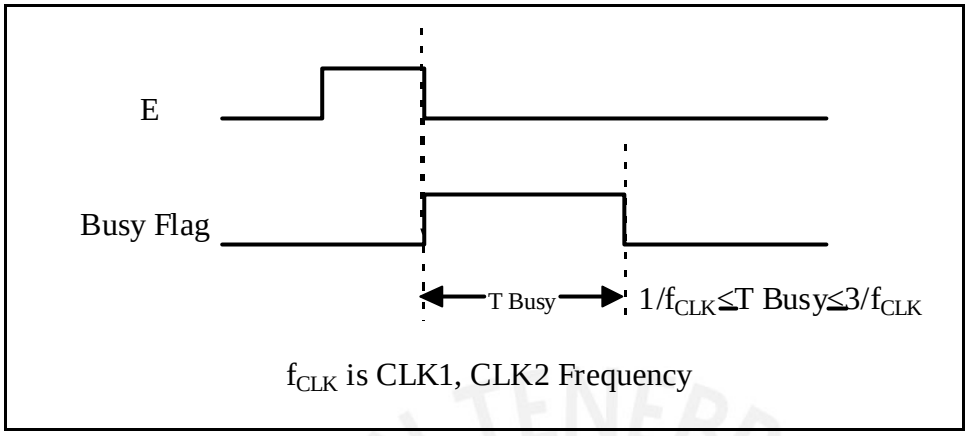
SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

5. Busy flag

Busy flag indicates that KS0108B is operating or no operating. When busy flag is high, KS0108B is in internal operating .

When busy flag is low, KS0108B can accept the data or instruction.

DB7indicates busy flag of the KS0108B.



6. Display On/Off Flip-Flop

The display on/off flip-flop makes on/off the liquid crystal display. When flip-flop is reset (logical low), selective voltage or non-selective voltage appears on segment output terminals. When flip-flop is set (logic high), non selective voltage appears on segment output terminals regardless of display RAM data.

The display on/off flip-flop can changes status by instruction. The display data at all segments disappear while RSTB is low.

The status of the flip-flop is output to DB5 by status read instruction.

The display on/off flip-flop synchronized by CL signal.

7. X Page Register

X page register designates pages of the internal display data RAM.

Count function is not available. An address is set by instruction.

8. Y address counter

Y address counter designates address of the internal display data RAM. An address is set by instruction and is increased by 1 automatically by read or writes operations of display data.

9. Display Data RAM

Display data RAM stores a display data for liquid crystal display. To indicate on state dot matrix of liquid crystal display, write datra1. The other way, off state, writes 0.

Display data RAM address and segment output can be controlled by ADC signal.

ADC=H => Y-address 0: S1~Y address 63: S64

ADC=L => Y-address 0: S64~Yaddress 63: S1

ADC terminal connect the V_{DD} or V_{SS}.

10. Display Start Line Register

The display start line register indicates of display data RAM to display top line of liquid crystal display. Bit data (DB<0.5>) of the display start line set instruction is latched in display start line register. Latched data is transferred to the Z address counter while FRM is high, presetting the Z address counter.

It is used for scrolling of the liquid crystal display screen.

SOUTH2/F, GUANGXIA BUILDING, TORCH HIGH-TECH DEVELOPMENT ARER,
XIAMEN 361006.P.R.CHINA TEL: 86-592-5650516 FAX: 86-592-5650695

FAIRCHILD SEMICONDUCTOR®

August 2012

LM78XX/LM78XXA
3-Terminal 1A Positive Voltage Regulator

Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

General Description

The LM78XX series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

Ordering Information

Product Number	Output Voltage Tolerance	Package	Operating Temperature
LM7805CT	±4%	TO-220 (Single Gauge)	-40°C to +125°C
LM7806CT			
LM7808CT			
LM7809CT			
LM7810CT			
LM7812CT			
LM7815CT			
LM7818CT			
LM7824CT			
LM7805ACT	±2%		0°C to +125°C
LM7806ACT			
LM7808ACT			
LM7809ACT			
LM7810ACT			
LM7812ACT			
LM7815ACT			
LM7818ACT			
LM7824ACT			

Block Diagram

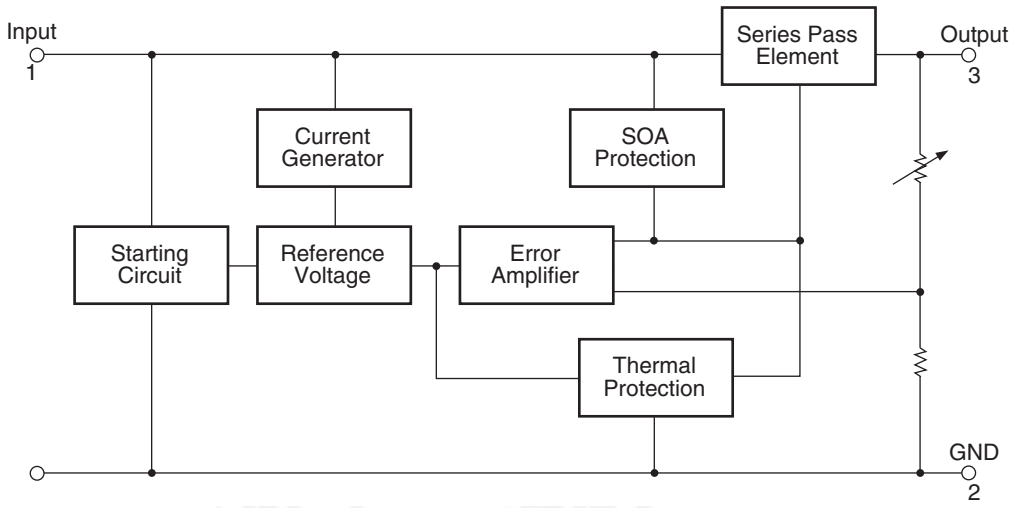


Figure 1.

Pin Assignment

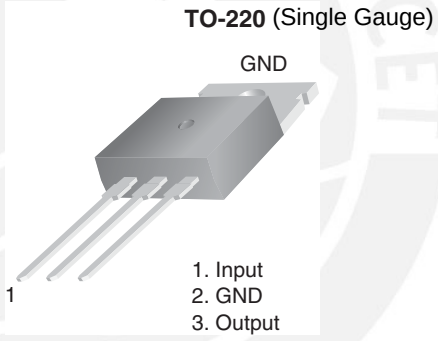


Figure 2.

Absolute Maximum Ratings

Absolute maximum ratings are those values beyond which damage to the device may occur. The datasheet specifications should be met, without exception, to ensure that the system design is reliable over its power supply, temperature, and output/input loading variables. Fairchild does not recommend operation outside datasheet specifications.

Symbol	Parameter		Value	Unit
V_I	Input Voltage	V_O = 5V to 18V	35	V
		V_O = 24V	40	V
$R_{\theta JC}$	Thermal Resistance Junction-Cases (TO-220)		5	°C/W
$R_{\theta JA}$	Thermal Resistance Junction-Air (TO-220)		65	°C/W
T_{OPR}	Operating Temperature Range	LM78xx	-40 to +125	°C
		LM78xxA	0 to +125	
T_{STG}	Storage Temperature Range		-65 to +150	°C

Electrical Characteristics (LM7805)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 10V, C_I = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		4.8	5.0	5.2	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 7V to 20V		4.75	5.0	5.25	
Regline	Line Regulation(1)	T_J = +25°C	V_O = 7V to 25V	–	4.0	100	mV
			V_I = 8V to 12V	–	1.6	50.0	
Regload	Load Regulation(1)	T_J = +25°C	I_O = 5mA to 1.5A	–	9.0	100	mV
			I_O = 250mA to 750mA	–	4.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	0.03	0.5	mA
		V_I = 7V to 25V		–	0.3	1.3	
$\Delta V_O/\Delta T$	Output Voltage Drift(2)	I_O = 5mA		–	-0.8	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	42.0	–	µV/V_O
RR	Ripple Rejection(2)	f = 120Hz, V_O = 8V to 18V		62.0	73.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(2)	f = 1kHz		–	15.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	230	–	mA
I_{PK}	Peak Current(2)	T_J = +25°C		–	2.2	–	A

Notes:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
2. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7806) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 11V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		5.75	6.0	6.25	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 8.0V to 21V		5.7	6.0	6.3	
Regline	Line Regulation(3)	T_J = +25°C	V_I = 8V to 25V	–	5.0	120	mV
			V_I = 9V to 13V	–	1.5	60.0	
Regload	Load Regulation(3)	T_J = +25°C	I_O = 5mA to 1.5A	–	9.0	120	mV
			I_O = 250mA to 750mA	–	3.0	60.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 8V to 25V		–	–	1.3	
$\Delta V_O/\Delta T$	Output Voltage Drift(4)	I_O = 5mA		–	-0.8	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	45.0	–	µV/V_O
RR	Ripple Rejection(4)	f = 120Hz, V_O = 8V to 18V		62.0	73.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(4)	f = 1kHz		–	19.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(4)	T_J = +25°C		–	2.2	–	A

Notes:

3. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
4. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7808) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 14V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		7.7	8.0	8.3	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 10.5V to 23V		7.6	8.0	8.4	
Regline	Line Regulation(5)	T_J = +25°C	V_I = 10.5V to 25V	–	5.0	160	mV
			V_I = 11.5V to 17V	–	2.0	80.0	
Regload	Load Regulation(5)	T_J = +25°C	I_O = 5mA to 1.5A	–	10.0	160	mV
			I_O = 250mA to 750mA	–	5.0	80.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	0.05	0.5	mA
		V_I = 10.5V to 25V		–	0.5	1.0	
$\Delta V_O/\Delta T$	Output Voltage Drift(6)	I_O = 5mA		–	-0.8	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	52.0	–	µV/V_O
RR	Ripple Rejection(6)	f = 120Hz, V_O = 11.5V to 21.5V		56.0	73.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(6)	f = 1kHz		–	17.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	230	–	mA
I_{PK}	Peak Current(6)	T_J = +25°C		–	2.2	–	A

Notes:

5. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
6. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7809) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 15V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		8.65	9.0	9.35	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 11.5V to 24V		8.6	9.0	9.4	
Regline	Line Regulation(7)	T_J = +25°C	V_I = 11.5V to 25V	–	6.0	180	mV
			V_I = 12V to 17V	–	2.0	90.0	
Regload	Load Regulation(7)	T_J = +25°C	I_O = 5mA to 1.5A	–	12.0	180	mV
			I_O = 250mA to 750mA	–	4.0	90.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 11.5V to 26V		–	–	1.3	
$\Delta V_O/\Delta T$	Output Voltage Drift(8)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	58.0	–	µV/V_O
RR	Ripple Rejection(8)	f = 120Hz, V_O = 13V to 23V		56.0	71.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(8)	f = 1kHz		–	17.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(8)	T_J = +25°C		–	2.2	–	A

Notes:

7. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
8. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7810) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 16V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		9.6	10.0	10.4	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 12.5V to 25V		9.5	10.0	10.5	
Regline	Line Regulation(9)	T_J = +25°C	V_I = 12.5V to 25V	–	10.0	200	mV
			V_I = 13V to 25V	–	3.0	100	
Regload	Load Regulation(9)	T_J = +25°C	I_O = 5mA to 1.5A	–	12.0	200	mV
			I_O = 250mA to 750mA	–	4.0	400	
I_Q	Quiescent Current	T_J = +25°C		–	5.1	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 12.5V to 29V		–	–	1.0	
$\Delta V_O/\Delta T$	Output Voltage Drift(10)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	58.0	–	µV/V_O
RR	Ripple Rejection(10)	f = 120Hz, V_O = 13V to 23V		56.0	71.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(10)	f = 1kHz		–	17.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(10)	T_J = +25°C		–	2.2	–	A

Notes:

9. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
10. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7812) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 19V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		11.5	12.0	12.5	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 14.5V to 27V		11.4	12.0	12.6	
Regline	Line Regulation(11)	T_J = +25°C	V_I = 14.5V to 30V	–	10.0	240	mV
			V_I = 16V to 22V	–	3.0	120	
Regload	Load Regulation(11)	T_J = +25°C	I_O = 5mA to 1.5A	–	11.0	240	mV
			I_O = 250mA to 750mA	–	5.0	120	
I_Q	Quiescent Current	T_J = +25°C		–	5.1	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	0.1	0.5	mA
		V_I = 14.5V to 30V		–	0.5	1.0	
$\Delta V_O/\Delta T$	Output Voltage Drift(12)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	76.0	–	µV/V_O
RR	Ripple Rejection(12)	f = 120Hz, V_I = 15V to 25V		55.0	71.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(12)	f = 1kHz		–	18.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	230	–	mA
I_{PK}	Peak Current(12)	T_J = +25°C		–	2.2	–	A

Notes:

11. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
12. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7815) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 23V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		14.4	15.0	15.6	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 17.5V to 30V		14.25	15.0	15.75	
Regline	Line Regulation(13)	T_J = +25°C	V_I = 17.5V to 30V	–	11.0	300	mV
			V_I = 20V to 26V	–	3.0	150	
Regload	Load Regulation(13)	T_J = +25°C	I_O = 5mA to 1.5A	–	12.0	300	mV
			I_O = 250mA to 750mA	–	4.0	150	
I_Q	Quiescent Current	T_J = +25°C		–	5.2	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 17.5V to 30V		–	–	1.0	
$\Delta V_O/\Delta T$	Output Voltage Drift(14)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	90.0	–	µV/V_O
RR	Ripple Rejection(14)	f = 120Hz, V_I = 18.5V to 28.5V		54.0	70.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(14)	f = 1kHz		–	19.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(14)	T_J = +25°C		–	2.2	–	A

Notes:

13. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
14. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7818) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 27V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		17.3	18.0	18.7	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 21V to 33V		17.1	18.0	18.9	
Regline	Line Regulation(15)	T_J = +25°C	V_I = 21V to 33V	–	15.0	360	mV
			V_I = 24V to 30V	–	5.0	180	
Regload	Load Regulation(15)	T_J = +25°C	I_O = 5mA to 1.5A	–	15.0	360	mV
			I_O = 250mA to 750mA	–	5.0	180	
I_Q	Quiescent Current	T_J = +25°C		–	5.2	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 21V to 33V		–	–	1.0	
$\Delta V_O/\Delta T$	Output Voltage Drift(16)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	110	–	µV/V_O
RR	Ripple Rejection(16)	f = 120Hz, V_I = 22V to 32V		53.0	69.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(16)	f = 1kHz		–	22.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(16)	T_J = +25°C		–	2.2	–	A

Notes:

15. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
16. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7824) (Continued)

Refer to the test circuits. -40°C < T_J < 125°C, I_O = 500mA, V_I = 33V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		23.0	24.0	25.0	V
		5mA ≤ I_O ≤ 1A, P_O ≤ 15W, V_I = 27V to 38V		22.8	24.0	25.25	
Regline	Line Regulation(17)	T_J = +25°C	V_I = 27V to 38V	–	17.0	480	mV
			V_I = 30V to 36V	–	6.0	240	
Regload	Load Regulation(17)	T_J = +25°C	I_O = 5mA to 1.5A	–	15.0	480	mV
			I_O = 250mA to 750mA	–	5.0	240	
I_Q	Quiescent Current	T_J = +25°C		–	5.2	8.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	0.1	0.5	mA
		V_I = 27V to 38V		–	0.5	1.0	
$\Delta V_O/\Delta T$	Output Voltage Drift(18)	I_O = 5mA		–	-1.5	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	60.0	–	µV/V_O
RR	Ripple Rejection(18)	f = 120Hz, V_I = 28V to 38V		50.0	67.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
rO	Output Resistance(18)	f = 1kHz		–	28.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	230	–	mA
I_{PK}	Peak Current(18)	T_J = +25°C		–	2.2	–	A

Notes:

17. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
18. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7805A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 10V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		4.9	5.0	5.1	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 7.5V to 20V		4.8	5.0	5.2	
Regline	Line Regulation[19]	V_I = 7.5V to 25V, I_O = 500mA		–	5.0	50.0	mV
		V_I = 8V to 12V		–	3.0	50.0	
		T_J = +25°C	V_I = 7.3V to 20V	–	5.0	50.0	
			V_I = 8V to 12V	–	1.5	25.0	
Regload	Load Regulation[19]	T_J = +25°C, I_O = 5mA to 1.5A		–	9.0	100	mV
		I_O = 5mA to 1A		–	9.0	100	
		I_O = 250mA to 750mA		–	4.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 8V to 25V, I_O = 500mA		–	–	0.8	
		V_I = 7.5V to 20V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift[20]	I_O = 5mA		–	-0.8	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection[20]	f = 120Hz, I_O = 500mA, V_I = 8V to 18V		–	68.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance[20]	f = 1kHz		–	17.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current[20]	T_J = +25°C		–	2.2	–	A

Notes:

19. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

20. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7806A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 11V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		5.58	6.0	6.12	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 8.6V to 21V		5.76	6.0	6.24	
Regline	Line Regulation(21)	V_I = 8.6V to 25V, I_O = 500mA		–	5.0	60.0	mV
		V_I = 9V to 13V		–	3.0	60.0	
		T_J = +25°C	V_I = 8.3V to 21V	–	5.0	60.0	
			V_I = 9V to 13V	–	1.5	30.0	
Regload	Load Regulation(21)	T_J = +25°C, I_O = 5mA to 1.5A		–	9.0	100	mV
		I_O = 5mA to 1A		–	9.0	100	
		I_O = 250mA to 750mA		–	5.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	4.3	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 19V to 25V, I_O = 500mA		–	–	0.8	
		V_I = 8.5V to 21V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift(22)	I_O = 5mA		–	-0.8	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(22)	f = 120Hz, I_O = 500mA, V_I = 9V to 19V		–	65.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(22)	f = 1kHz		–	17.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(22)	T_J = +25°C		–	2.2	–	A

Notes:

21. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

22. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7808A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 14V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Unit
V_O	Output Voltage	T_J = +25°C		7.84	8.0	8.16	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 10.6V to 23V		7.7	8.0	8.3	
Regline	Line Regulation(23)	V_I = 10.6V to 25V, I_O = 500mA		–	6.0	80.0	mV
		V_I = 11V to 17V		–	3.0	80.0	
		T_J = +25°C	V_I = 10.4V to 23V	–	6.0	80.0	
			V_I = 11V to 17V	–	2.0	40.0	
Regload	Load Regulation(23)	T_J = +25°C, I_O = 5mA to 1.5A		–	12.0	100	mV
		I_O = 5mA to 1A		–	12.0	100	
		I_O = 250mA to 750mA		–	5.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 11V to 25V, I_O = 500mA		–	–	0.8	
		V_I = 10.6V to 23V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift(24)	I_O = 5mA		–	-0.8	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(24)	f = 120Hz, I_O = 500mA, V_I = 11.5V to 21.5V		–	62.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(24)	f = 1kHz		–	18.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(24)	T_J = +25°C		–	2.2	–	A

Notes:

23. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

24. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7809A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 15V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
V_O	Output Voltage	T_J = +25°C		8.82	9.0	9.16	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 11.2V to 24V		8.65	9.0	9.35	
Regline	Line Regulation(25)	V_I = 11.7V to 25V, I_O = 500mA		–	6.0	90.0	mV
		V_I = 12.5V to 19V		–	4.0	45.0	
		T_J = +25°C	V_I = 11.5V to 24V	–	6.0	90.0	
			V_I = 12.5V to 19V	–	2.0	45.0	
Regload	Load Regulation(25)	T_J = +25°C, I_O = 5mA to 1.5A		–	12.0	100	mV
		I_O = 5mA to 1A		–	12.0	100	
		I_O = 250mA to 750mA		–	5.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 12V to 25V, I_O = 500mA		–	–	0.8	
		V_I = 11.7V to 25V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift(26)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(26)	f = 120Hz, I_O = 500mA, V_I = 12V to 22V		–	62.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(26)	f = 1kHz		–	17.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(26)	T_J = +25°C		–	2.2	–	A

Notes:

25. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

26. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7810A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 16V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
V_O	Output Voltage	T_J = +25°C		9.8	10.0	10.2	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 12.8V to 25V		9.6	10.0	10.4	
Regline	Line Regulation(27)	V_I = 12.8V to 26V, I_O = 500mA		–	8.0	100	mV
		V_I = 13V to 20V		–	4.0	50.0	
		T_J = +25°C	V_I = 12.5V to 25V	–	8.0	100	
			V_I = 13V to 20V	–	3.0	50.0	
Regload	Load Regulation(27)	T_J = +25°C, I_O = 5mA to 1.5A		–	12.0	100	mV
		I_O = 5mA to 1A		–	12.0	100	
		I_O = 250mA to 750mA		–	5.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.0	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 12.8V to 25V, I_O = 500mA		–	–	0.8	
		V_I = 13V to 26V, T_J = +25°C		–	–	0.5	
$\Delta V_O/\Delta T$	Output Voltage Drift(28)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(28)	f = 120Hz, I_O = 500mA, V_I = 14V to 24V		–	62.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(28)	f = 1kHz		–	17.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(28)	T_J = +25°C		–	2.2	–	A

Notes:

27. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

28. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7812A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 19V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
V_O	Output Voltage	T_J = +25°C		11.75	12.0	12.25	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 14.8V to 27V		11.5	12.0	12.5	
Regline	Line Regulation(29)	V_I = 14.8V to 30V, I_O = 500mA		–	10.0	120	mV
		V_I = 16V to 22V		–	4.0	120	
		T_J = +25°C	V_I = 14.5V to 27V	–	10.0	120	
			V_I = 16V to 22V	–	3.0	60.0	
Regload	Load Regulation(29)	T_J = +25°C, I_O = 5mA to 1.5A		–	12.0	100	mV
		I_O = 5mA to 1A		–	12.0	100	
		I_O = 250mA to 750mA		–	5.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.1	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 14V to 27V, I_O = 500mA		–	–	0.8	
		V_I = 15V to 30V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift(30)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(30)	f = 120Hz, I_O = 500mA, V_I = 14V to 24V		–	60.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(30)	f = 1kHz		–	18.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(30)	T_J = +25°C		–	2.2	–	A

Note:

29. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

30. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7815A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 23V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
V_O	Output Voltage	T_J = +25°C		14.75	15.0	15.3	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 17.7V to 30V		14.4	15.0	15.6	
Regline	Line Regulation(31)	V_I = 17.4V to 30V, I_O = 500mA		–	10.0	150	mV
		V_I = 20V to 26V		–	5.0	150	
		T_J = +25°C	V_I = 17.5V to 30V	–	11.0	150	
			V_I = 20V to 26V	–	3.0	75.0	
Regload	Load Regulation(31)	T_J = +25°C, I_O = 5mA to 1.5A		–	12.0	100	mV
		I_O = 5mA to 1A		–	12.0	100	
		I_O = 250mA to 750mA		–	5.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.2	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 17.5V to 30V, I_O = 500mA		–	–	0.8	
		V_I = 17.5V to 30V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift(32)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(32)	f = 120Hz, I_O = 500mA, V_I = 18.5V to 28.5V		–	58.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(32)	f = 1kHz		–	19.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(32)	T_J = +25°C		–	2.2	–	A

Notes:

31. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

32. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7818A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 27V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
V_O	Output Voltage	T_J = +25°C		17.64	18.0	18.36	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 21V to 33V		17.3	18.0	18.7	
Regline	Line Regulation(33)	V_I = 21V to 33V, I_O = 500mA		–	15.0	180	mV
		V_I = 21V to 33V		–	5.0	180	
		T_J = +25°C	V_I = 20.6V to 33V	–	15.0	180	
			V_I = 24V to 30V	–	5.0	90.0	
Regload	Load Regulation(33)	T_J = +25°C, I_O = 5mA to 1.5A		–	15.0	100	mV
		I_O = 5mA to 1A		–	15.0	100	
		I_O = 250mA to 750mA		–	7.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.2	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 12V to 33V, I_O = 500mA		–	–	0.8	
		V_I = 12V to 33V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift(34)	I_O = 5mA		–	-1.0	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(34)	f = 120Hz, I_O = 500mA, V_I = 22V to 32V		–	57.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(34)	f = 1kHz		–	19.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(34)	T_J = +25°C		–	2.2	–	A

Notes:

33. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

34. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (LM7824A) (Continued)

Refer to the test circuits. 0°C < T_J < 125°C, I_O = 1A, V_I = 33V, C_I = 0.33µF, C_O = 0.1µF, unless otherwise specified.

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
V_O	Output Voltage	T_J = +25°C		23.5	24.0	24.5	V
		I_O = 5mA to 1A, $P_O \leq$ 15W, V_I = 27.3V to 38V		23.0	24.0	25.0	
Regline	Line Regulation(35)	V_I = 27V to 38V, I_O = 500mA		–	18.0	240	mV
		V_I = 21V to 33V		–	6.0	240	
		T_J = +25°C	V_I = 26.7V to 38V	–	18.0	240	
			V_I = 30V to 36V	–	6.0	120	
Regload	Load Regulation(35)	T_J = +25°C, I_O = 5mA to 1.5A		–	15.0	100	mV
		I_O = 5mA to 1A		–	15.0	100	
		I_O = 250mA to 750mA		–	7.0	50.0	
I_Q	Quiescent Current	T_J = +25°C		–	5.2	6.0	mA
ΔI_Q	Quiescent Current Change	I_O = 5mA to 1A		–	–	0.5	mA
		V_I = 27.3V to 38V, I_O = 500mA		–	–	0.8	
		V_I = 27.3V to 38V, T_J = +25°C		–	–	0.8	
$\Delta V_O/\Delta T$	Output Voltage Drift(36)	I_O = 5mA		–	-1.5	–	mV/°C
V_N	Output Noise Voltage	f = 10Hz to 100kHz, T_A = +25°C		–	10.0	–	µV/V_O
RR	Ripple Rejection(36)	f = 120Hz, I_O = 500mA, V_I = 28V to 38V		–	54.0	–	dB
V_{DROP}	Dropout Voltage	I_O = 1A, T_J = +25°C		–	2.0	–	V
r_O	Output Resistance(36)	f = 1kHz		–	20.0	–	mΩ
I_{SC}	Short Circuit Current	V_I = 35V, T_A = +25°C		–	250	–	mA
I_{PK}	Peak Current(36)	T_J = +25°C		–	2.2	–	A

Notes:

35. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

36. These parameters, although guaranteed, are not 100% tested in production.

Typical Performance Characteristics

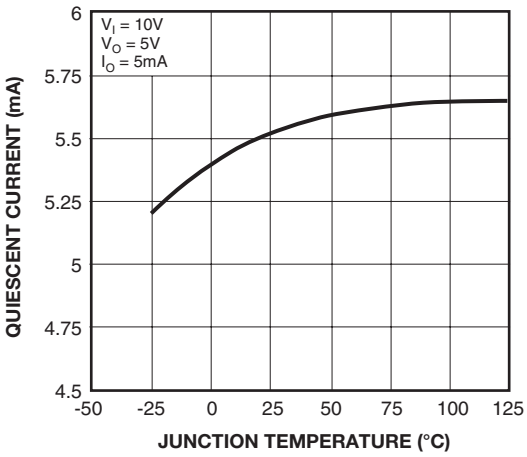


Figure 3. Quiescent Current

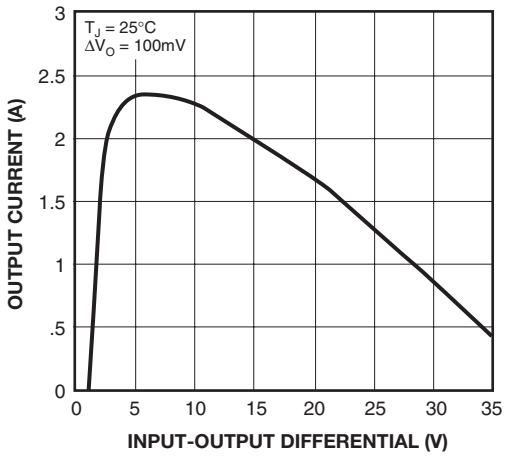


Figure 4. Peak Output Current

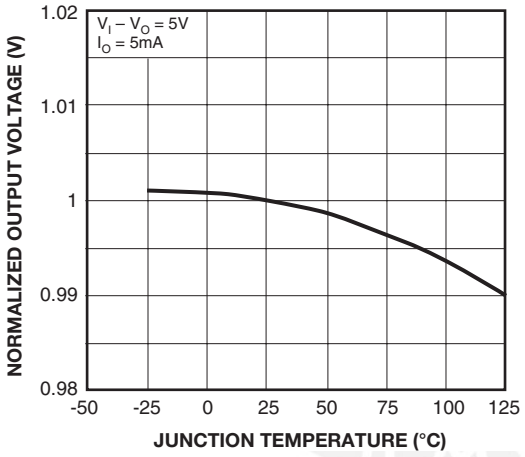


Figure 5. Output Voltage

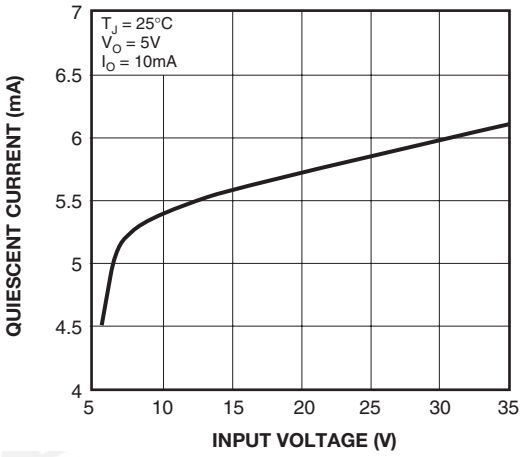


Figure 6. Quiescent Current

Typical Applications

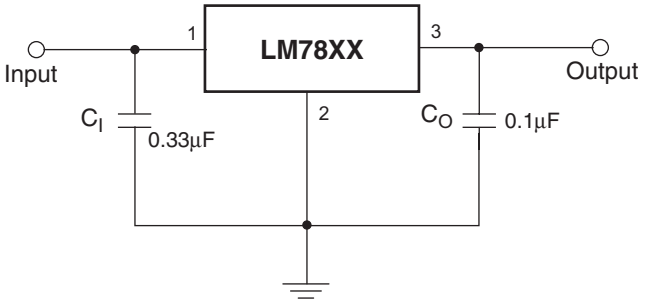


Figure 7. DC Parameters

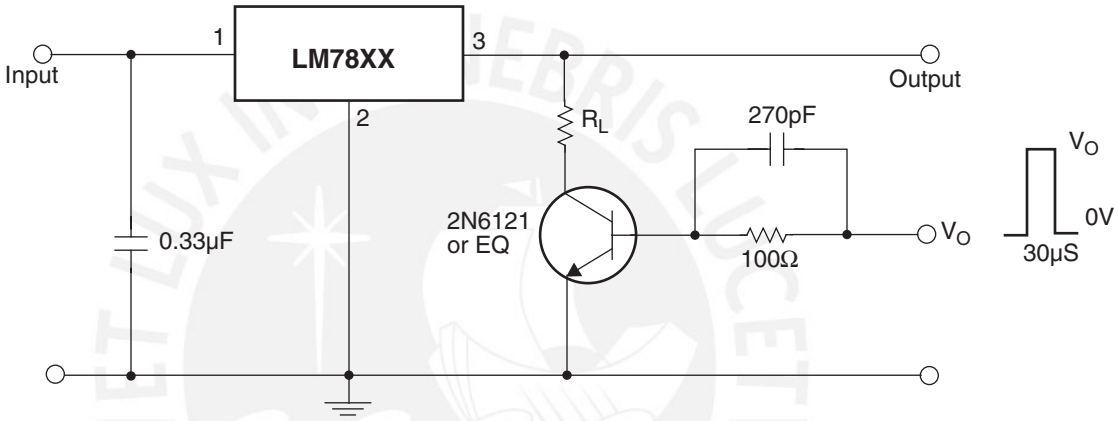


Figure 8. Load Regulation

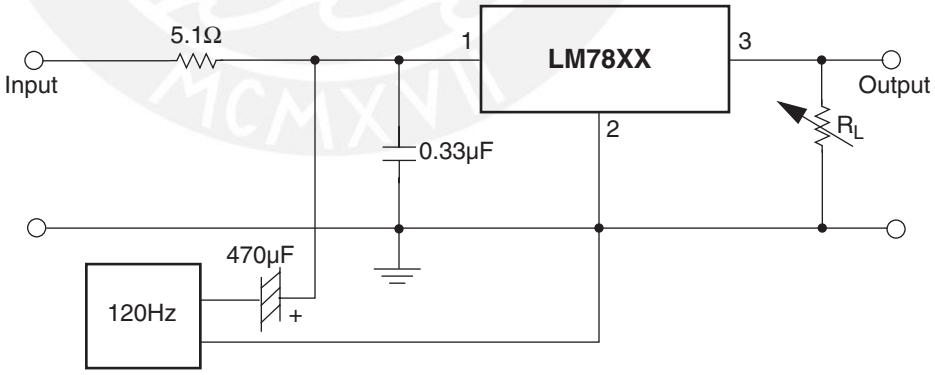


Figure 9. Ripple Rejection

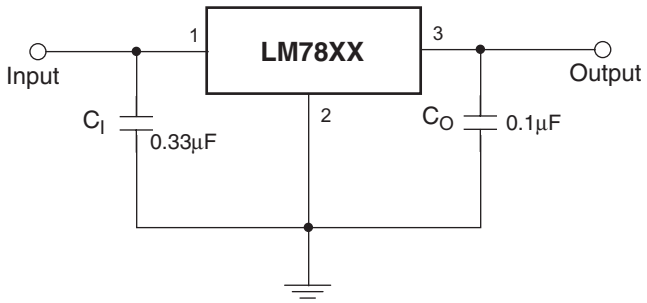
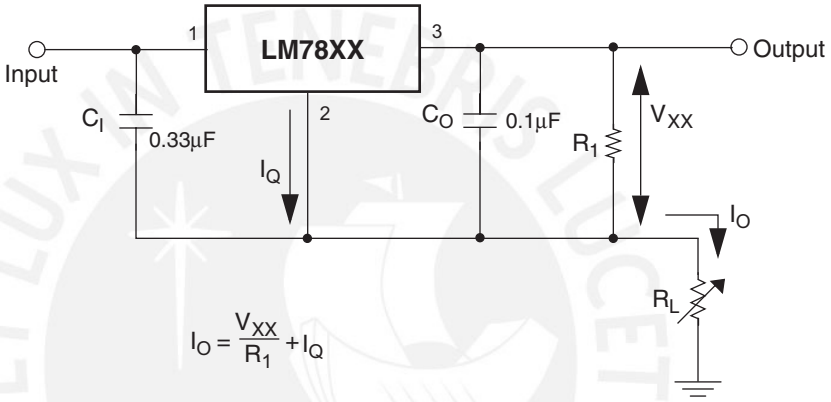


Figure 10. Fixed Output Regulator



Notes:

1. To specify an output voltage, substitute voltage value for "XX." A common ground is required between the input and the output voltage. The input voltage must remain typically 2.0V above the output voltage even during the low point on the input ripple voltage.
2. C_I is required if regulator is located an appreciable distance from power supply filter.
3. C_O improves stability and transient response.

Figure 11.

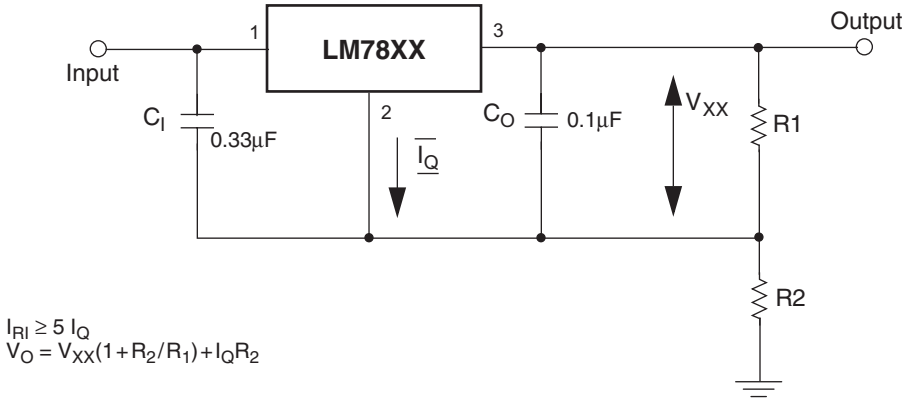


Figure 12. Circuit for Increasing Output Voltage

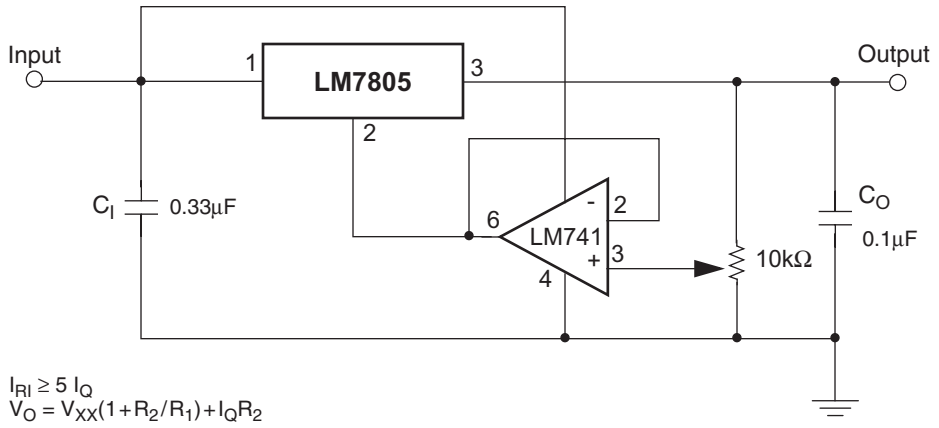


Figure 13. Adjustable Output Regulator (7V to 30V)

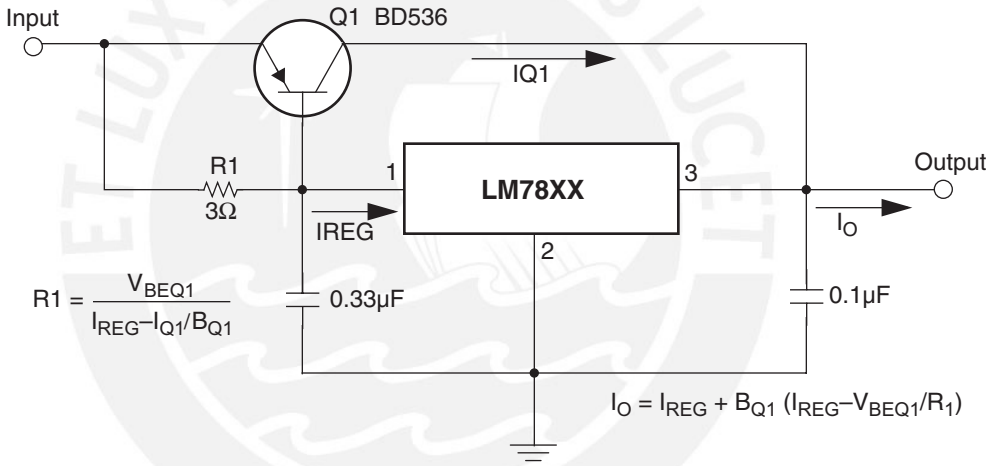


Figure 14. High Current Voltage Regulator

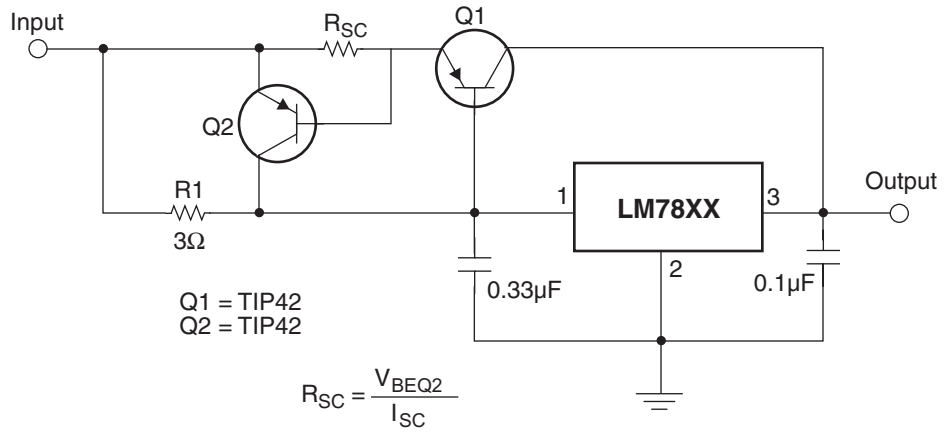


Figure 15. High Output Current with Short Circuit Protection

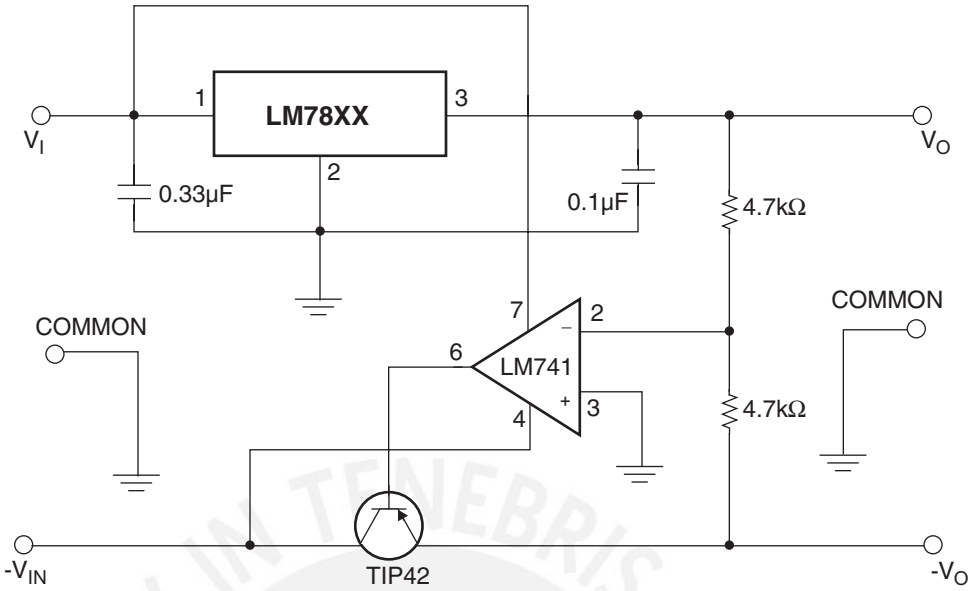


Figure 16. Tracking Voltage Regulator

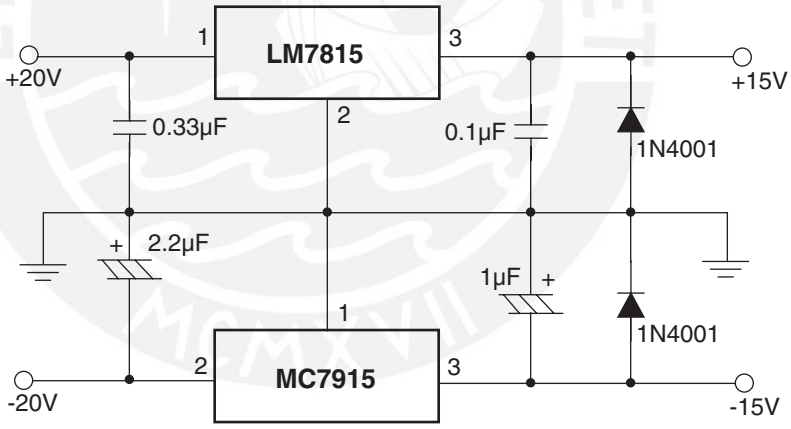


Figure 17. Split Power Supply (±15V – 1A)

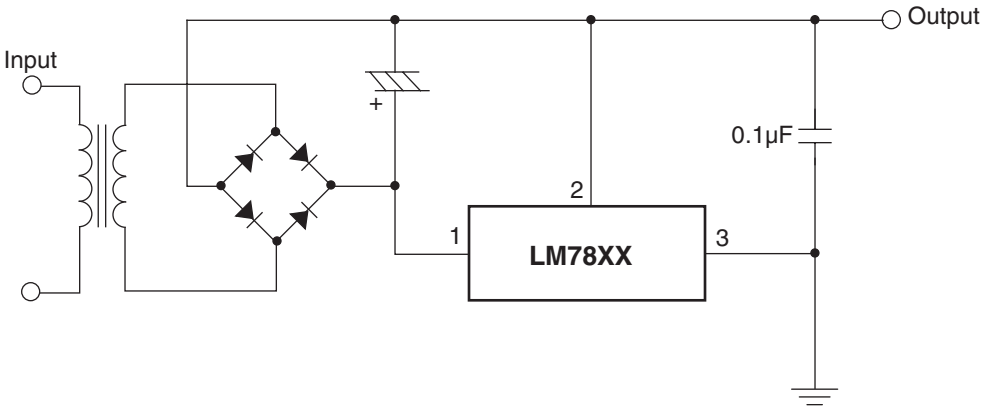


Figure 18. Negative Output Voltage Circuit

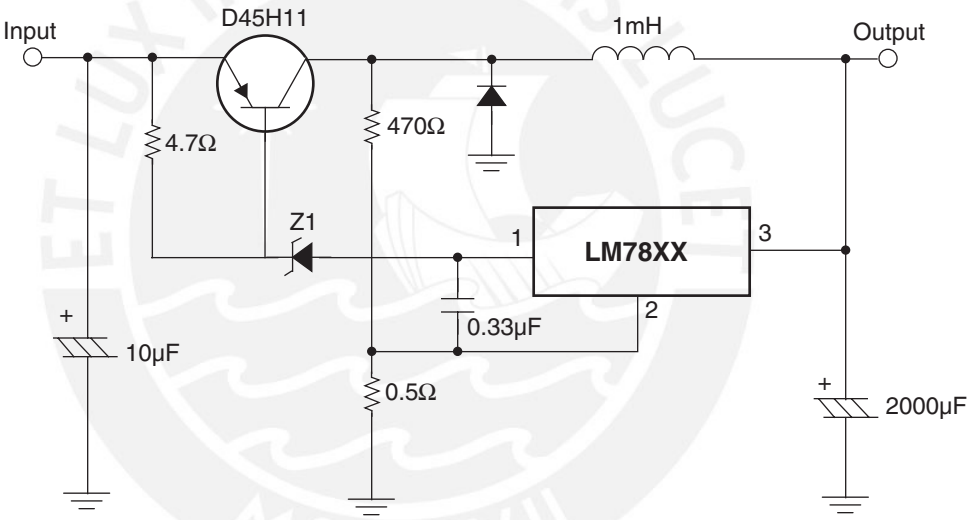
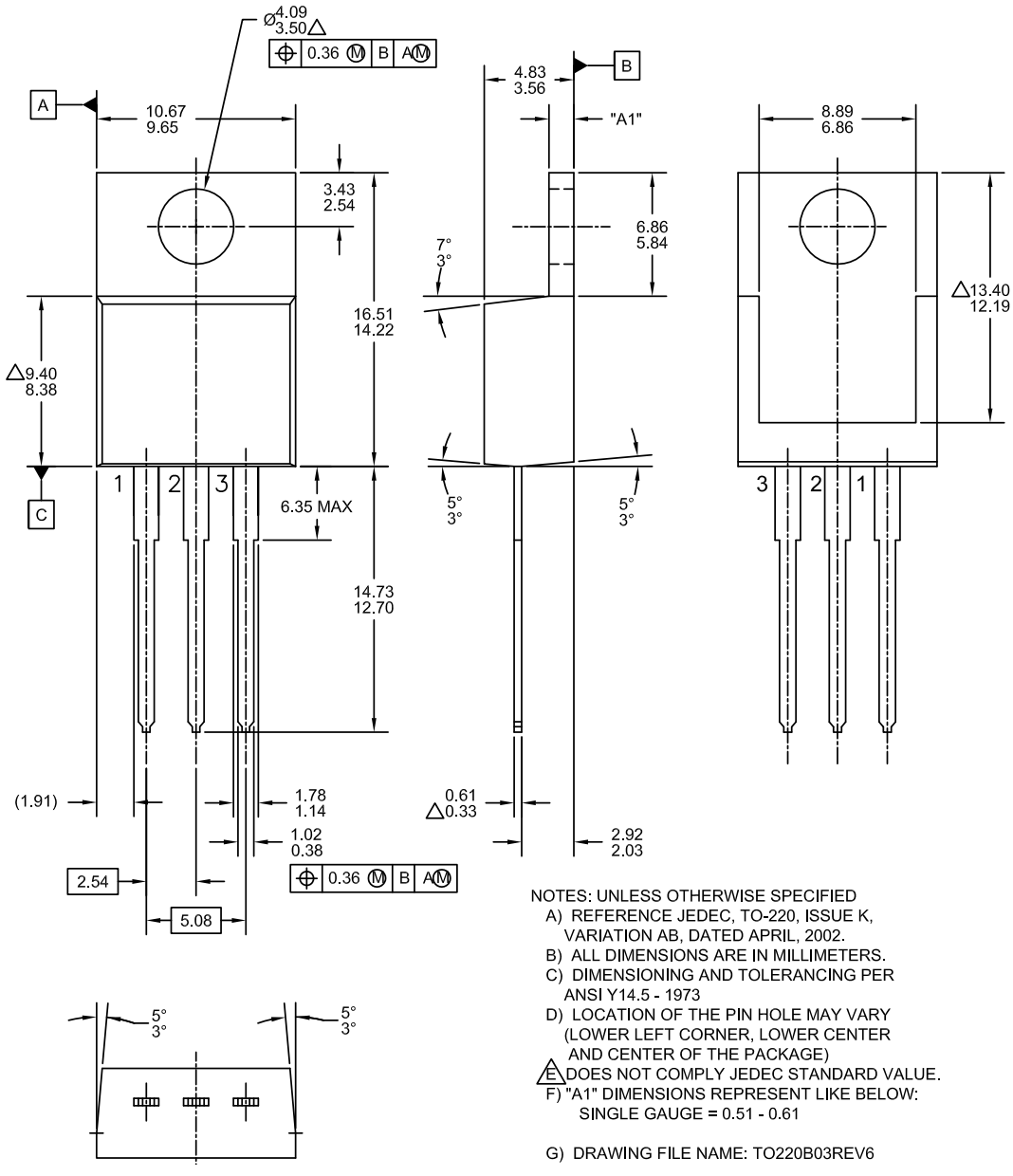


Figure 19. Switching Regulator

Mechanical Dimensions

Dimensions in millimeters

TO-220 [SINGLE GAUGE]



NOTES: UNLESS OTHERWISE SPECIFIED

A) REFERENCE JEDEC, TO-220, ISSUE K, VARIATION AB, DATED APRIL, 2002.

B) ALL DIMENSIONS ARE IN MILLIMETERS.

C) DIMENSIONING AND TOLERANCING PER ANSI Y14.5 - 1973

D) LOCATION OF THE PIN HOLE MAY VARY (LOWER LEFT CORNER, LOWER CENTER AND CENTER OF THE PACKAGE)

△E DOES NOT COMPLY JEDEC STANDARD VALUE.

F) "A1" DIMENSIONS REPRESENT LIKE BELOW:
SINGLE GAUGE = 0.51 - 0.61

G) DRAWING FILE NAME: TO220B03REV6

FAIRCHILD SEMICONDUCTOR®

TRADEMARKS

The following includes registered and unregistered trademarks and service marks, owned by Fairchild Semiconductor and/or its global subsidiaries, and is not intended to be an exhaustive list of all such trademarks.

2Cool™
AccuPower™
AX-CAP™*
BitSiC™
Build it Now™
CorePLUS™
CorePOWER™
CROSSVOLT™
CTL™
Current Transfer Logic™
DEUXPEED®
Dual Cool™
EcoSPARK®
EfficientMax™
ESBC™
F®
Fairchild®
Fairchild Semiconductor®
FACT Quiet Series™
FACT®
FAST®
FastvCore™
FETBench™
FlashWriter®*
FPS™
F-PFS™
FRFET®
Global Power Resource℠
GreenBridge™
Green FPS™
Green FPS™ e-Series™
G*max*™
GTO™
IntelliMAX™
ISOPLANAR™
Making Small Speakers Sound Louder and Better™
MegaBuck™
MICROCOUPLER™
MicroFET™
MicroPak™
MicroPak2™
MillerDrive™
MotionMax™
Motion-SPM™
mWSaver™
OptoHiT™
OPTOLOGIC®
OPTOPLANAR®
®
PowerTrench®
PowerXS™
Programmable Active Droop™
QFET®
QS™
Quiet Series™
RapidConfigure™
™
Saving our world, 1mW/W/kW at a time™
SignalWise™
SmartMax™
SMART START™
Solutions for Your Success™
SPM®
STEALTH™
SuperFET®
SuperSOT™-3
SuperSOT™-6
SuperSOT™-8
SupreMOS®
SyncFET™
Sync-Lock™
SYSTEM GENERAL®*
The Power Franchise®
the power franchise
TinyBoost™
TinyBuck™
TinyCalc™
TinyLogic®
TINYOPTO™
TinyPower™
TinyPWM™
TinyWire™
TranSiC™
TriFault Detect™
TRUECURRENT®*
µSerDes™
µSerDes™
UHC®
Ultra FRFET™
UniFET™
VCX™
VisualMax™
VoltagePlus™
XS™

* Trademarks of System General Corporation, used under license by Fairchild Semiconductor.

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION, OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS. THESE SPECIFICATIONS DO NOT EXPAND THE TERMS OF FAIRCHILD'S WORLDWIDE TERMS AND CONDITIONS, SPECIFICALLY THE WARRANTY THEREIN, WHICH COVERS THESE PRODUCTS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support, device, or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

ANTI-COUNTERFEITING POLICY

Fairchild Semiconductor Corporation's Anti-Counterfeiting Policy. Fairchild's Anti-Counterfeiting Policy is also stated on our external website, www.fairchildsemi.com, under Sales Support.

Counterfeiting of semiconductor parts is a growing problem in the industry. All manufacturers of semiconductor products are experiencing counterfeiting of their parts. Customers who inadvertently purchase counterfeit parts experience many problems such as loss of brand reputation, substandard performance, failed applications, and increased cost of production and manufacturing delays. Fairchild is taking strong measures to protect ourselves and our customers from the proliferation of counterfeit parts. Fairchild strongly encourages customers to purchase Fairchild parts either directly from Fairchild or from Authorized Fairchild Distributors who are listed by country on our web page cited above. Products customers buy either from Fairchild directly or from Authorized Fairchild Distributors are genuine parts, have full traceability, meet Fairchild's quality standards for handling and storage and provide access to Fairchild's full range of up-to-date technical and product information. Fairchild and our Authorized Distributors will stand behind all warranties and will appropriately address any warranty issues that may arise. Fairchild will not provide any warranty coverage or other assistance for parts bought from Unauthorized Sources. Fairchild is committed to combat this global problem and encourage our customers to do their part in stopping this practice by buying direct or from authorized distributors.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative / In Design	Datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	Datasheet contains preliminary data; supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve design.
No Identification Needed	Full Production	Datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve the design.
Obsolete	Not In Production	Datasheet contains specifications on a product that is discontinued by Fairchild Semiconductor. The datasheet is for reference information only.

Rev. I61

LM117/LM317A/LM317-N 3-Terminal Adjustable Regulator

Check for Samples: LM117, LM317-N

FEATURES

- **Specified 1% Output Voltage Tolerance (LM317A)**
- **Specified Max. 0.01%/V Line Regulation (LM317A)**
- **Specified Max. 0.3% Load Regulation (LM117)**
- **Specified 1.5A Output Current**
- **Adjustable Output Down to 1.2V**
- **Current Limit Constant With Temperature**
- **P^+ Product Enhancement Tested**
- **80 dB Ripple Rejection**
- **Output is Short-Circuit Protected**

DESCRIPTION

The LM117 series of adjustable 3-terminal positive voltage regulators is capable of supplying in excess of 1.5A over a 1.2V to 37V output range. They are exceptionally easy to use and require only two external resistors to set the output voltage. Further, both line and load regulation are better than standard fixed regulators. Also, the LM117 is packaged in standard transistor packages which are easily mounted and handled.

In addition to higher performance than fixed regulators, the LM117 series offers full overload protection available only in IC's. Included on the chip are current limit, thermal overload protection and safe area protection. All overload protection circuitry remains fully functional even if the adjustment terminal is disconnected.

Normally, no capacitors are needed unless the device is situated more than 6 inches from the input filter capacitors in which case an input bypass is needed. An optional output capacitor can be added to improve transient response. The adjustment terminal can be bypassed to achieve very high ripple rejection ratios which are difficult to achieve with standard 3-terminal regulators.

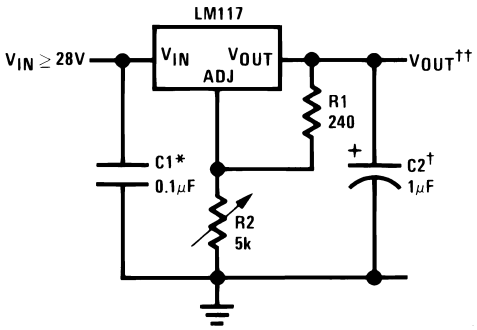
Besides replacing fixed regulators, the LM117 is useful in a wide variety of other applications. Since the regulator is “floating” and sees only the input-to-output differential voltage, supplies of several hundred volts can be regulated as long as the maximum input to output differential is not exceeded, i.e., avoid short-circuiting the output.

Also, it makes an especially simple adjustable switching regulator, a programmable output regulator, or by connecting a fixed resistor between the adjustment pin and output, the LM117 can be used as a precision current regulator. Supplies with electronic shutdown can be achieved by clamping the adjustment terminal to ground which programs the output to 1.2V where most loads draw little current.

For applications requiring greater output current, see LM150 series (3A) and LM138 series (5A) data sheets. For the negative complement, see LM137 series data sheet.

Typical Applications

Figure 1. 1.2V–25V Adjustable Regulator



Full output current not available at high input-output voltages

*Needed if device is more than 6 inches from filter capacitors.

†Optional—improves transient response. Output capacitors in the range of 1μF to 1000μF of aluminum or tantalum electrolytic are commonly used to provide improved output impedance and rejection of transients.

$$\dagger\dagger V_{OUT} = 1.25V\left(1 + \frac{R2}{R1}\right) + I_{ADJ}(R_2)$$

LM117/LM317A/LM317-N Package Options

Part Number	Suffix	Package	Output Current
LM117, LM317-N	NDS	TO-3	1.5A
LM317A, LM317-N	NDE	TO-220	1.5A
LM317-N	KTT	TO-263	1.5A
LM317A, LM317-N	DCY	SOT-223	1.0A
LM117, LM317A, LM317-N	NDT	TO	0.5A
LM117	NAJ	LCCC	0.5A
LM317A, LM317-N	NDP	PFM	0.5A

SOT-223 vs. PFM Packages

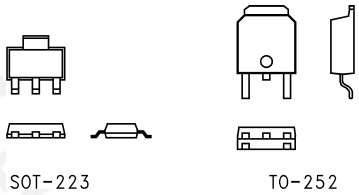
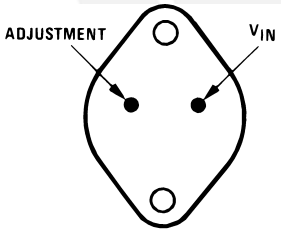


Figure 2. Scale 1:1

Connection Diagrams

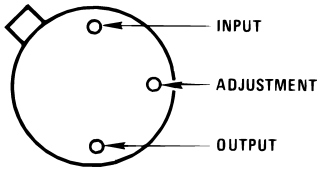
TO-3 (NDS)
Metal Can Package



CASE IS OUTPUT

Figure 3. Bottom View
Bottom View
Package Number NDS or K

TO (NDT)
Metal Can Package



CASE IS OUTPUT

Figure 4. Bottom View
Package Number NDT

Figure 5. TO-263 (KTT)
Surface-Mount Package

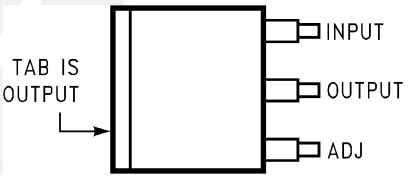


Figure 6. Top View

TO-263 (KTT)
Surface-Mount Package

Figure 7. Side View
Package Number KTT

TO-220 (NDE)
Plastic Package

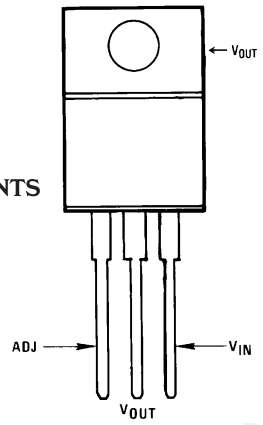


Figure 8. Front View
Package Number NDE

Ceramic Leadless Chip Carrier (NAJ)

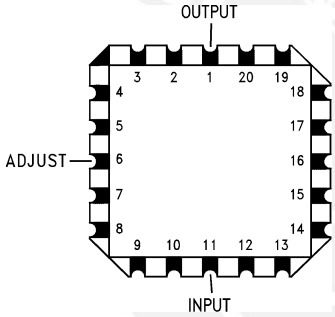


Figure 9. Top View
Package Number NAJ

4-Lead SOT-223 (DCY)

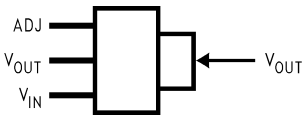


Figure 10. Front View
Package Number DCY

PFM (NDP)

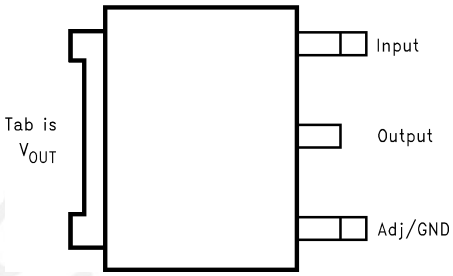


Figure 11. Front View
Package Number NDP

These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

Absolute Maximum Ratings (1) (2)

Power Dissipation		Internally Limited
Input-Output Voltage Differential		+40V, −0.3V
Storage Temperature		−65°C to +150°C
Lead Temperature	Metal Package (Soldering, 10 seconds)	300°C
	Plastic Package (Soldering, 4 seconds)	260°C
ESD Tolerance (3)		3 kV

(1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not ensure specific performance limits. For ensured specifications and test conditions, see the Electrical Characteristics. The ensured specifications apply only for the test conditions listed.
(2) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/Distributors for availability and specifications.
(3) Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Operating Temperature Range

LM117	$-55°C \le T_J \le +150°C$
LM317A	$-40°C \le T_J \le +125°C$
LM317-N	$0°C \le T_J \le +125°C$
Preconditioning	
Thermal Limit Burn-In	All Devices 100%

LM117 Electrical Characteristics (1)

Specifications with standard type face are for T_J = 25°C, and those with **boldface type** apply over **full Operating Temperature Range**. Unless otherwise specified, $V_{IN} - V_{OUT}$ = 5V, and I_{OUT} = 10 mA.

Parameter	Conditions		LM117 (2)			
			Min	Typ	Max	Units
Reference Voltage	$3V \le (V_{IN} - V_{OUT}) \le 40V$, $10\ mA \le I_{OUT} \le I_{MAX}$ (1)		**1.20**	**1.25**	**1.30**	V
Line Regulation	$3V \le (V_{IN} - V_{OUT}) \le 40V$ (3)			0.01 **0.02**	0.02 **0.05**	%/V
Load Regulation	$10\ mA \le I_{OUT} \le I_{MAX}$ (1) (3)			0.1 **0.3**	0.3 **1**	%
Thermal Regulation	20 ms Pulse			0.03	0.07	%/W
Adjustment Pin Current				**50**	**100**	µA
Adjustment Pin Current Change	$10\ mA \le I_{OUT} \le I_{MAX}$ (1) $3V \le (V_{IN} - V_{OUT}) \le 40V$			**0.2**	**5**	µA
Temperature Stability	$T_{MIN} \le T_J \le T_{MAX}$			**1**		%
Minimum Load Current	$(V_{IN} - V_{OUT}) = 40V$			**3.5**	**5**	mA
Current Limit	$(V_{IN} - V_{OUT}) \le 15V$	NDS Package	**1.5**	**2.2**	**3.4**	A
		NDT, NAJ Package	**0.5**	**0.8**	**1.8**	
	$(V_{IN} - V_{OUT}) = 40V$	NDS Package	0.3	0.4		A
		NDT, NAJ Package	0.15	0.20		
RMS Output Noise, % of V_{OUT}	10 Hz ≤ f ≤ 10 kHz			0.003		%

(1) I_{MAX} = 1.5A for the NDS (TO-3), NDE (TO-220), and KTT (TO-263) packages. I_{MAX} = 1.0A for the DCY (SOT-223) package. I_{MAX} = 0.5A for the NDT (TO), MDT (PFM), and NAJ (LCCC) packages. Device power dissipation (P_D) is limited by ambient temperature (T_A), device maximum junction temperature (T_J), and package thermal resistance (θ_{JA}). The maximum allowable power dissipation at any temperature is : $P_{D(MAX)} = ((T_{J(MAX)} - T_A)/\theta_{JA})$. All Min. and Max. limits are ensured to TI's Average Outgoing Quality Level (AOQL).
(2) Refer to RETS117H drawing for the LM117H, or the RETS117K for the LM117K military specifications.
(3) Regulation is measured at a constant junction temperature, using pulse testing with a low duty cycle. Changes in output voltage due to heating effects are covered under the specifications for thermal regulation.

LM117 Electrical Characteristics(1) (continued)

Specifications with standard type face are for T_J = 25°C, and those with **boldface type** apply over **full Operating Temperature Range**. Unless otherwise specified, $V_{IN} - V_{OUT}$ = 5V, and I_{OUT} = 10 mA.

Parameter	Conditions	LM117 (2)			
		Min	Typ	Max	Units
Ripple Rejection Ratio	V_{OUT} = 10V, f = 120 Hz, C_{ADJ} = 0 µF		**65**		dB
	V_{OUT} = 10V, f = 120 Hz, C_{ADJ} = 10 µF	**66**	**80**		dB
Long-Term Stability	T_J = 125°C, 1000 hrs		0.3	1	%
Thermal Resistance, θ_{JC} Junction-to-Case	NDS (TO-3) Package		2		°C/W
	NDT (TO) Package		21		
	NAJ (LCCC) Package		12		
Thermal Resistance, θ_{JA} Junction-to-Ambient (No Heat Sink)	NDS (TO-3) Package		39		°C/W
	NDT (TO) Package		186		
	NAJ (LCCC) Package		88		

LM317A and LM317-N Electrical Characteristics(1)

Specifications with standard type face are for T_J = 25°C, and those with **boldface type** apply over **full Operating Temperature Range**. Unless otherwise specified, $V_{IN} - V_{OUT}$ = 5V, and I_{OUT} = 10 mA.

Parameter	Conditions		LM317A			LM317-N			
			Min	Typ	Max	Min	Typ	Max	Units
Reference Voltage			1.238	1.250	1.262	-	1.25	-	V
	3V ≤ ($V_{IN} - V_{OUT}$) ≤ 40V, 10 mA ≤ I_{OUT} ≤ I_{MAX} (1)		**1.225**	**1.250**	**1.270**	**1.20**	**1.25**	**1.30**	V
Line Regulation	3V ≤ ($V_{IN} - V_{OUT}$) ≤ 40V (2)			0.005 **0.01**	0.01 **0.02**		0.01 **0.02**	0.04 **0.07**	%/V
Load Regulation	10 mA ≤ I_{OUT} ≤ I_{MAX} (1) (2)			0.1 **0.3**	0.5 **1**		0.1 **0.3**	0.5 **1.5**	%
Thermal Regulation	20 ms Pulse			0.04	0.07		0.04	0.07	%/W
Adjustment Pin Current				**50**	**100**		**50**	**100**	µA
Adjustment Pin Current Change	10 mA ≤ I_{OUT} ≤ I_{MAX} (1) 3V ≤ ($V_{IN} - V_{OUT}$) ≤ 40V			**0.2**	**5**		**0.2**	**5**	µA
Temperature Stability	T_{MIN} ≤ T_J ≤ T_{MAX}			**1**			**1**		%
Minimum Load Current	($V_{IN} - V_{OUT}$) = 40V			**3.5**	**10**		**3.5**	**10**	mA
Current Limit	($V_{IN} - V_{OUT}$) ≤ 15V	NDS, KTT Packages	-	-	-	**1.5**	**2.2**	**3.4**	A
		DCY, NDE Packages	**1.5**	**2.2**	**3.4**	**1.5**	**2.2**	**3.4**	
		NDT, MDT Packages	**0.5**	**0.8**	**1.8**	**0.5**	**0.8**	**1.8**	
	($V_{IN} - V_{OUT}$) = 40V	NDS, KTT Packages	-	-		0.15	0.40		A
		DCY, NDE Packages	0.112	0.30		0.112	0.30		
		NDT, MDT Packages	0.075	0.20		0.075	0.20		
RMS Output Noise, % of V_{OUT}	10 Hz ≤ f ≤ 10 kHz			0.003			0.003		%
Ripple Rejection Ratio	V_{OUT} = 10V, f = 120 Hz, C_{ADJ} = 0 µF			**65**			**65**		dB
	V_{OUT} = 10V, f = 120 Hz, C_{ADJ} = 10 µF		**66**	**80**		**66**	**80**		dB
Long-Term Stability	T_J = 125°C, 1000 hrs			0.3	1		0.3	1	%

(1) I_{MAX} = 1.5A for the NDS (TO-3), NDE (TO-220), and KTT (TO-263) packages. I_{MAX} = 1.0A for the DCY (SOT-223) package. I_{MAX} = 0.5A for the NDT (TO), MDT (PFM), and NAJ (LCCC) packages. Device power dissipation (P_D) is limited by ambient temperature (T_A), device maximum junction temperature (T_J), and package thermal resistance (θ_{JA}). The maximum allowable power dissipation at any temperature is : $P_{D(MAX)} = ((T_{J(MAX)} - T_A)/\theta_{JA})$. All Min. and Max. limits are ensured to TI's Average Outgoing Quality Level (AOQL).

(2) Regulation is measured at a constant junction temperature, using pulse testing with a low duty cycle. Changes in output voltage due to heating effects are covered under the specifications for thermal regulation.

LM317A and LM317-N Electrical Characteristics[1] (continued)

Specifications with standard type face are for T_J = 25°C, and those with **boldface type** apply over **full Operating Temperature Range**. Unless otherwise specified, $V_{IN} - V_{OUT}$ = 5V, and I_{OUT} = 10 mA.

Parameter	Conditions	LM317A			LM317-N			Units
		Min	Typ	Max	Min	Typ	Max	
Thermal Resistance, θ_{JC} Junction-to-Case	NDS (TO-3) Package		-			2		°C/W
	NDE (TO-220) Package		4			4		
	KTT (TO-263) Package		-			4		
	DCY (SOT-223) Package		23.5			23.5		
	NDT (TO) Package		21			21		
	MDT (PFM) Package		12			12		
Thermal Resistance, θ_{JA} Junction-to-Ambient (No Heat Sink)	NDS (TO-3) Package		-			39		°C/W
	NDE (TO-220) Package		50			50		
	KTT (TO-263) Package [3]		-			50		
	DCY (SOT-223) Package [3]		140			140		
	NDT (TO) Package		186			186		
	MDT (PFM) Package [3]		103			103		

(3) When surface mount packages are used (TO-263, SOT-223, PFM), the junction to ambient thermal resistance can be reduced by increasing the PC board copper area that is thermally connected to the package. See the Applications Hints section for heatsink techniques.

Typical Performance Characteristics

Output Capacitor = 0 µF unless otherwise noted

Load Regulation

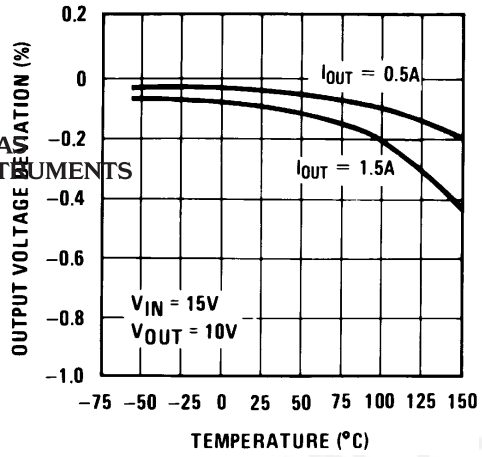


Figure 12.

Current Limit

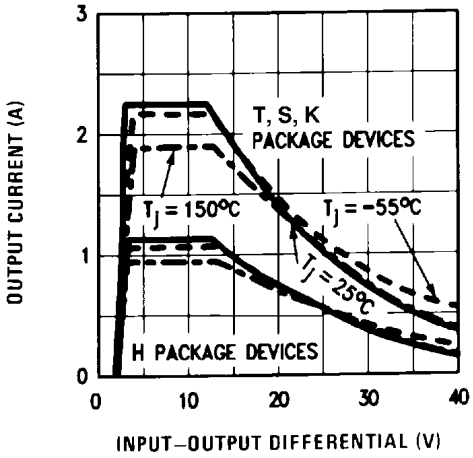


Figure 13.

Adjustment Current

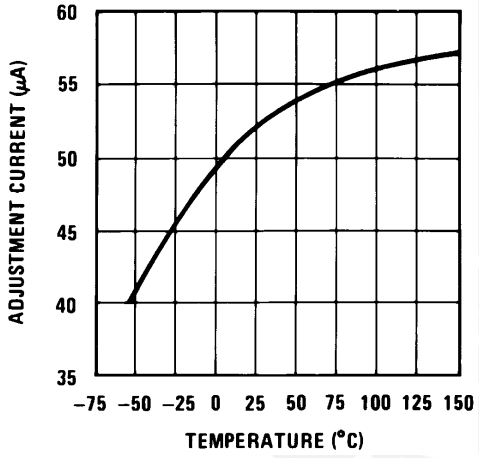


Figure 14.

Dropout Voltage

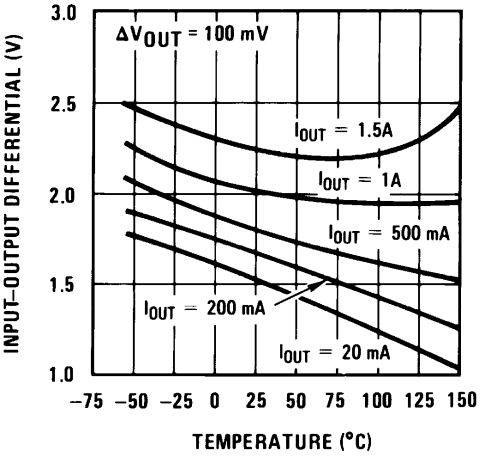


Figure 15.

V_{OUT} vs V_{IN}, V_{OUT} = V_{REF}

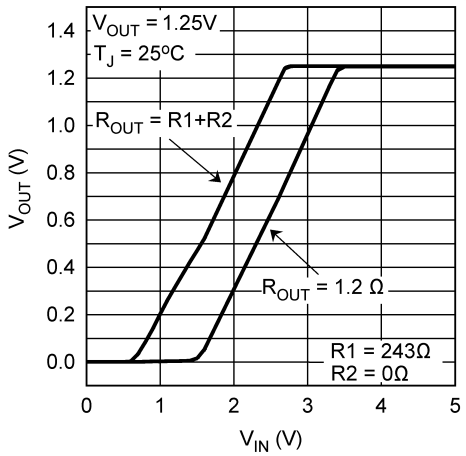


Figure 16.

V_{OUT} vs V_{IN}, V_{OUT} = 5V

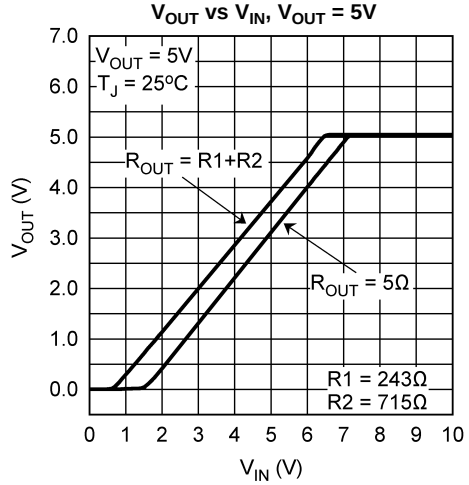


Figure 17.

Typical Performance Characteristics (continued)

Output Capacitor = 0 µF unless otherwise noted

Temperature Stability

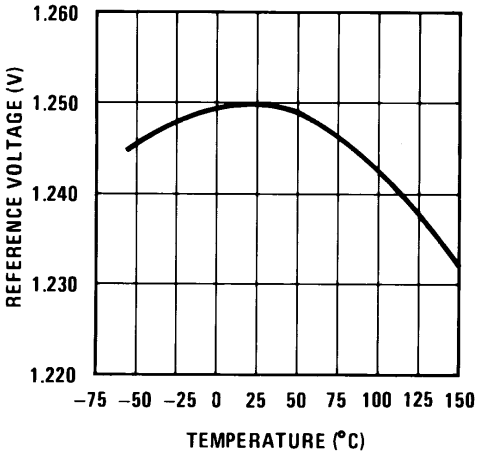


Figure 18.

Minimum Operating Current

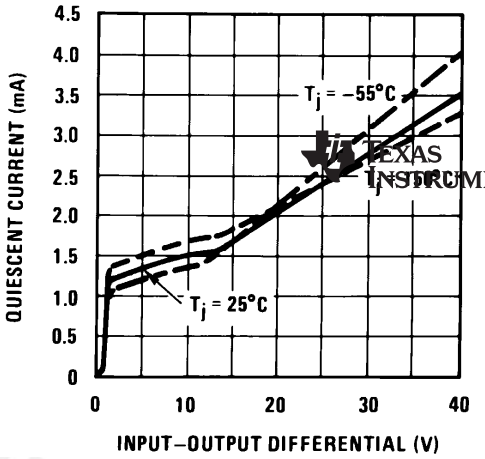


Figure 19.

Ripple Rejection

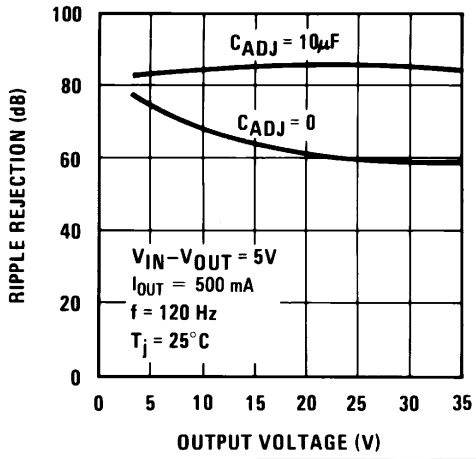


Figure 20.

Ripple Rejection

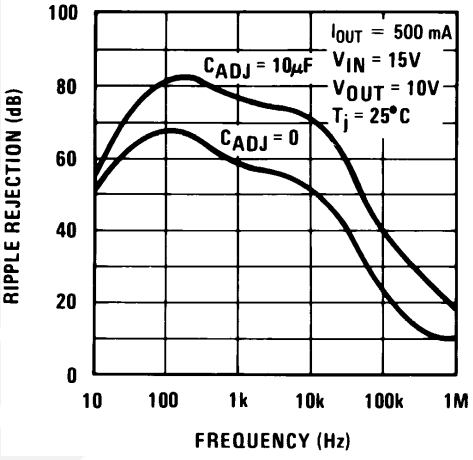


Figure 21.

Ripple Rejection

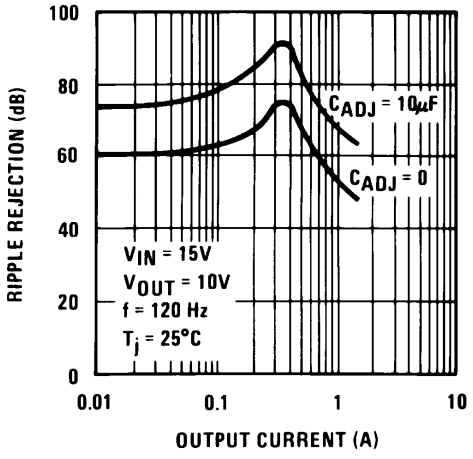


Figure 22.

Output Impedance

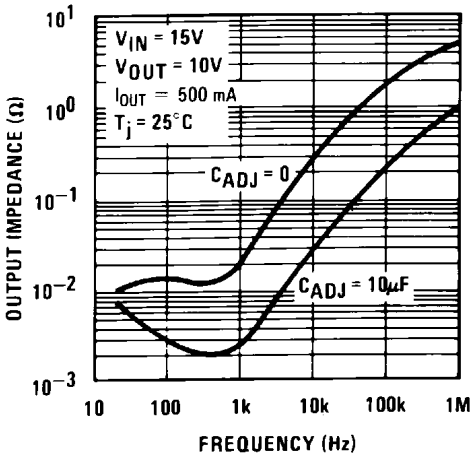


Figure 23.

Typical Performance Characteristics (continued)

Output Capacitor = 0 µF unless otherwise noted

Line Transient Response

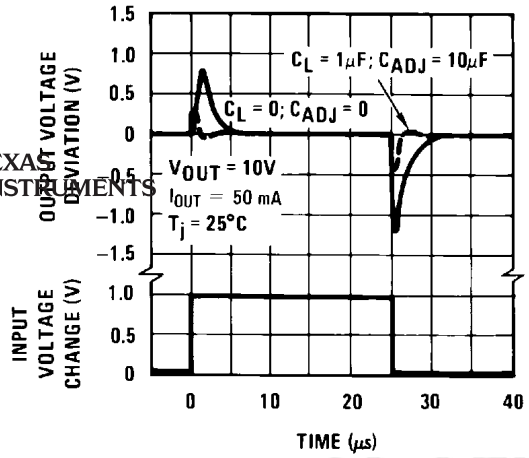


Figure 24.

Load Transient Response

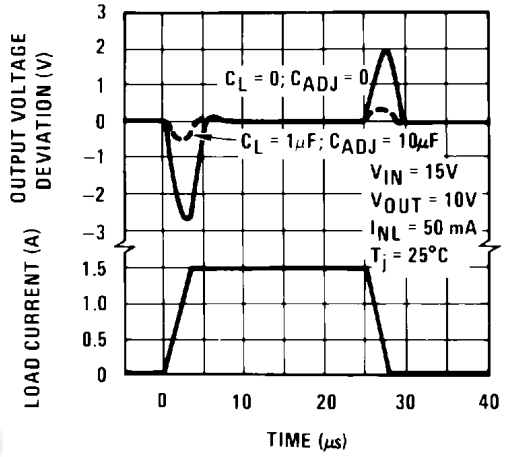
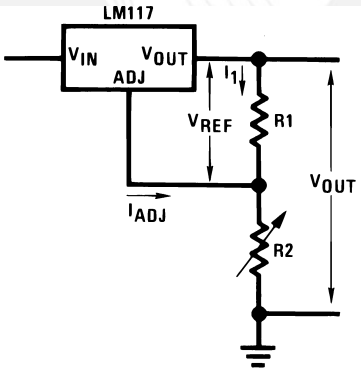


Figure 25.

APPLICATION HINTS

In operation, the LM117 develops a nominal 1.25V reference voltage, V_{REF}, between the output and adjustment terminal. The reference voltage is impressed across program resistor R1 and, since the voltage is constant, a constant current I_1 then flows through the output set resistor R2, giving an output voltage of

$$V_{OUT} = V_{REF}\left(1 + \frac{R2}{R1}\right) + I_{ADJ}R2 \quad (1)$$



Since the 100µA current from the adjustment terminal represents an error term, the LM117 was designed to minimize I_{ADJ} and make it very constant with line and load changes. To do this, all quiescent operating current is returned to the output establishing a minimum load current requirement. If there is insufficient load on the output, the output will rise.

External Capacitors

An input bypass capacitor is recommended. A 0.1µF disc or 1µF solid tantalum on the input is suitable input bypassing for almost all applications. The device is more sensitive to the absence of input bypassing when adjustment or output capacitors are used but the above values will eliminate the possibility of problems.

The adjustment terminal can be bypassed to ground on the LM117 to improve ripple rejection. This bypass capacitor prevents ripple from being amplified as the output voltage is increased. With a 10 µF bypass capacitor 80dB ripple rejection is obtainable at any output level. Increases over 10 µF do not appreciably improve the ripple rejection at frequencies above 120Hz. If the bypass capacitor is used, it is sometimes necessary to include protection diodes to prevent the capacitor from discharging through internal low current paths and damaging the device.

In general, the best type of capacitors to use is solid tantalum. Solid tantalum capacitors have low impedance even at high frequencies. Depending upon capacitor construction, it takes about 25 µF in aluminum electrolytic to equal 1µF solid tantalum at high frequencies. Ceramic capacitors are also good at high frequencies; but some types have a large decrease in capacitance at frequencies around 0.5 MHz. For this reason, 0.01 µF disc may seem to work better than a 0.1 µF disc as a bypass.

Although the LM117 is stable with no output capacitors, like any feedback circuit, certain values of external capacitance can cause excessive ringing. This occurs with values between 500 pF and 5000 pF. A 1 µF solid tantalum (or 25 µF aluminum electrolytic) on the output swamps this effect and insures stability. Any increase of the load capacitance larger than 10 µF will merely improve the loop stability and output impedance.

Load Regulation

The LM117 is capable of providing extremely good load regulation but a few precautions are needed to obtain maximum performance. The current set resistor connected between the adjustment terminal and the output terminal (usually 240Ω) should be tied directly to the output (case) of the regulator rather than near the load. This eliminates line drops from appearing effectively in series with the reference and degrading regulation. For example, a 15V regulator with 0.05Ω resistance between the regulator and load will have a load regulation due to line resistance of 0.05Ω × I_L. If the set resistor is connected near the load the effective line resistance will be 0.05Ω (1 + R2/R1) or in this case, 11.5 times worse.

Figure 26 shows the effect of resistance between the regulator and 240Ω set resistor.

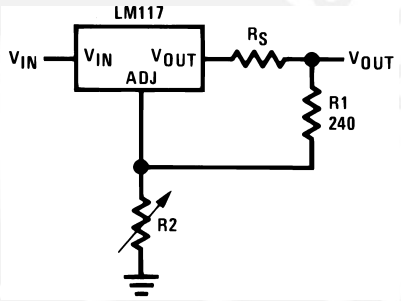


Figure 26. Regulator with Line Resistance in Output Lead

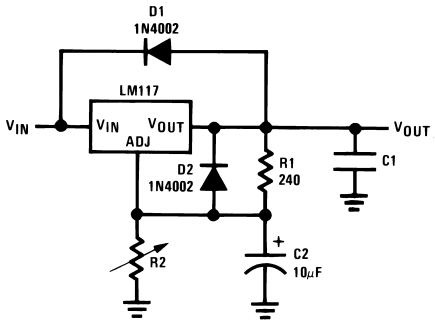
With the TO-3 package, it is easy to minimize the resistance from the case to the set resistor, by using two separate leads to the case. However, with the TO package, care should be taken to minimize the wire length of the output lead. The ground of R2 can be returned near the ground of the load to provide remote ground sensing and improve load regulation.

Protection Diodes

When external capacitors are used with any IC regulator it is sometimes necessary to add protection diodes to prevent the capacitors from discharging through low current points into the regulator. Most 10 µF capacitors have low enough internal series resistance to deliver 20A spikes when shorted. Although the surge is short, there is enough energy to damage parts of the IC.

When an output capacitor is connected to a regulator and the input is shorted, the output capacitor will discharge into the output of the regulator. The discharge current depends on the value of the capacitor, the output voltage of the regulator, and the rate of decrease of V_{IN}. In the LM117, this discharge path is through a large junction that is able to sustain 15A surge with no problem. This is not true of other types of positive regulators. For output capacitors of 25 µF or less, there is no need to use diodes.

The bypass capacitor on the adjustment terminal can discharge through a low current junction. Discharge occurs when *either* the input, or the output, is shorted. Internal to the LM117 is a 50Ω resistor which limits the peak discharge current. No protection is needed for output voltages of 25V or less and 10 µF capacitance. *Figure 27* shows an LM117 with protection diodes included for use with outputs greater than 25V and high values of output capacitance.



$$V_{OUT} = 1.25V\left(1 + \frac{R2}{R1}\right) + I_{ADJ}R2$$

D1 protects against C1
D2 protects against C2

Figure 27. Regulator with Protection Diodes

Heatsink Requirements

The LM317-N regulators have internal thermal shutdown to protect the device from over-heating. Under all operating conditions, the junction temperature of the LM317-N should not exceed the rated maximum junction temperature (T_J) of 150°C for the LM117, or 125°C for the LM317A and LM317-N. A heatsink may be required depending on the maximum device power dissipation and the maximum ambient temperature of the application. To determine if a heatsink is needed, the power dissipated by the regulator, P_D, must be calculated:

$$P_D = ((V_{IN} - V_{OUT}) \times I_L) + (V_{IN} \times I_G) \quad (2)$$

Figure 28 shows the voltage and currents which are present in the circuit.

The next parameter which must be calculated is the maximum allowable temperature rise, $T_{R(MAX)}$:

$$T_{R(MAX)} = T_{J(MAX)} - T_{A(MAX)} \quad (3)$$

where $T_{J(MAX)}$ is the maximum allowable junction temperature (150°C for the LM117, or 125°C for the LM317A/LM317-N), and $T_{A(MAX)}$ is the maximum ambient temperature which will be encountered in the application.

Using the calculated values for $T_{R(MAX)}$ and P_D, the maximum allowable value for the junction-to-ambient thermal resistance (θ_{JA}) can be calculated:

$$\theta_{JA} = (T_{R(MAX)} / P_D) \quad (4)$$

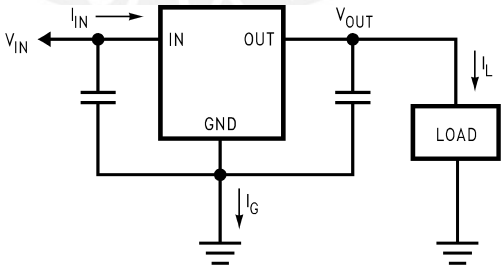


Figure 28. Power Dissipation Diagram

If the calculated maximum allowable thermal resistance is higher than the actual package rating, then no additional work is needed. If the calculated maximum allowable thermal resistance is lower than the actual package rating either the power dissipation (P_D) needs to be reduced, the maximum ambient temperature $T_{A(MAX)}$ needs to be reduced, the thermal resistance (θ_{JA}) must be lowered by adding a heatsink, or some combination of these.

If a heatsink is needed, the value can be calculated from the formula:

$$\theta_{HA} \leq (\theta_{JA} - (\theta_{CH} + \theta_{JC})) \quad (5)$$

where (θ_{CH} is the thermal resistance of the contact area between the device case and the heatsink surface, and θ_{JC} is thermal resistance from the junction of the die to surface of the package case.

When a value for $\theta_{(H-A)}$ is found using the equation shown, a heatsink must be selected that has a value that is less than, or equal to, this number.

The $\theta_{(H-A)}$ rating is specified numerically by the heatsink manufacturer in the catalog, or shown in a curve that plots temperature rise vs power dissipation for the heatsink.

Heatsinking Surface Mount Packages

The TO-263 (KTT), SOT-223 (DCY) and PFM (MDT) packages use a copper plane on the PCB and the PCB itself as a heatsink. To optimize the heat sinking ability of the plane and PCB, solder the tab of the package to the plane.

Heatsinking the SOT-223 Package

Figure 29 and Figure 30 show the information for the SOT-223 package. Figure 30 assumes a $\theta_{(J-A)}$ of 74°C/W for 1 ounce copper and 51°C/W for 2 ounce copper and a maximum junction temperature of 125°C. Please see AN-1028 (literature number SNVA036) for thermal enhancement techniques to be used with SOT-223 and PFM packages.

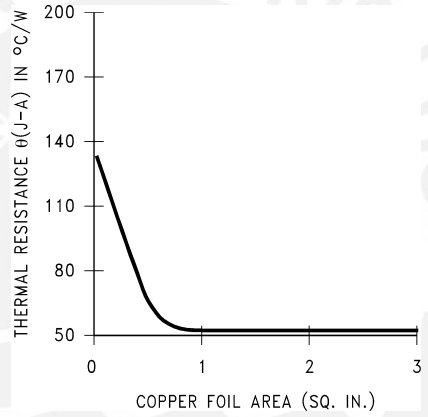


Figure 29. $\theta_{(J-A)}$ vs Copper (2 ounce) Area for the SOT-223 Package

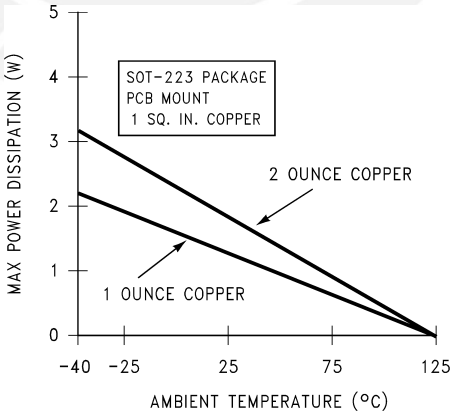


Figure 30. Maximum Power Dissipation vs T_{AMB} for the SOT-223 Package

Heatsinking the TO-263 Package

Figure 31 shows for the TO-263 the measured values of $\theta_{(J-A)}$ for different copper area sizes using a typical PCB with 1 ounce copper and no solder mask over the copper area used for heatsinking.

As shown in Figure 31, increasing the copper area beyond 1 square inch produces very little improvement. It should also be observed that the minimum value of $\theta_{(J-A)}$ for the TO-263 package mounted to a PCB is 32°C/W.

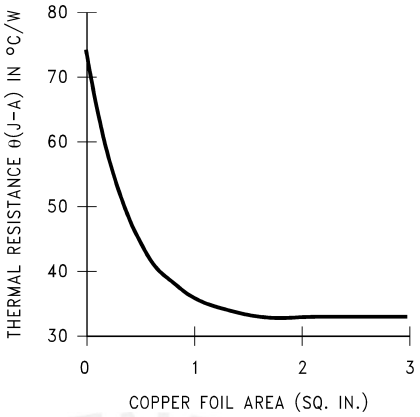


Figure 31. $\theta_{(J-A)}$ vs Copper (1 ounce) Area for the TO-263 Package

As a design aid, Figure 32 shows the maximum allowable power dissipation compared to ambient temperature for the TO-263 device (assuming $\theta_{(J-A)}$ is 35°C/W and the maximum junction temperature is 125°C).

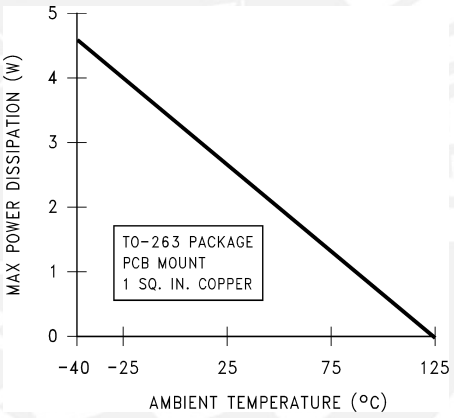


Figure 32. Maximum Power Dissipation vs T_{AMB} for the TO-263 Package

Heatsinking the PFM Package

If the maximum allowable value for θ_{JA} is found to be ≥103°C/W (Typical Rated Value) for PFM package, no heatsink is needed since the package alone will dissipate enough heat to satisfy these requirements. If the calculated value for θ_{JA} falls below these limits, a heatsink is required.

As a design aid, Table 1 shows the value of the θ_{JA} of PFM for different heatsink area. The copper patterns that we used to measure these θ_{JA}s are shown in Figure 37. Figure 33 reflects the same test results as what are in Table 1.

Figure 34 shows the maximum allowable power dissipation vs. ambient temperature for the PFM device. Figure 35 shows the maximum allowable power dissipation vs. copper area (in^2) for the PFM device. Please see AN-1028 (literature number SNVA036) for thermal enhancement techniques to be used with SOT-223 and PFM packages.

Table 1. θ_{JA} Different Heatsink Area

Layout	Copper Area		Thermal Resistance (θ_{JA}°C/W) PFM
	Top Side (in^2)(1)	Bottom Side (in^2)	
1	0.0123	0	103
2	0.066	0	87
3	0.3	0	60
4	0.53	0	54
5	0.76	0	52
6	1.0	0	47
7	0.066	0.2	84
8	0.066	0.4	70
9	0.066	0.6	63
10	0.066	0.8	57
11	0.066	1.0	57
12	0.066	0.066	89
13	0.175	0.175	72
14	0.284	0.284	61
15	0.392	0.392	55
16	0.5	0.5	53

(1) Tab of device attached to topside of copper.

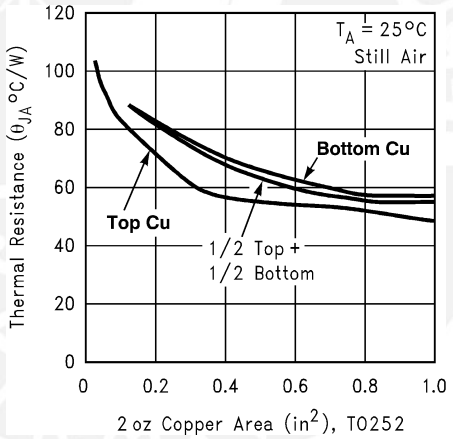


Figure 33. θ_{JA} vs 2oz Copper Area for PFM

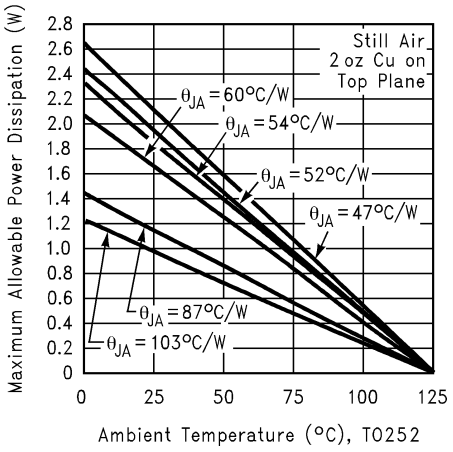


Figure 34. Maximum Allowable Power Dissipation vs. Ambient Temperature for PFM

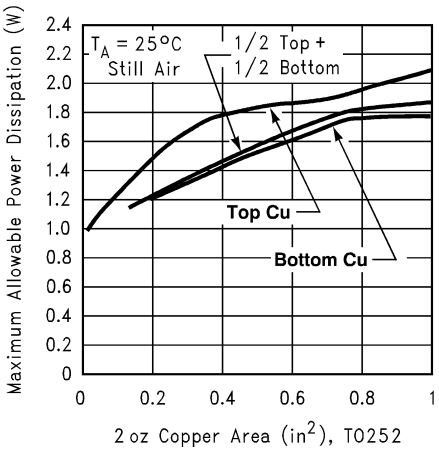


Figure 35. Maximum Allowable Power Dissipation vs. 2oz Copper Area for PFM

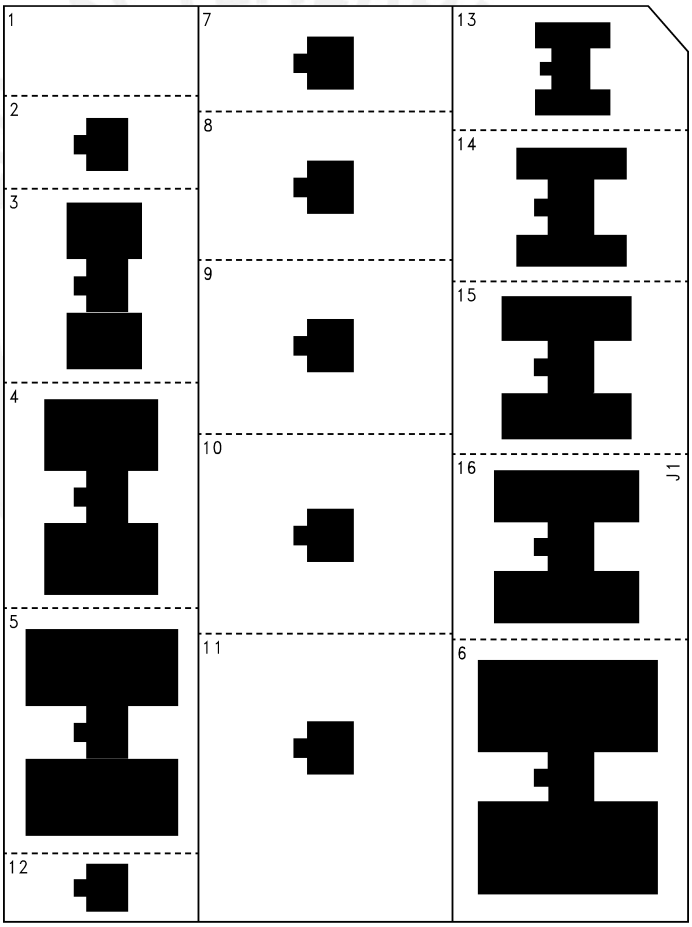


Figure 36. Top View of the Thermal Test Pattern in Actual Scale

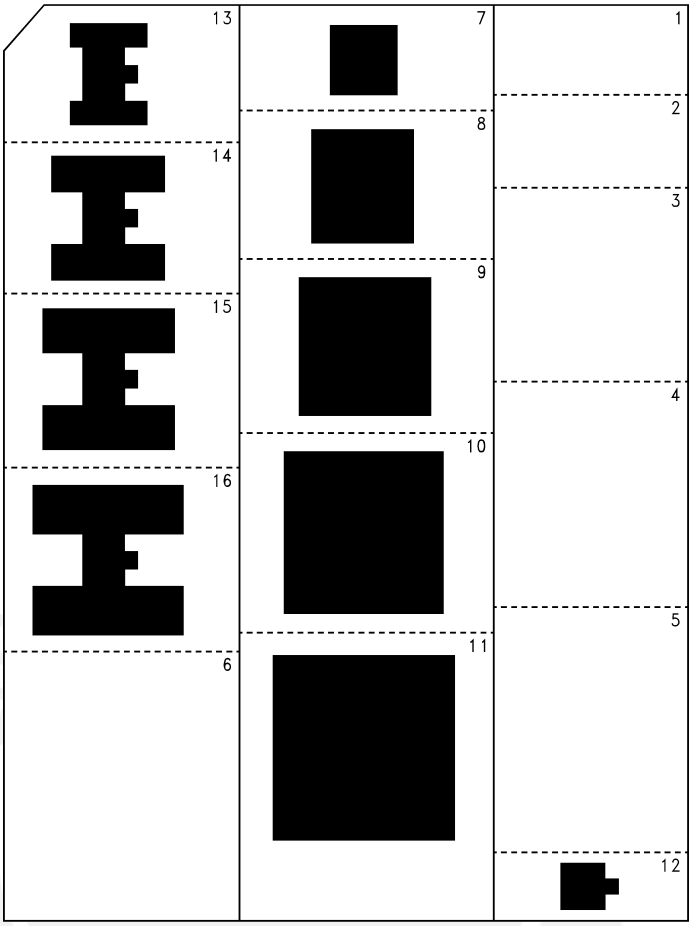
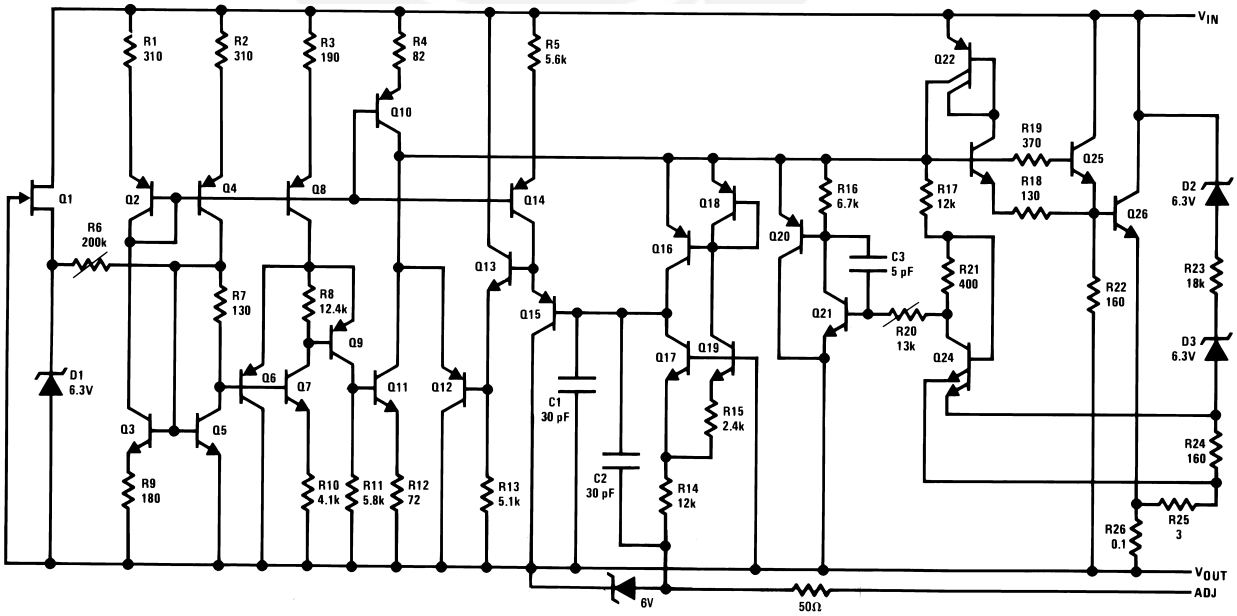
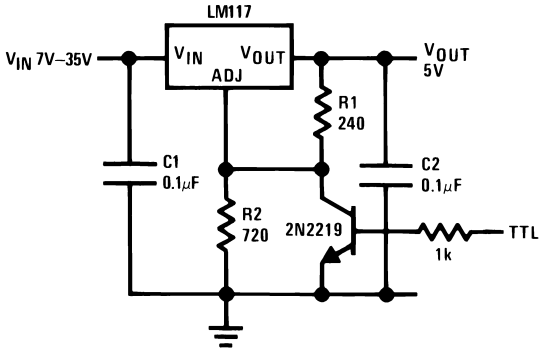


Figure 37. Bottom View of the Thermal Test Pattern in Actual Scale

Schematic Diagram



Typical Applications



Note: Min. output ≅ 1.2V

Figure 38. 5V Logic Regulator with Electronic Shutdown

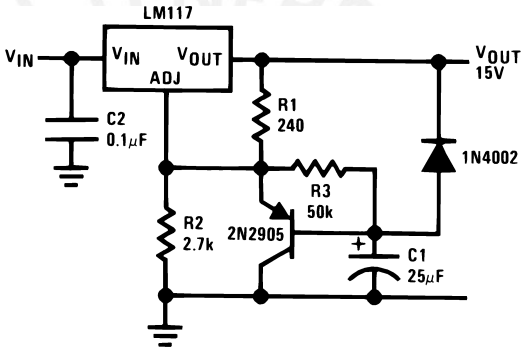
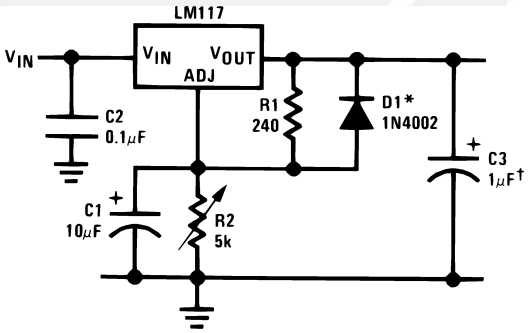


Figure 39. Slow Turn-On 15V Regulator

Figure 40.



†Solid tantalum
*Discharges C1 if output is shorted to ground

Figure 41. Adjustable Regulator with Improved Ripple Rejection

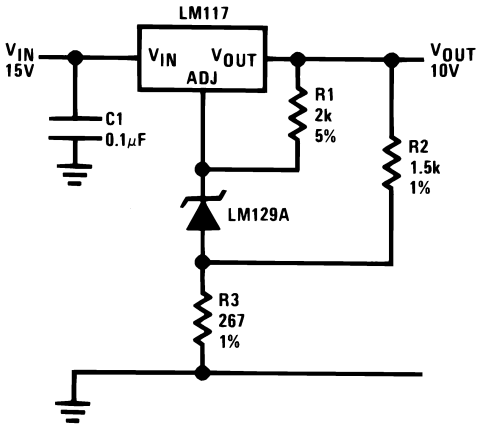
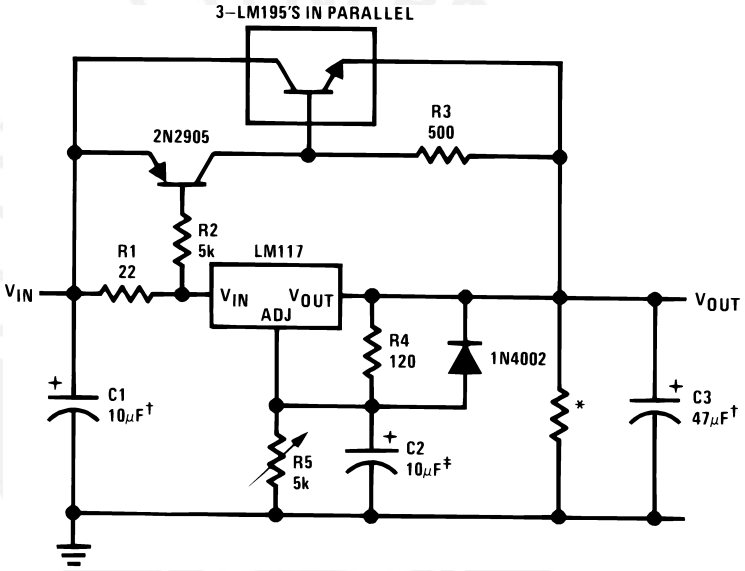
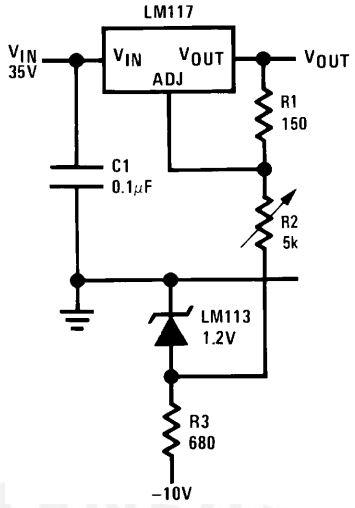


Figure 42. High Stability 10V Regulator



‡Optional—improves ripple rejection
†Solid tantalum
*Minimum load current = 30 mA

Figure 43. High Current Adjustable Regulator



Full output current not available at high input-output voltages

Figure 44. 0 to 30V Regulator

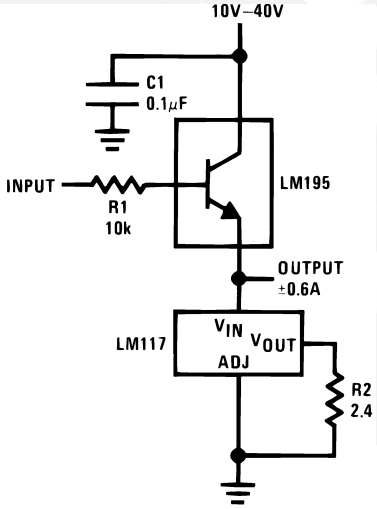
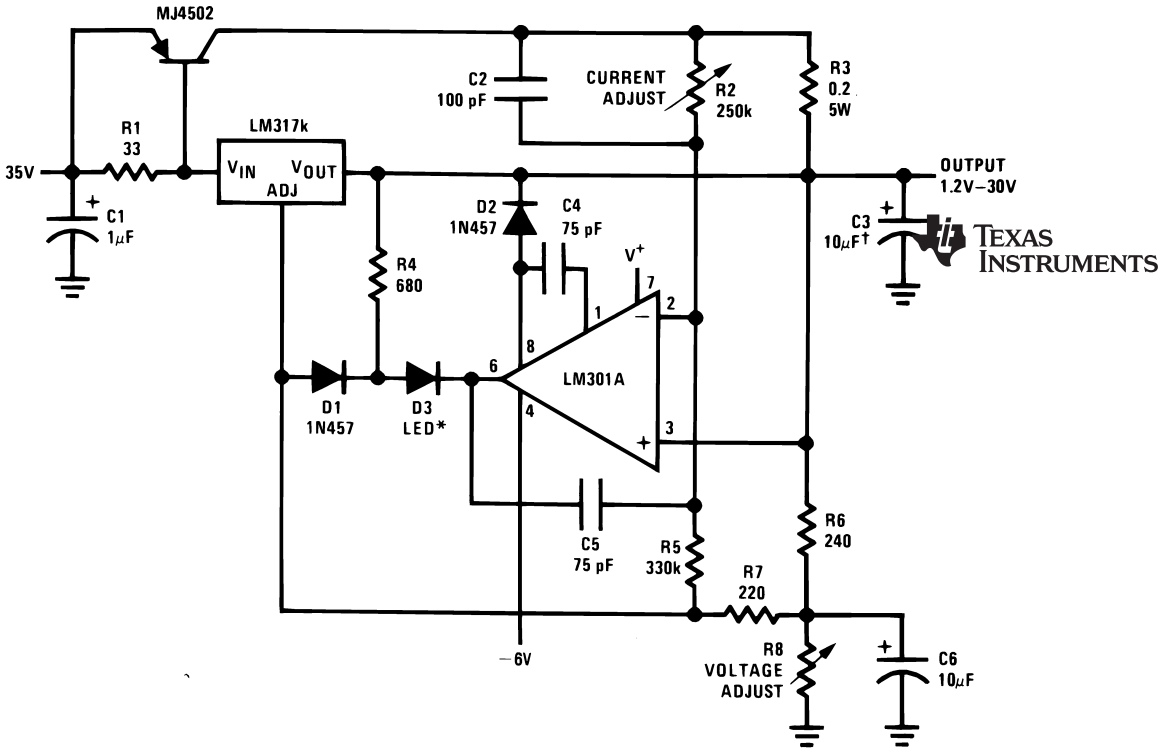


Figure 45. Power Follower



†Solid tantalum
*Lights in constant current mode

Figure 46. 5A Constant Voltage/Constant Current Regulator

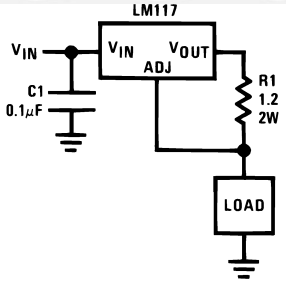
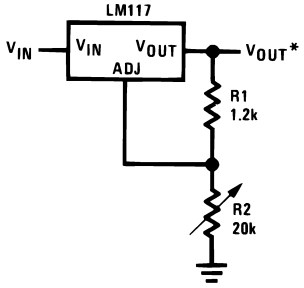


Figure 47. 1A Current Regulator



*Minimum load current ≅ 4 mA

Figure 48. 1.2V–20V Regulator with Minimum Program Current

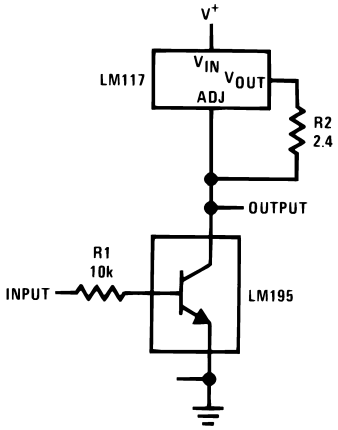
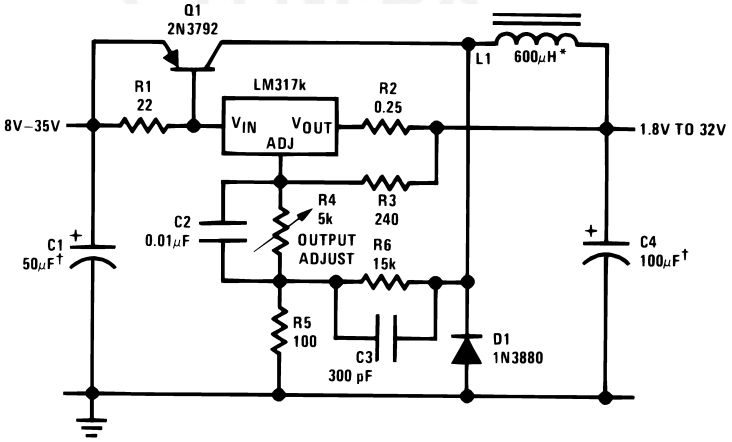
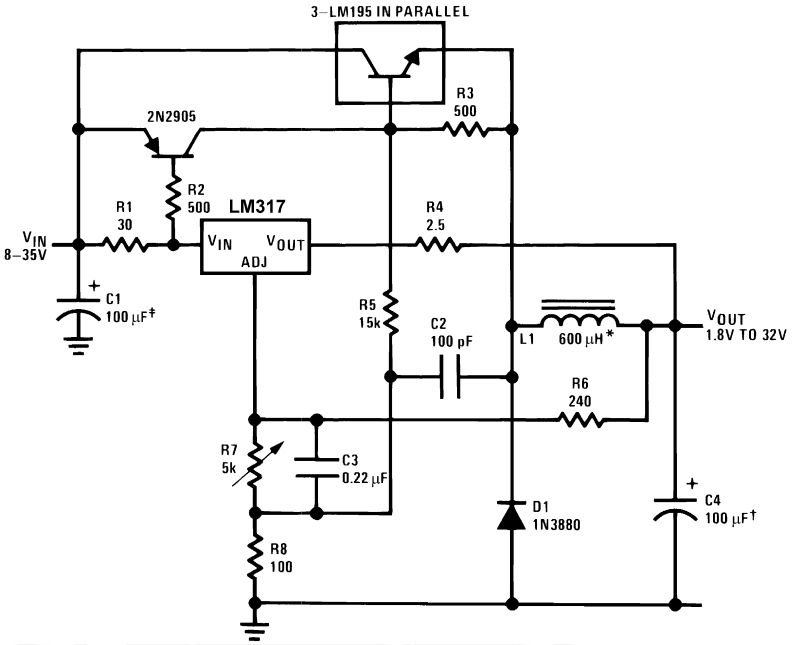


Figure 49. High Gain Amplifier



†Solid tantalum
*Core—Arnold A-254168-2 60 turns

Figure 50. Low Cost 3A Switching Regulator



†Solid tantalum
*Core—Arnold A-254168-2 60 turns

Figure 51. 4A Switching Regulator with Overload Protection

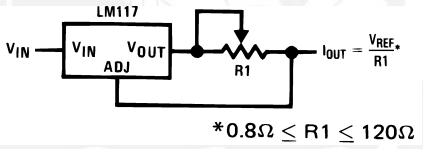


Figure 52. Precision Current Limiter

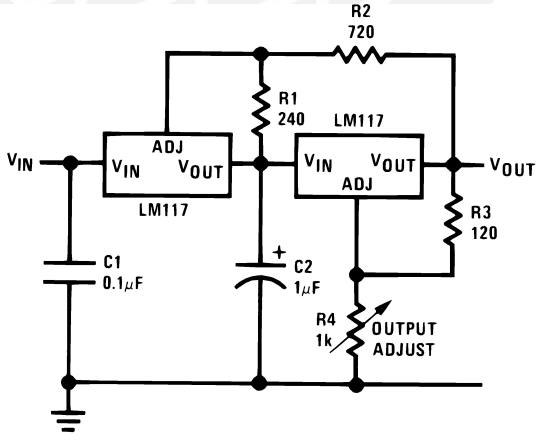
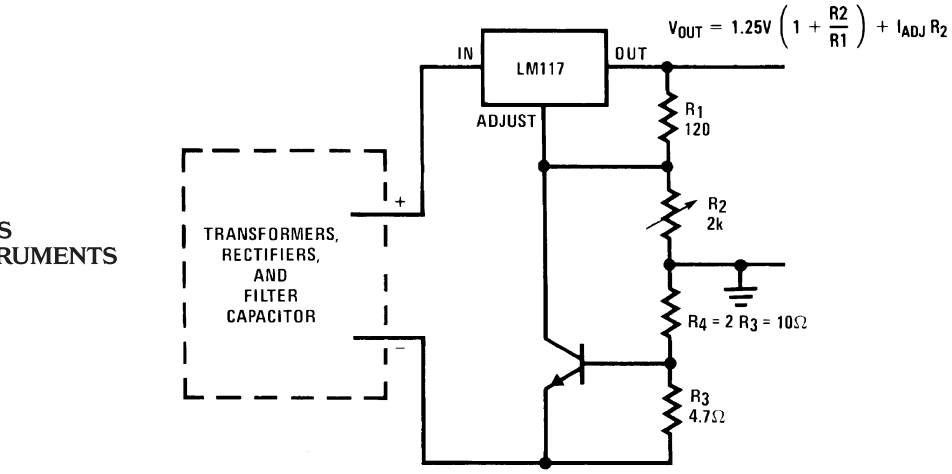


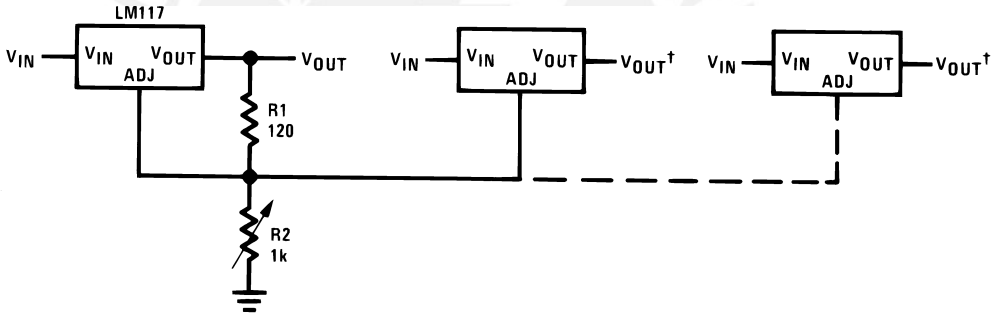
Figure 53. Tracking Preregulator



– Short circuit current is approximately $\frac{600\ mV}{R3}$, or 120 mA

(Compared to LM117's higher current limit)
—At 50 mA output only ¾ volt of drop occurs in R_3 and R_4

Figure 54. Current Limited Voltage Regulator



Note: All outputs within ±100 mV
†Minimum load—10 mA

Figure 55. Adjusting Multiple On-Card Regulators with Single Control

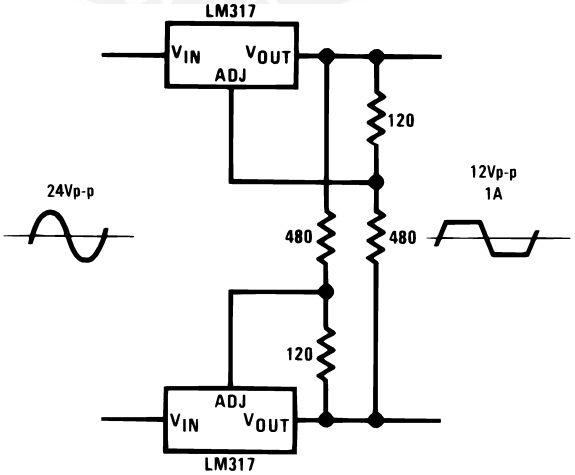
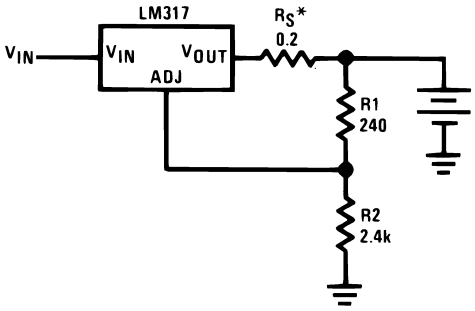


Figure 56. AC Voltage Regulator



*R_S—sets output impedance of charger: $Z_{OUT} = R_S \left(1 + \frac{R2}{R1}\right)$

Use of R_S allows low charging rates with fully charged battery.

Figure 57. 12V Battery Charger

Figure 58.

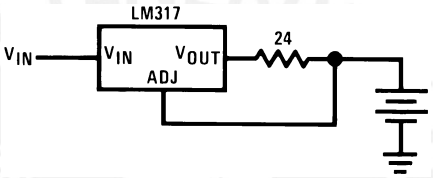


Figure 59. 50mA Constant Current Battery Charger

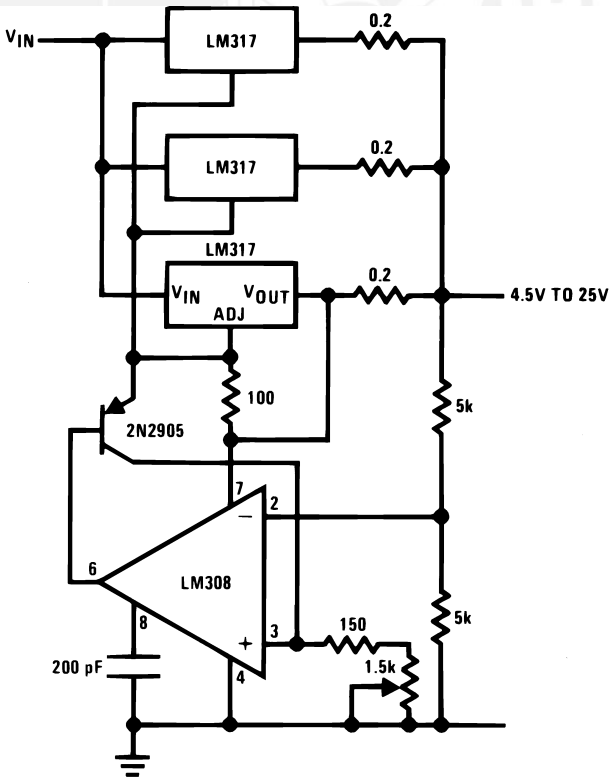
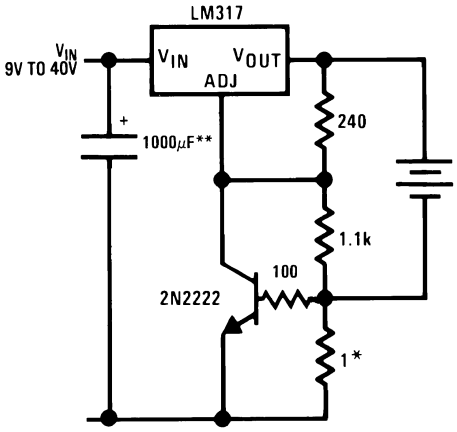
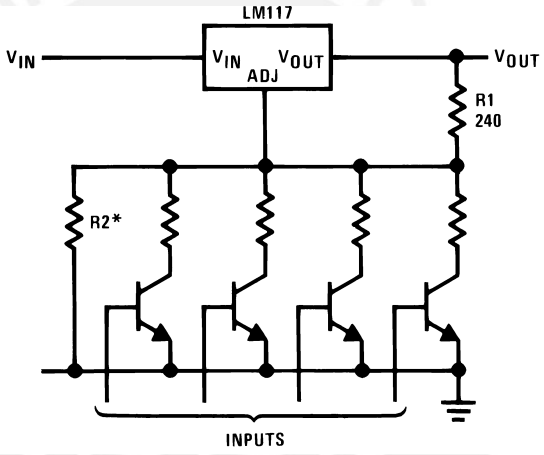


Figure 60. Adjustable 4A Regulator



*Sets peak current (0.6A for 1Ω)
**The 1000μF is recommended to filter out input transients

Figure 61. Current Limited 6V Charger



*Sets maximum V_{OUT}

Figure 62. Digitally Selected Outputs

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)	Op Temp (°C)	Top-Side Markings (4)	Samples
LM117H	ACTIVE	TO	NDT	3	500	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	-55 to 125	LM117HP+	Samples
LM117H/NOPB	ACTIVE	TO	NDT	3	500	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	-55 to 125	LM117HP+	Samples
LM117K	ACTIVE	TO-3	NDS	2	50	TBD	Call TI	Call TI	-55 to 125	LM117K STEELP+	Samples
LM117K STEEL	ACTIVE	TO-3	NDS	2	50	TBD	Call TI	Call TI	-55 to 125	LM117K STEELP+	Samples
LM117K STEEL/NOPB	ACTIVE	TO-3	NDS	2	50	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	-55 to 125	LM117K STEELP+	Samples
LM317AEMP	ACTIVE	SOT-223	DCY	4	1000	TBD	Call TI	Call TI	-40 to 125	N07A	Samples
LM317AEMP/NOPB	ACTIVE	SOT-223	DCY	4	1000	Green (RoHS & no Sb/Br)	CU SN	Level-1-260C-UNLIM	-40 to 125	N07A	Samples
LM317AEMPX	ACTIVE	SOT-223	DCY	4	2000	TBD	Call TI	Call TI		N07A	Samples
LM317AEMPX/NOPB	ACTIVE	SOT-223	DCY	4	2000	Green (RoHS & no Sb/Br)	CU SN	Level-1-260C-UNLIM	-40 to 125	N07A	Samples
LM317AH	ACTIVE	TO	NDT	3	500	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	-40 to 125	LM317AHP+	Samples
LM317AH/NOPB	ACTIVE	TO	NDT	3	500	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	-40 to 125	LM317AHP+	Samples
LM317AMDT	ACTIVE	PFM	NDP	3	75	TBD	Call TI	Call TI	-40 to 125	LM317 AMDT	Samples
LM317AMDT/NOPB	ACTIVE	PFM	NDP	3	75	Green (RoHS & no Sb/Br)	CU SN	Level-2-260C-1 YEAR	-40 to 125	LM317 AMDT	Samples
LM317AMDTX	ACTIVE	PFM	NDP	3	2500	TBD	Call TI	Call TI	-40 to 125	LM317 AMDT	Samples
LM317AMDTX/NOPB	ACTIVE	PFM	NDP	3	2500	Green (RoHS & no Sb/Br)	CU SN	Level-2-260C-1 YEAR	-40 to 125	LM317 AMDT	Samples
LM317AT	ACTIVE	TO-220	NDE	3	45	TBD	Call TI	Call TI	-40 to 125	LM317AT P+	Samples
LM317AT/NOPB	ACTIVE	TO-220	NDE	3	45	Pb-Free (RoHS Exempt)	CU SN	Level-1-NA-UNLIM	-40 to 125	LM317AT P+	Samples
LM317EMP	ACTIVE	SOT-223	DCY	4	1000	TBD	Call TI	Call TI	0 to 125	N01A	Samples

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)	Op Temp (°C)	Top-Side Markings (4)	Samples
LM317EMP/NOPB	ACTIVE	SOT-223	DCY	4	1000	Green (RoHS & no Sb/Br)	CU SN	Level-1-260C-UNLIM	0 to 125	N01A	Samples
LM317EMPX/NOPB	ACTIVE	SOT-223	DCY	4	2000	Green (RoHS & no Sb/Br)	CU SN	Level-1-260C-UNLIM	0 to 125	N01A	Samples
LM317H	ACTIVE	TO	NDT	3	500	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	0 to 125	LM317HP+	Samples
LM317H/NOPB	ACTIVE	TO	NDT	3	500	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	0 to 125	LM317HP+	Samples
LM317K STEEL	ACTIVE	TO-3	NDS	2	50	TBD	Call TI	Call TI	0 to 125	LM317K STEELP+	Samples
LM317K STEEL/NOPB	ACTIVE	TO-3	NDS	2	50	Green (RoHS & no Sb/Br)	POST-PLATE	Level-1-NA-UNLIM	0 to 125	LM317K STEELP+	Samples
LM317MDT/NOPB	ACTIVE	PFM	NDP	3	75	Green (RoHS & no Sb/Br)	CU SN	Level-2-260C-1 YEAR	0 to 125	LM317 MDT	Samples
LM317MDTX/NOPB	ACTIVE	PFM	NDP	3	2500	Green (RoHS & no Sb/Br)	CU SN	Level-2-260C-1 YEAR	0 to 125	LM317 MDT	Samples
LM317S/NOPB	ACTIVE	DDPAK/ TO-263	KTT	3	45	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR	0 to 125	LM317S P+	Samples
LM317SX/NOPB	ACTIVE	DDPAK/ TO-263	KTT	3	500	Pb-Free (RoHS Exempt)	CU SN	Level-3-245C-168 HR	0 to 125	LM317S P+	Samples
LM317T	ACTIVE	TO-220	NDE	3	45	TBD	Call TI	Call TI		LM317T P+	Samples
LM317T/LF01	ACTIVE	TO-220	NDG	3	45	Pb-Free (RoHS Exempt)	CU SN	Level-4-260C-72 HR		LM317T P+	Samples
LM317T/NOPB	ACTIVE	TO-220	NDE	3	45	Green (RoHS & no Sb/Br)	CU SN	Level-1-NA-UNLIM	0 to 125	LM317T P+	Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

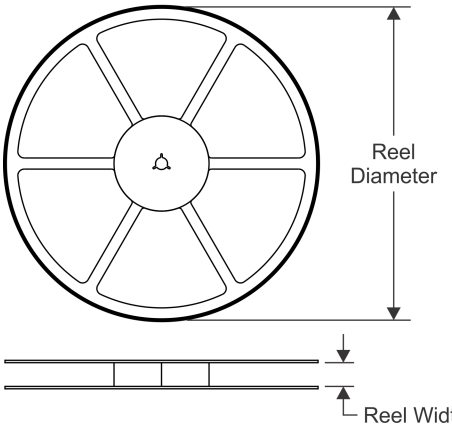
(4) Multiple Top-Side Markings will be inside parentheses. Only one Top-Side Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Top-Side Marking for that device.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

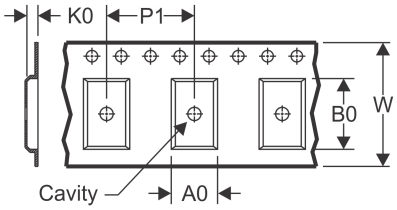
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

TAPE AND REEL INFORMATION

REEL DIMENSIONS

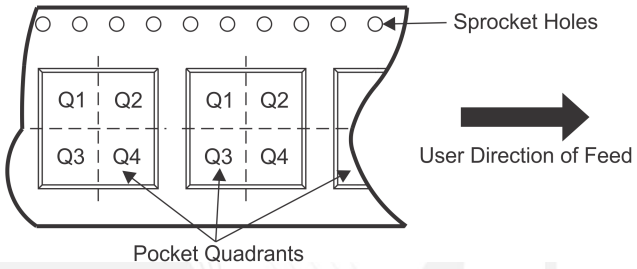


TAPE DIMENSIONS



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

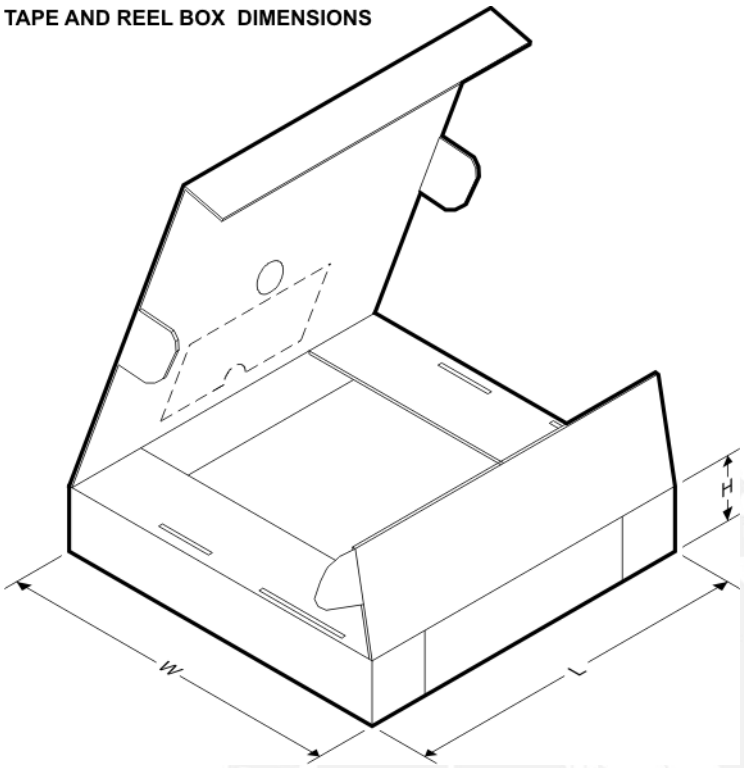
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

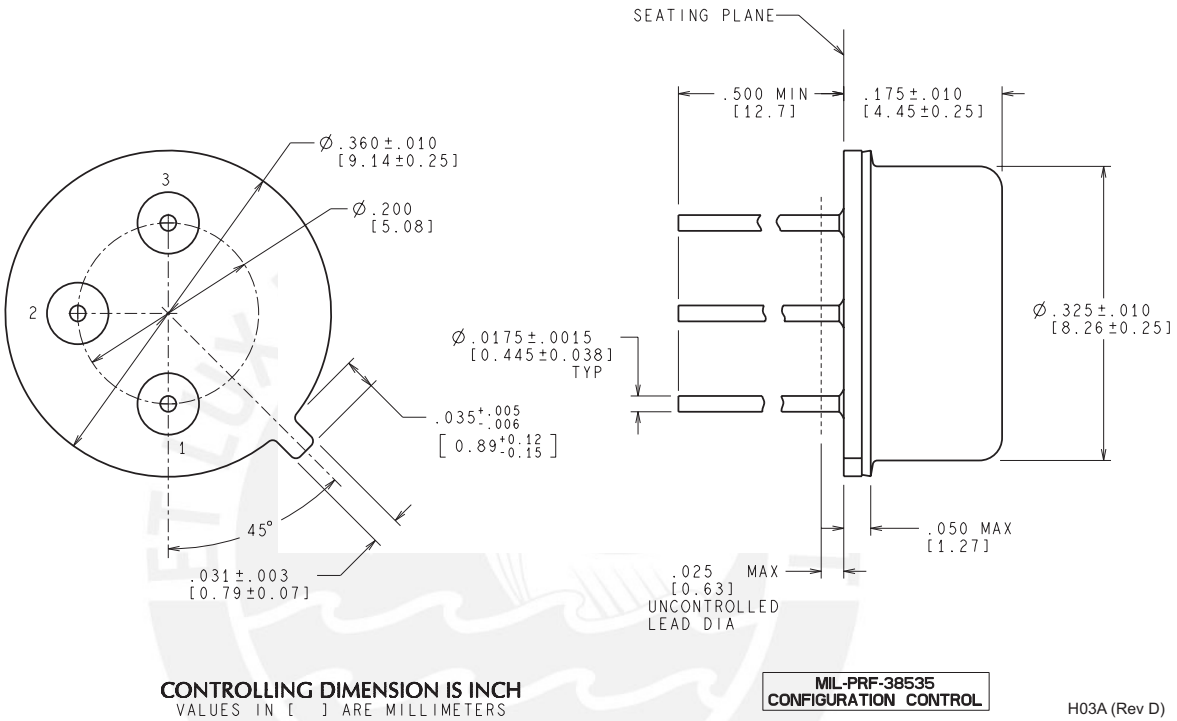
Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
LM317AEMP	SOT-223	DCY	4	1000	330.0	16.4	7.0	7.5	2.2	12.0	16.0	Q3
LM317AEMP/NOPB	SOT-223	DCY	4	1000	330.0	16.4	7.0	7.5	2.2	12.0	16.0	Q3
LM317AEMPX	SOT-223	DCY	4	2000	330.0	16.4	7.0	7.5	2.2	12.0	16.0	Q3
LM317AEMPX/NOPB	SOT-223	DCY	4	2000	330.0	16.4	7.0	7.5	2.2	12.0	16.0	Q3
LM317AMDTX	PFM	NDP	3	2500	330.0	16.4	6.9	10.5	2.7	8.0	16.0	Q2
LM317AMDTX/NOPB	PFM	NDP	3	2500	330.0	16.4	6.9	10.5	2.7	8.0	16.0	Q2
LM317EMP	SOT-223	DCY	4	1000	330.0	16.4	7.0	7.5	2.2	12.0	16.0	Q3
LM317EMP/NOPB	SOT-223	DCY	4	1000	330.0	16.4	7.0	7.5	2.2	12.0	16.0	Q3
LM317EMPX/NOPB	SOT-223	DCY	4	2000	330.0	16.4	7.0	7.5	2.2	12.0	16.0	Q3
LM317MDTX/NOPB	PFM	NDP	3	2500	330.0	16.4	6.9	10.5	2.7	8.0	16.0	Q2
LM317SX/NOPB	DDPAK/ TO-263	KTT	3	500	330.0	24.4	10.75	14.85	5.0	16.0	24.0	Q2

TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

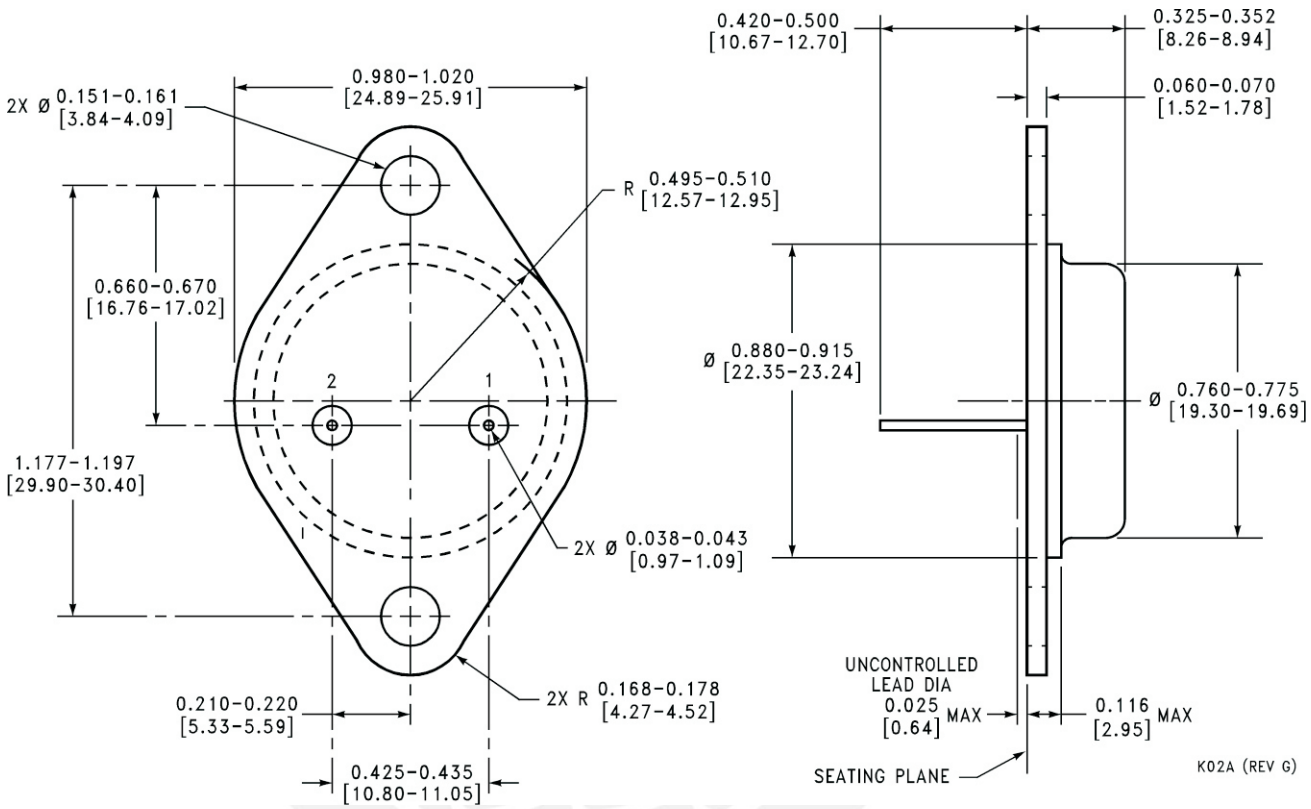
Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
LM317AEMP	SOT-223	DCY	4	1000	367.0	367.0	35.0
LM317AEMP/NOPB	SOT-223	DCY	4	1000	367.0	367.0	35.0
LM317AEMPX	SOT-223	DCY	4	2000	367.0	367.0	35.0
LM317AEMPX/NOPB	SOT-223	DCY	4	2000	367.0	367.0	35.0
LM317AMDTX	PFM	NDP	3	2500	367.0	367.0	35.0
LM317AMDTX/NOPB	PFM	NDP	3	2500	367.0	367.0	38.0
LM317EMP	SOT-223	DCY	4	1000	367.0	367.0	35.0
LM317EMP/NOPB	SOT-223	DCY	4	1000	367.0	367.0	35.0
LM317EMPX/NOPB	SOT-223	DCY	4	2000	367.0	367.0	35.0
LM317MDTX/NOPB	PFM	NDP	3	2500	367.0	367.0	38.0
LM317SX/NOPB	DDPAK/TO-263	KTT	3	500	367.0	367.0	45.0

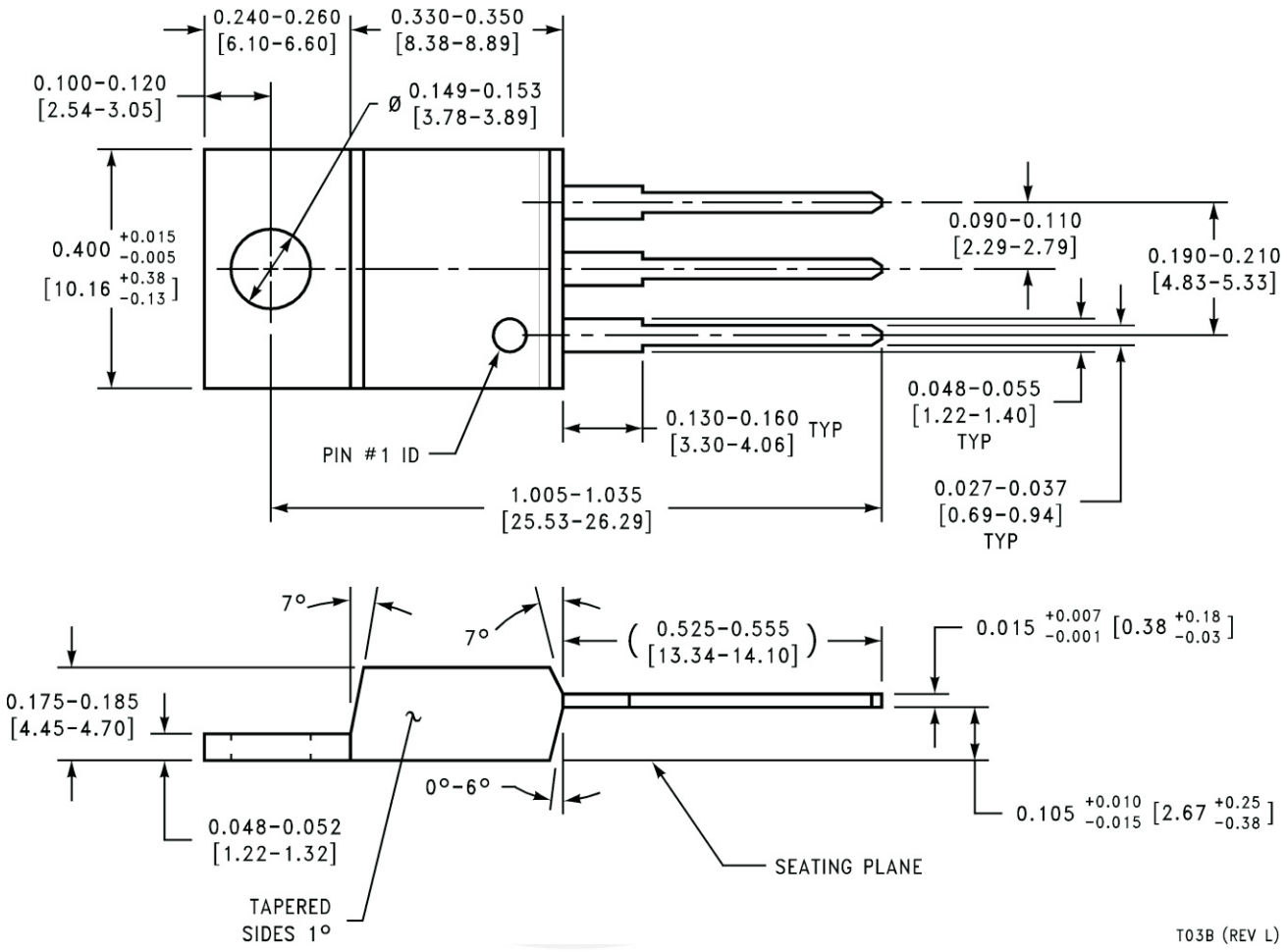


CONTROLLING DIMENSION IS INCH
VALUES IN [ ] ARE MILLIMETERS

MIL-PRF-38535
CONFIGURATION CONTROL

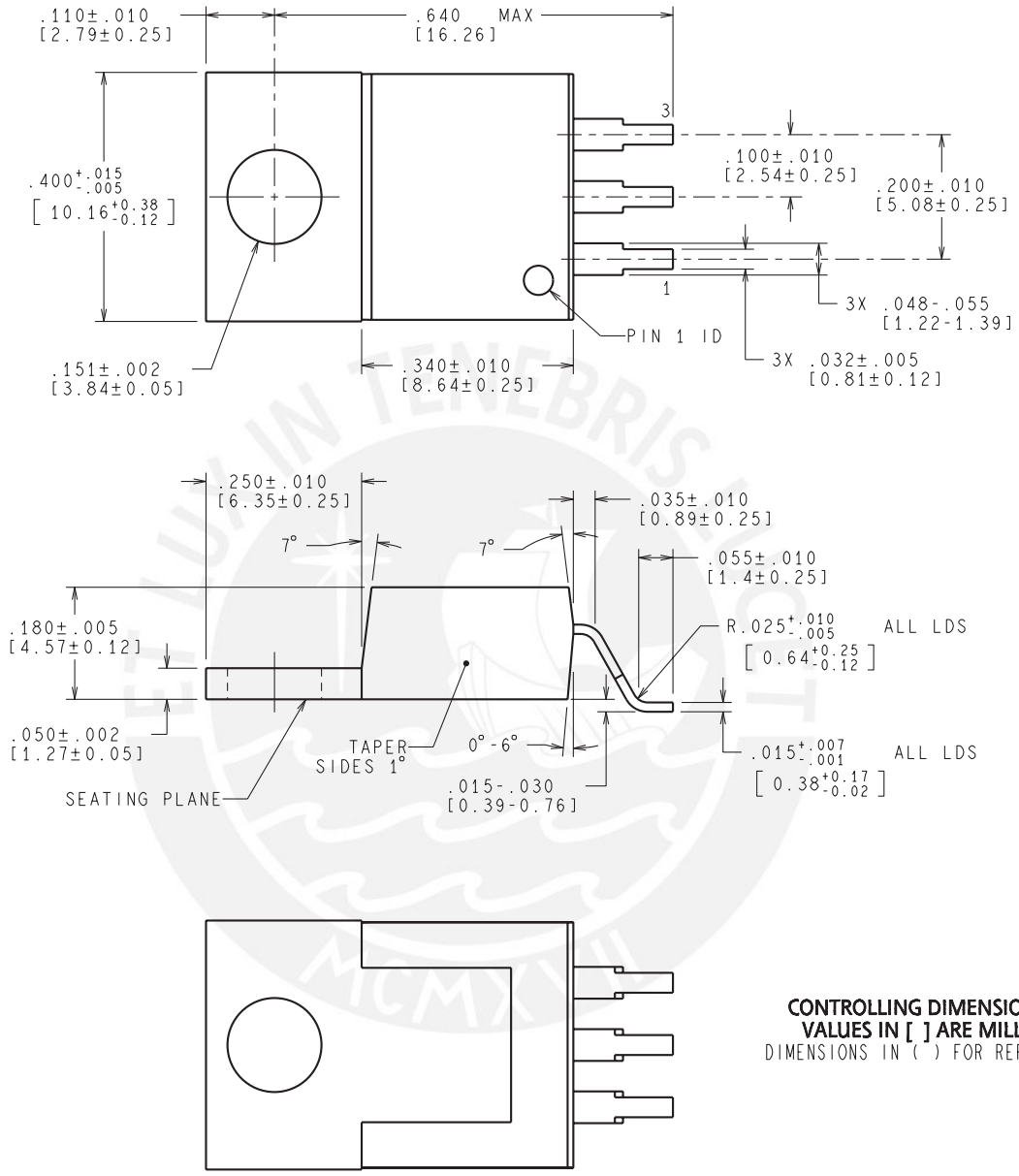
H03A (Rev D)





MECHANICAL DATA

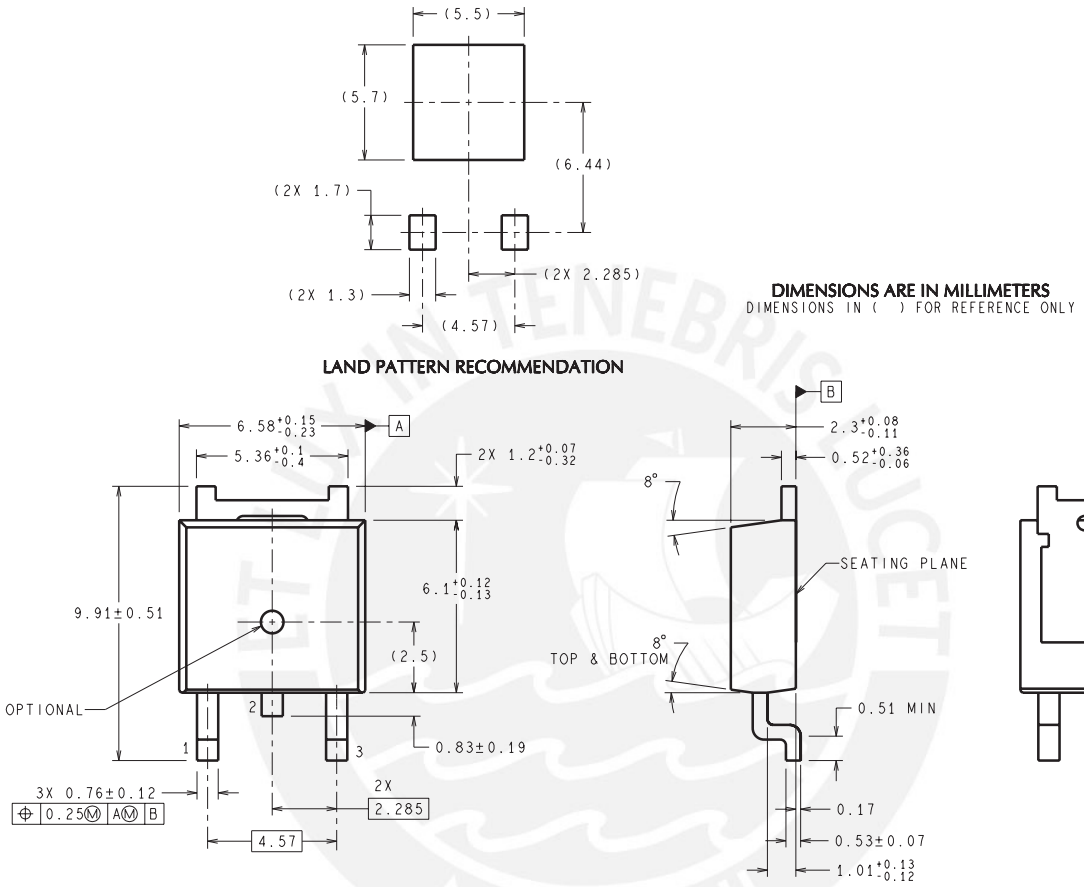
NDG0003F



CONTROLLING DIMENSION IS INCH
VALUES IN [ ] ARE MILLIMETERS
DIMENSIONS IN ( ) FOR REFERENCE ONLY

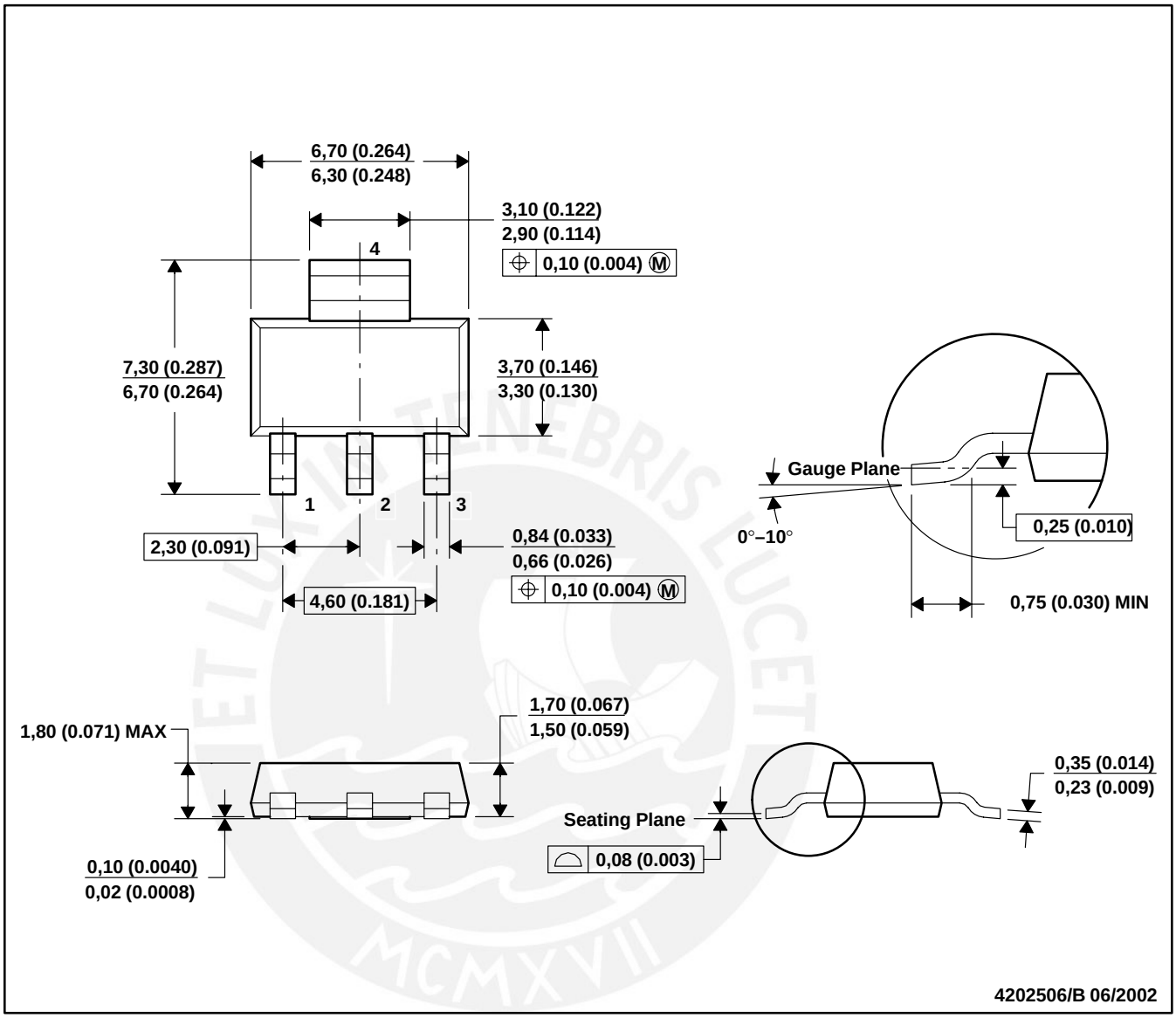
T03F (Rev B)

NDP0003B

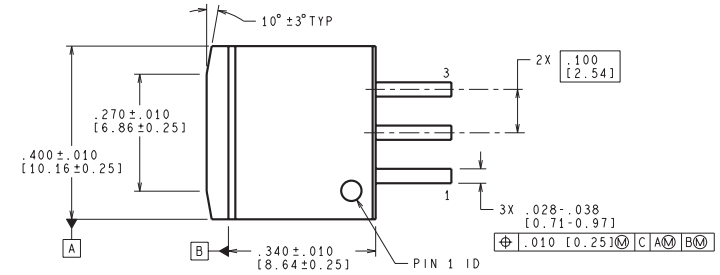


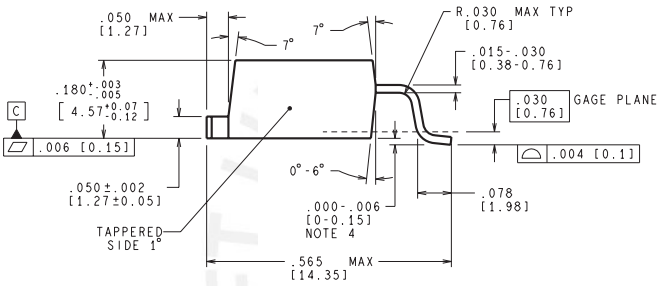
TD03B (Rev F)

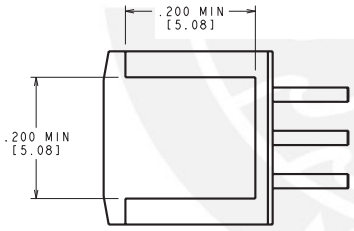
DCY (R-PDSO-G4) **PLASTIC SMALL-OUTLINE**



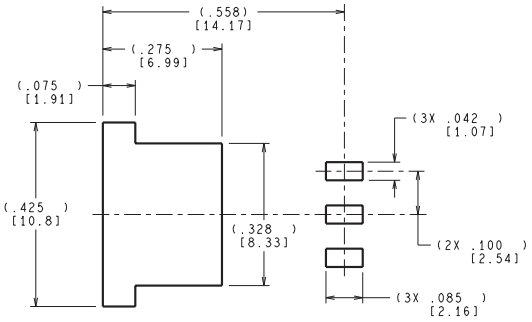
NOTES:
A. All linear dimensions are in millimeters (inches).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion.
D. Falls within JEDEC TO-261 Variation AA.







BOTTOM SIDE OF PACKAGE



LAND PATTERN RECOMMENDATION

CONTROLLING DIMENSION IS INCH
VALUES IN [ ] ARE MILLIMETERS
DIMENSIONS IN ( ) FOR REFERENCE ONLY

TS3B (Rev F)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

FAIRCHILD SEMICONDUCTOR®

LM2902,LM324/LM324A,LM224/LM224A

Quad Operational Amplifier

Features

- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain: 100dB
- Wide Power Supply Range:
  LM224/LM224A, LM324/LM324A : 3V~32V (or ±1.5 ~ 16V)
  LM2902: 3V~26V (or ±1.5V ~ 13V)
- Input Common Mode Voltage Range Includes Ground
- Large Output Voltage Swing: 0V to V_{CC} -1.5V
- Power Drain Suitable for Battery Operation

Description

The LM324/LM324A,LM2902,LM224/LM224A consist of four independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide voltage range. operation from split power supplies is also possible so long as the difference between the two supplies is 3 volts to 32 volts. Application areas include transducer amplifier, DC gain blocks and all the conventional OP Amp circuits which now can be easily implemented in single power supply systems.

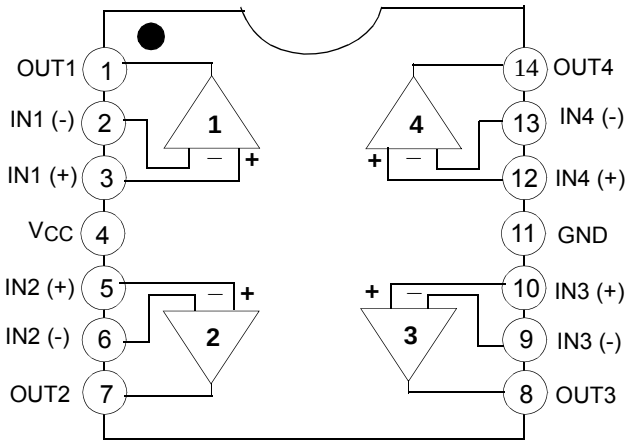
14-DIP

1

14-SOP

1

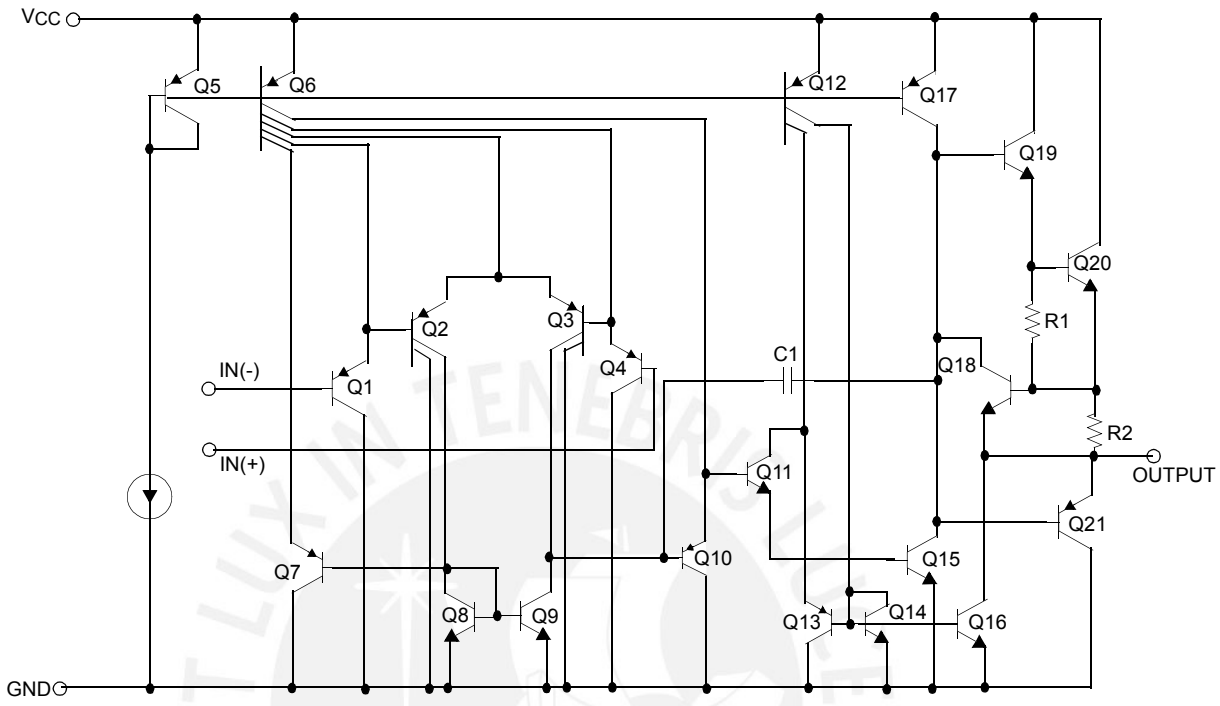
Internal Block Diagram



Rev. 1.0.4

Schematic Diagram

(One Section Only)



Absolute Maximum Ratings

Parameter	Symbol	LM224/LM224A	LM324/LM324A	LM2902	Unit
Power Supply Voltage	V_{CC}	±16 or 32	±16 or 32	±13 or 26	V
Differential Input Voltage	$V_{I(DIFF)}$	32	32	26	V
Input Voltage	V_I	-0.3 to +32	-0.3 to +32	-0.3 to +26	V
Output Short Circuit to GND Vcc≤15V, T_A=25°C(one Amp)	-	Continuous	Continuous	Continuous	-
Power Dissipation, T_A=25°C 14-DIP 14-SOP	P_D	 1310 640	 1310 640	 1310 640	mW
Operating Temperature Range	T_{OPR}	-25 ~ +85	0 ~ +70	-40 ~ +85	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	-65 ~ +150	-65 ~ +150	°C

Thermal Data

Parameter	Symbol	Value	Unit
Thermal Resistance Junction-Ambient Max. 14-DIP 14-SOP	Rθja	 95 195	°C/W

Electrical Characteristics

(V_{CC} = 5.0V, V_{EE} = GND, T_A = 25°C, unless otherwise specified)

Parameter	Symbol	Conditions	LM224			LM324			LM2902			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Input Offset Voltage	V_{IO}	V_{CM} = 0V to V_{CC} -1.5V $V_{O(P)}$ = 1.4V, R_S = 0Ω (Note1)	-	1.5	5.0	-	1.5	7.0	-	1.5	7.0	mV
Input Offset Current	I_{IO}	V_{CM} = 0V	-	2.0	30	-	3.0	50	-	3.0	50	nA
Input Bias Current	I_{BIAS}	V_{CM} = 0V	-	40	150	-	40	250	-	40	250	nA
Input Common-Mode Voltage Range	$V_{I(R)}$	Note1	0	-	V_{CC} -1.5	0	V_{CC} -1.5	-	0	-	V_{CC} -1.5	V
Supply Current	I_{CC}	R_L = ∞,V_{CC} = 30V (LM2902,V_{CC}=26V)	-	1.0	3	-	1.0	3	-	1.0	3	mA
		R_L = ∞,V_{CC} = 5V	-	0.7	1.2	-	0.7	1.2	-	0.7	1.2	mA
Large Signal Voltage Gain	G_V	V_{CC} = 15V,R_L=2kΩ $V_{O(P)}$ = 1V to 11V	50	100	-	25	100	-	25	100	-	V/ mV
Output Voltage Swing	$V_{O(H)}$	Note1, R_L = 2kΩ	26	-	-	26	-	-	22	-	-	V
		Note1, R_L=10kΩ	27	28	-	27	28	-	23	24	-	V
	$V_{O(L)}$	V_{CC} = 5V,R_L=10kΩ	-	5	20	-	5	20	-	5	100	mV
Common-Mode Rejection Ratio	CMRR	-	70	85	-	65	75	-	50	75	-	dB
Power Supply Rejection Ratio	PSRR	-	65	100	-	65	100	-	50	100	-	dB
Channel Separation	CS	f = 1kHz to 20kHz (Note2)	-	120	-	-	120	-	-	120	-	dB
Short Circuit to GND	I_{SC}	V_{CC} = 15V	-	40	60	-	40	60	-	40	60	mA
Output Current	I_{SOURCE}	$V_{I(+)}$ = 1V, $V_{I(-)}$ = 0V V_{CC} = 15V $V_{O(P)}$ = 2V	20	40	-	20	40	-	20	40	-	mA
	I_{SINK}	$V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V V_{CC} = 15V $V_{O(P)}$ = 2V	10	13	-	10	13	-	10	13	-	mA
		$V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V V_{CC} = 5V,$V_{O(R)}$ = 200mV	12	45	-	12	45	-	-	-	-	µA
Differential Input Voltage	$V_{I(DIFF)}$	-	-	-	V_{CC}	-	-	V_{CC}	-	-	V_{CC}	V

Note :

1. V_{CC}=30V for LM224 and LM324 , V_{CC} = 26V for LM2902
2. This parameter, although guaranteed, is not 100% tested in production.

Electrical Characteristics (Continued)

(V_{CC} = 5.0V, V_{EE} = GND, unless otherwise specified)
The following specification apply over the range of -25°C ≤ T_A ≤ + 85°C for the LM224; and the 0°C ≤ T_A ≤ +70°C for the LM324 ; and the -40°C ≤ T_A ≤ +85°C for the LM2902

Parameter	Symbol	Conditions	LM224			LM324			LM2902			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Input Offset Voltage	V_{IO}	V_{ICM} = 0V to V_{CC} -1.5V $V_{O(P)}$ = 1.4V, R_S = 0Ω (Note1)	-	-	7.0	-	-	9.0	-	-	10.0	mV
Input Offset Voltage Drift	$\Delta V_{IO}/\Delta T$	R_S = 0Ω (Note2)	-	7.0	-	-	7.0	-	-	7.0	-	μV/°C
Input Offset Current	I_{IO}	V_{CM} = 0V	-	-	100	-	-	150	-	-	200	nA
Input Offset Current Drift	$\Delta I_{IO}/\Delta T$	R_S = 0Ω (Note2)	-	10	-	-	10	-	-	10	-	pA/°C
Input Bias Current	I_{BIAS}	V_{CM} = 0V	-	-	300	-	-	500	-	-	500	nA
Input Common-Mode Voltage Range	$V_{I(R)}$	Note1	0	-	V_{CC} -2.0	0	-	V_{CC} -2.0	0	-	V_{CC} -2.0	V
Large Signal Voltage Gain	G_V	V_{CC} = 15V, R_L = 2.0kΩ $V_{O(P)}$ = 1V to 11V	25	-	-	15	-	-	15	-	-	V/mV
Output Voltage Swing	$V_{O(H)}$	Note1, R_L=2kΩ	26	-	-	26	-	-	22	-	-	V
		Note1, R_L=10kΩ	27	28	-	27	28	-	23	24	-	V
	$V_{O(L)}$	V_{CC} = 5V, R_L=10kΩ	-	5	20	-	5	20	-	5	100	mV
Output Current	I_{SOURCE}	$V_{I(+)}$ = 1V, $V_{I(-)}$ = 0V V_{CC} = 15V, VO(P) = 2V	10	20	-	10	20	-	10	20	-	mA
	I_{SINK}	$V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V V_{CC} = 15V, $V_{O(P)}$ = 2V	10	13	-	5	8	-	5	8	-	mA
Differential Input Voltage	$V_{I(DIFF)}$	-	-	-	V_{CC}	-	-	V_{CC}	-	-	V_{CC}	V

Note:
1. V_{CC}=30V for LM224 and LM324 , V_{CC} = 26V for LM2902
2. These parameters, although guaranteed, are not 100% tested in production.

Electrical Characteristics (Continued)

(V_{CC} = 5.0V, V_{EE} = GND, T_A = 25°C, unless otherwise specified)

Parameter	Symbol	Conditions	LM224A			LM324A			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Input Offset Voltage	V_{IO}	V_{CM} = 0V to V_{CC} -1.5V $V_{O(P)}$ = 1.4V, R_S = 0Ω (Note1)	-	1.0	3.0	-	1.5	3.0	mV
Input Offset Current	I_{IO}	V_{CM} = 0V	-	2	15	-	3.0	30	nA
Input Bias Current	I_{BIAS}	V_{CM} = 0V	-	40	80	-	40	100	nA
Input Common-Mode Voltage Range	$V_{I(R)}$	V_{CC} = 30V	0	-	V_{CC} -1.5	0	-	V_{CC} -1.5	V
Supply Current	I_{CC}	V_{CC} = 30V, R_L = ∞	-	1.5	3	-	1.5	3	mA
		V_{CC} = 5V, R_L = ∞	-	0.7	1.2	-	0.7	1.2	mA
Large Signal Voltage Gain	G_V	V_{CC} = 15V, R_L= 2kΩ $V_{O(P)}$ = 1V to 11V	50	100	-	25	100	-	V/mV
Output Voltage Swing	$V_{O(H)}$	Note1, R_L = 2kΩ	26	-	-	26	-	-	V
		Note1, R_L = 10kΩ	27	28	-	27	28	-	V
	$V_{O(L)}$	V_{CC} = 5V, R_L=10kΩ	-	5	20	-	5	20	mV
Common-Mode Rejection Ratio	CMRR	-	70	85	-	65	85	-	dB
Power Supply Rejection Ratio	PSRR	-	65	100	-	65	100	-	dB
Channel Separation	CS	f = 1kHz to 20kHz (Note2)	-	120	-	-	120	-	dB
Short Circuit to GND	I_{SC}	V_{CC} = 15V	-	40	60	-	40	60	mA
Output Current	I_{SOURCE}	$V_{I(+)}$ = 1V, $V_{I(-)}$ = 0V V_{CC} =15V, $V_{O(P)}$ = 2V	20	40	-	20	40	-	mA
	I_{SINK}	$V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V V_{CC} = 15V, $V_{O(P)}$ = 2V	10	20	-	10	20	-	mA
		$V_{I(+)}$ = 0v, $V_{I(-)}$ = 1V V_{CC} = 5V $V_{O(P)}$ = 200mV	12	50	-	12	50	-	µA
Differential Input Voltage	$V_{I(DIFF)}$	-	-	-	V_{CC}	-	-	V_{CC}	V

Note:

1. V_{CC}=30V for LM224A, LM324A
2. This parameter, although guaranteed, is not 100% tested in production.

Electrical Characteristics (Continued)

(V_{CC} = 5.0V, V_{EE} = GND, unless otherwise specified)
The following specification apply over the range of -25°C ≤ T_A ≤ +85°C for the LM224A; and the 0°C ≤ T_A ≤ +70°C for the LM324A

Parameter	Symbol	Conditions	LM224A			LM324A			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Input Offset Voltage	V_{IO}	V_{CM} = 0V to V_{CC} -1.5V $V_{O(P)}$ = 1.4V, R_S = 0Ω (Note1)	-	-	4.0	-	-	5.0	mV
Input Offset Voltage Drift	$\Delta V_{IO}/\Delta T$	R_S = 0Ω (Note2)	-	7.0	20	-	7.0	30	μV/°C
Input Offset Current	I_{IO}	V_{CM} = 0V	-	-	30	-	-	75	nA
Input Offset Current Drift	$\Delta I_{IO}/\Delta T$	R_S = 0Ω (Note2)	-	10	200	-	10	300	pA/°C
Input Bias Current	I_{BIAS}	-	-	40	100	-	40	200	nA
Input Common-Mode Voltage Range	$V_{I(R)}$	Note1	0	-	V_{CC} -2.0	0	-	V_{CC} -2.0	V
Large Signal Voltage Gain	G_V	V_{CC} = 15V, R_L= 2.0kΩ	25	-	-	15	-	-	V/mV
Output Voltage Swing	$V_{O(H)}$	Note1, R_L = 2kΩ	26	-	-	26	-	-	V
		Note1, R_L = 10kΩ	27	28	-	27	28	-	V
	$V_{O(L)}$	V_{CC} = 5V, R_L= 10kΩ	-	5	20	-	5	20	mV
Output Current	I_{SOURCE}	$V_{I(+)}$ = 1V, $V_{I(-)}$ = 0V V_{CC} = 15V, $V_{O(P)}$ = 2V	10	20	-	10	20	-	mA
	I_{SINK}	$V_{I(+)}$ = 0V, $V_{I(-)}$ = 1V V_{CC} = 15V, $V_{O(P)}$ = 2V	5	8	-	5	8	-	mA
Differential Input Voltage	$V_{I(DIFF)}$	-	-	-	V_{CC}	-	-	V_{CC}	V

Note:
1. V_{CC}=30V for LM224A and LM324A.
2. These parameters, although guaranteed, are not 100% tested in production.

Typical Performance Characteristics

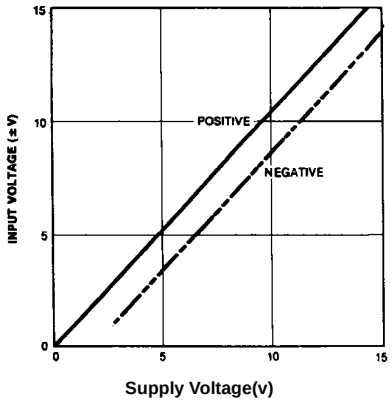


Figure 1. Input Voltage Range vs Supply Voltage

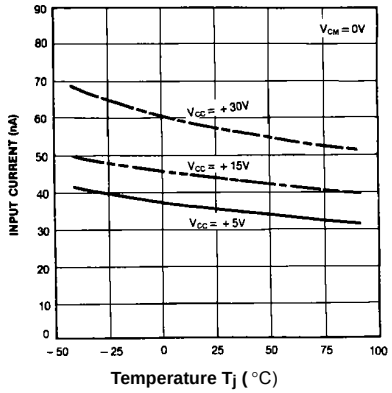


Figure 2. Input Current vs Temperature

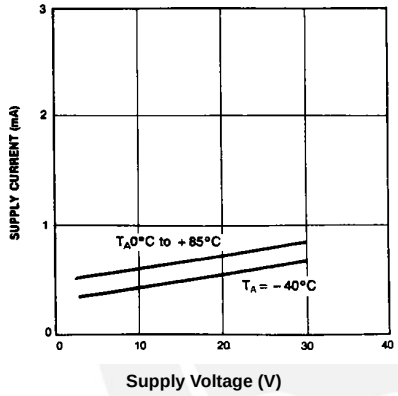


Figure 3. Supply Current vs Supply Voltage

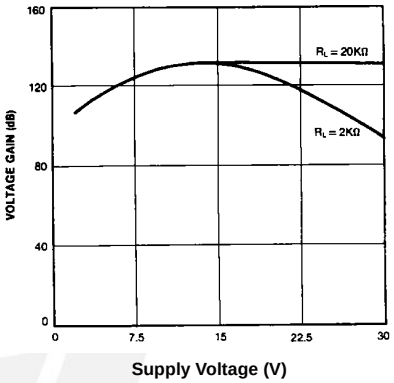


Figure 4. Voltage Gain vs Supply Voltage

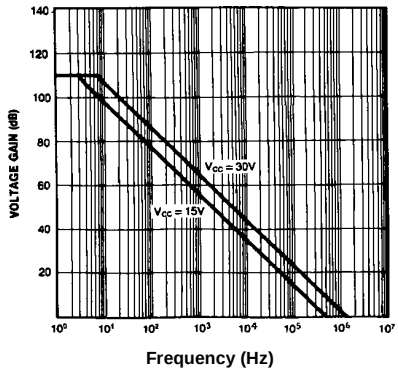


Figure 5. Open Loop Frequency Response

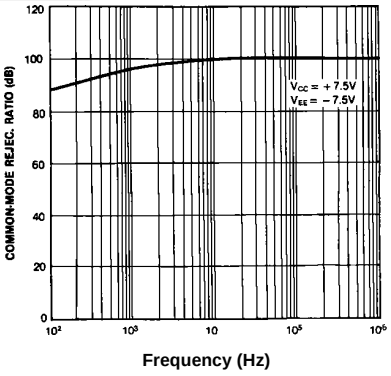


Figure 6. Common mode Rejection Ratio

Typical Performance Characteristics (Continued)

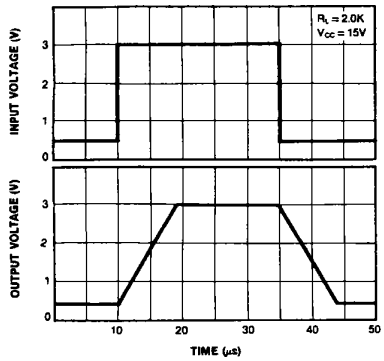


Figure 7. Voltage Follower Pulse Response

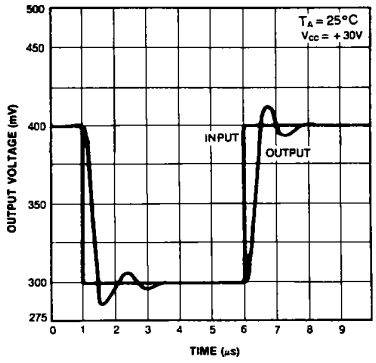


Figure 8. Voltage Follower Pulse Response (Small Signal)

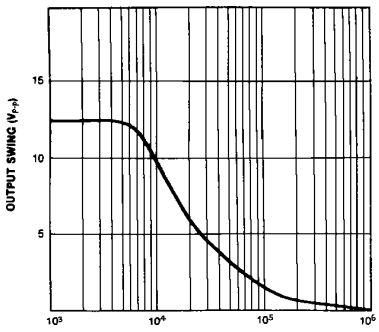


Figure 8. Large Signal Frequency Response

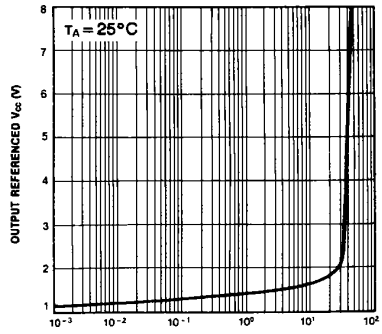


Figure 9. Output Characteristics vs Current Sourcing

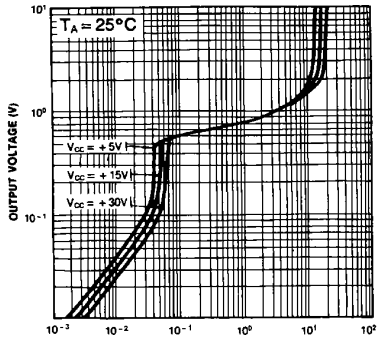


Figure 10. Output Characteristics vs Current Sinking

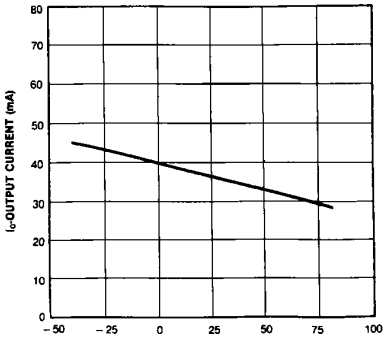


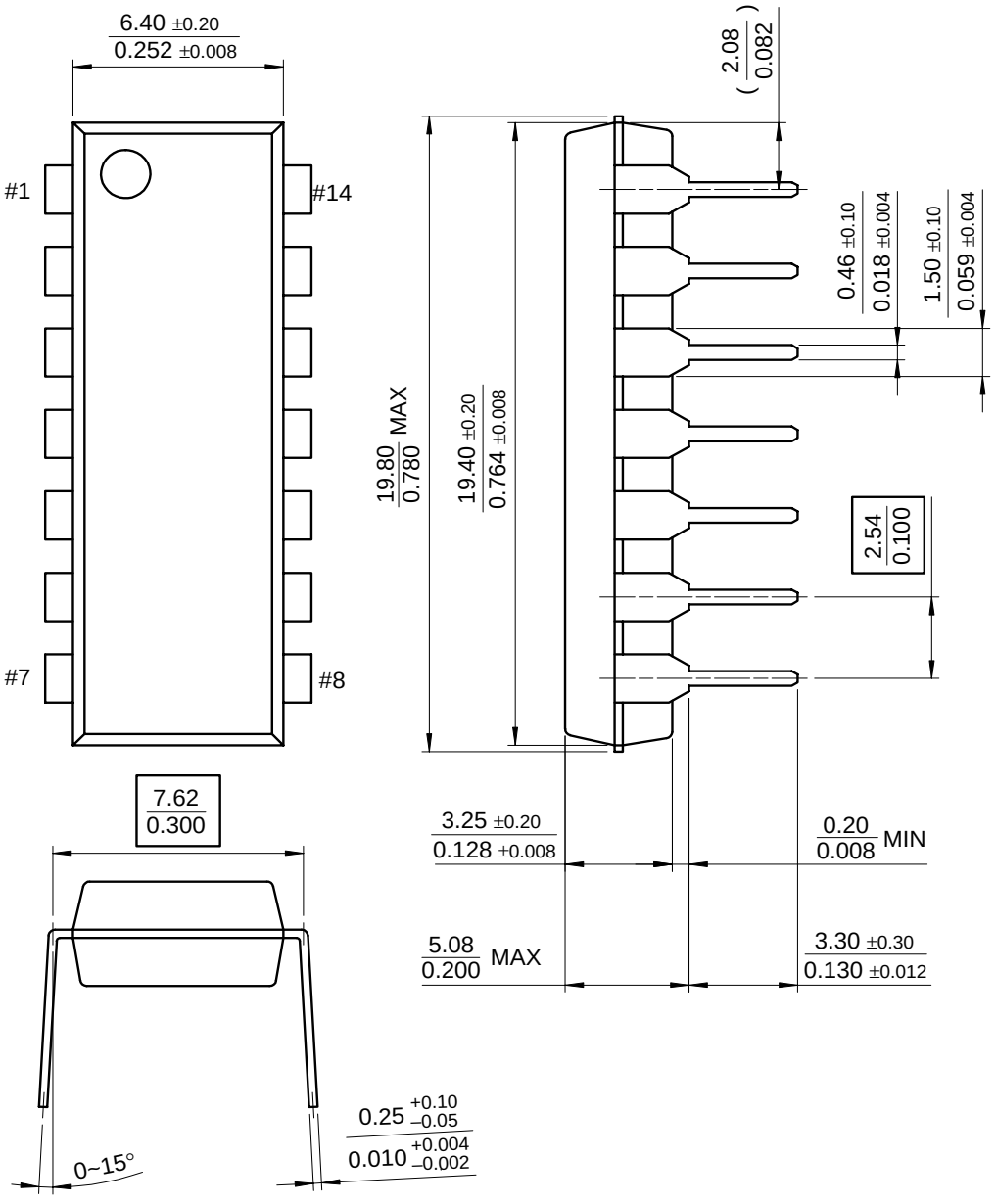
Figure 11. Current Limiting vs Temperature

Mechanical Dimensions

Package

Dimensions in millimeters

14-DIP

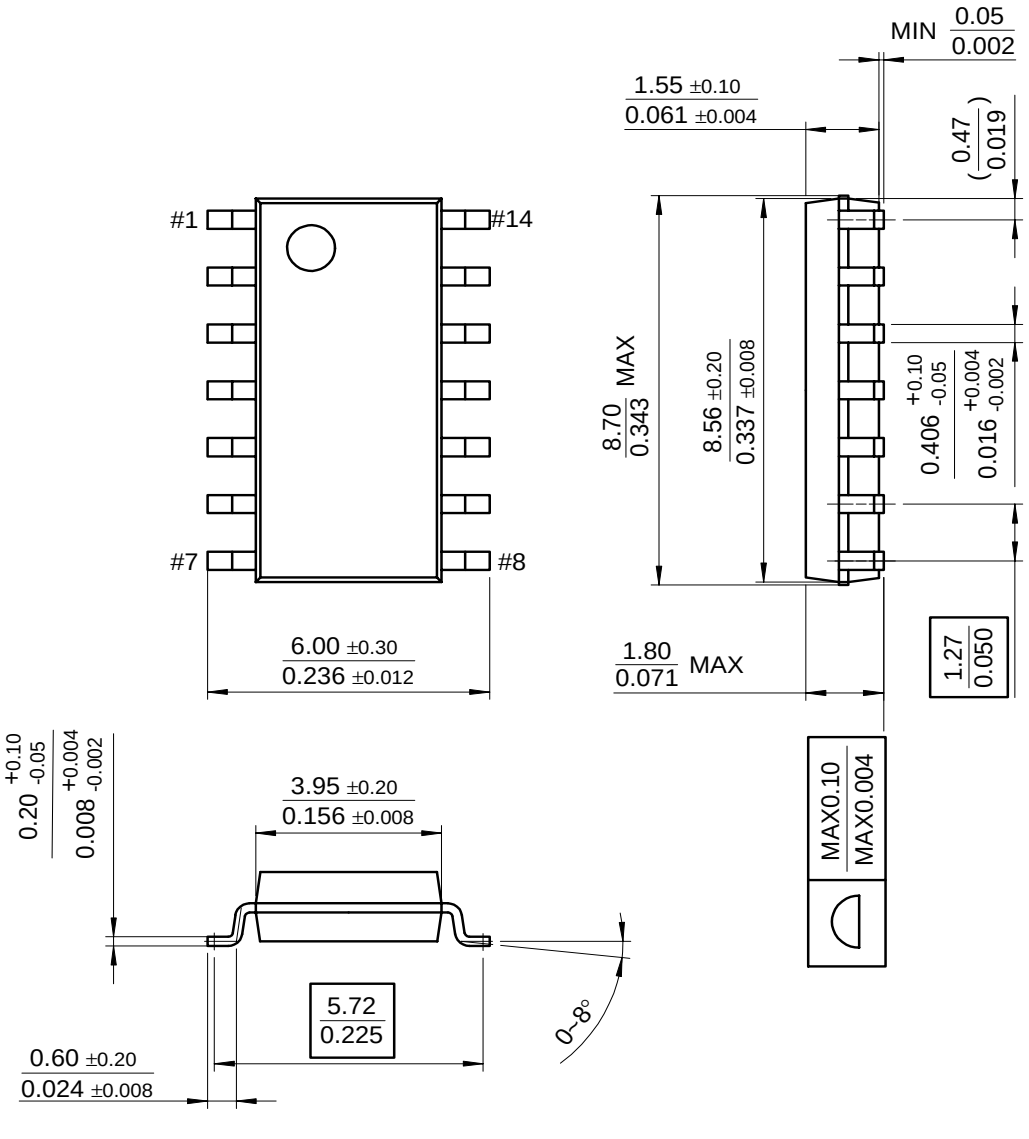


Mechanical Dimensions (Continued)

Package

Dimensions in millimeters

14-SOP



Ordering Information

Product Number	Package	Operating Temperature
LM324N	14-DIP	0 ~ +70°C
LM324AN		
LM324M	14-SOP	
LM324AM		
LM2902N	14-DIP	-40 ~ +85°C
LM2902M	14-SOP	
LM224N	14-DIP	-25 ~ +85°C
LM224AN		
LM224M	14-SOP	
LM224AM		

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

11/19/02 0.0m 001
Stock#DSxxxxxxxx

maxim integrated™

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multidrop RS-232 Networks

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Next-Generation Device Features

- **For Low-Voltage, Integrated ESD Applications MAX3222E/MAX3232E/MAX3237E/MAX3241E/ MAX3246E: +3.0V to +5.5V, Low-Power, Up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors (MAX3246E Available in a UCSP™ Package)**
- **For Low-Cost Applications MAX221E: ±15kV ESD-Protected, +5V, 1µA, Single RS-232 Transceiver with AutoShutdown™**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE+	0°C to +70°C	16 Plastic DIP
MAX220CSE+	0°C to +70°C	16 Narrow SO
MAX220CWE+	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE+	-40°C to +85°C	16 Plastic DIP
MAX220ESE+	-40°C to +85°C	16 Narrow SO
MAX220EWE+	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

+Denotes a lead(Pb)-free/RoHS-compliant package.
**Contact factory for dice specifications.*

Ordering Information continued at end of data sheet.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

19-4323; Rev 16; 7/10

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

(Voltages referenced to GND.)
V_{CC} .....-0.3V to +6V
V+ (Note 1) .....(V_{CC} - 0.3V) to +14V
V- (Note 1) .....+0.3V to -14V
Input Voltages
TIN .....-0.3V to (V_{CC} - 0.3V)
RIN (Except MAX220) .....±30V
RIN (MAX220) .....±25V
TOUT (Except MAX220) (Note 2) .....±15V
TOUT (MAX220) .....±13.2V
Output Voltages
TOUT .....±15V
ROUT .....-0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND .....Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)..842mW
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)..889mW
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) ..440mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C) ...696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C).....762mW
18-Pin Wide SO (derate 9.52mW/°C above +70°C).....762mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C)....800mW
20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C).....800mW
18-Pin CERDIP (derate 10.53mW/°C above +70°C).....842mW
Operating Temperature Ranges
MAX2_ _AC_ _, MAX2_ _C_ _ .....0°C to +70°C
MAX2_ _AE_ _, MAX2_ _E_ _ .....-40°C to +85°C
MAX2_ _AM_ _, MAX2_ _M_ _ .....-55°C to +125°C
Storage Temperature Range .....-65°C to +160°C
Lead Temperature (soldering, 10s) .....+300°C
Soldering Temperature (reflow)
20 PDIP (P20M+1) .....+225°C
All other lead(Pb)-free packages.....+260°C
All other packages containing lead(Pb) .....+240°C

Note 1: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.
Note 2: Input voltage measured with TOUT in high-impedance state, $V_{\overline{SHDN}}$ or V_{CC} = 0V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 3)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic-Low Voltage				1.4	0.8	V
Input Logic-High Voltage	All devices except MAX220		2	1.4		V
	MAX220: V_{CC} = +5.0V		2.4			
Logic Pullup/Input Current	All except MAX220, normal operation			5	40	µA
	$V_{\overline{SHDN}}$ = 0V, MAX222/MAX242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	V_{CC} = +5.5V, $V_{\overline{SHDN}}$ = 0V, V_{OUT} = ±15V, MAX222/MAX242			±0.01	±10	µA
	V_{CC} = $V_{\overline{SHDN}}$ = 0V	V_{OUT} = ±15V		±0.01	±10	
		MAX220, V_{OUT} = ±12V			±25	
Data Rate				200	116	kbps
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V	V_{OUT} = 0V	±7	±22		mA
		MAX220			±60	
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
		MAX220			±25	
RS-232 Input Threshold Low	V_{CC} = +5V	All except MAX243 R2IN	0.8	1.3		V
		MAX243 R2IN (Note 4)	-3			
RS-232 Input Threshold High	V_{CC} = +5V	All except MAX243 R2IN		1.8	2.4	V
		MAX243 R2IN (Note 4)		-0.5	-0.1	

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 3)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Hysteresis	All except MAX220/MAX243, V_{CC} = +5V, no hysteresis in shutdown		0.2	0.5	1.0	V
	MAX220			0.3		
	MAX243			1		
RS-232 Input Resistance	T_A = +25°C (MAX220)		3	5	7	kΩ
			3	5	7	
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
	I_{OUT} = 1.6mA (MAX220)				0.4	
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = V_{GND}		-2	-10		mA
	Sinking V_{OUT} = V_{CC}		10	30		
TTL/CMOS Output Leakage Current	$V_{\overline{SHDN}}$ = V_{CC} or $V_{\overline{EN}}$ = V_{CC} ($V_{\overline{SHDN}}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA
EN Input Threshold Low	MAX242			1.4	0.8	V
EN Input Threshold High	MAX242		2.0	1.4		V
Supply Voltage Range			4.5		5.5	V
V_{CC} Supply Current ($V_{\overline{SHDN}}$ = V_{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
		MAX222/MAX232A/MAX233A/ MAX242/MAX243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/MAX232A/MAX233A/ MAX242/MAX243		15		
Shutdown Supply Current	MAX222/ MAX242	T_A = +25°C		0.1	10	µA
		T_A = 0°C to +70°C		2	50	
		T_A = -40°C to +85°C		2	50	
		T_A = -55°C to +125°C		35	100	
$\overline{SHDN}$ Input Leakage Current	MAX222/MAX242				±1	µA
$\overline{SHDN}$ Threshold Low	MAX222/MAX242			1.4	0.8	V
$\overline{SHDN}$ Threshold High	MAX222/MAX242		2.0	1.4		V
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = +5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/MAX232A/ MAX233/MAX242/MAX243	6	12	30	V/µs
		MAX220	1.5	3	30.0	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation)	t_{PHLT}, Figure 1	MAX222/MAX232A/ MAX233/MAX242/MAX243		1.3	3.5	µs
		MAX220		4	10	
	t_{PLHT}, Figure 1	MAX222/MAX232A/ MAX233/MAX242/MAX243		1.5	3.5	
		MAX220		5	10	

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 3)

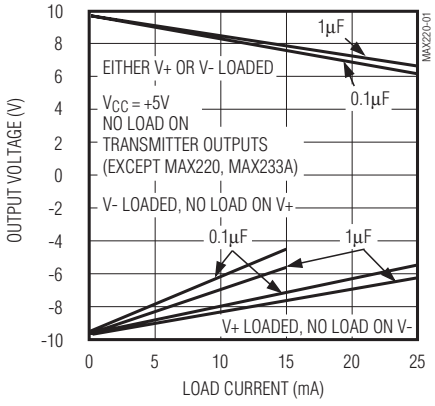
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Receiver Propagation Delay RS-232 to TLL (Normal Operation)	t_{PHLR}, Figure 2	MAX222/MAX232A/MAX233/ MAX242/MAX243		0.5	1	µs
		MAX220		0.6	3	
	t_{PLHR}, Figure 2	MAX222/MAX232A/MAX233/ MAX242/MAX243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (Shutdown)	t_{PHLS}, Figure 2	MAX242		0.5	10	µs
	t_{PHLS}, Figure 2	MAX242		2.5	10	
Receiver-Output Enable Time	t_{ER}	MAX242, Figure 3		125	500	ns
Receiver-Output Disable Time	t_{DR}	MAX242, Figure 3		160	500	ns
Transmitter-Output Enable Time ($\overline{SHDN}$ Goes High)	t_{ET}	MAX222/MAX242, 0.1µF caps (includes charge-pump start-up), Figure 4		250		µs
Transmitter-Output Disable Time ($\overline{SHDN}$ Goes Low)	t_{DT}	MAX222/MAX242, 0.1µF caps, Figure 4		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t_{PHLT} - t_{PLHT}	MAX222/MAX232A/MAX233/ MAX242/MAX243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	t_{PHLR} - t_{PLHR}	MAX222/MAX232A/MAX233/ MAX242/MAX243		100		ns
		MAX220		225		

Note 3: All units are production tested at hot. Specifications over temperature are guaranteed by design.
Note 4: MAX243 R2OUT is guaranteed to be low when R2IN ≥ 0V or is unconnected.

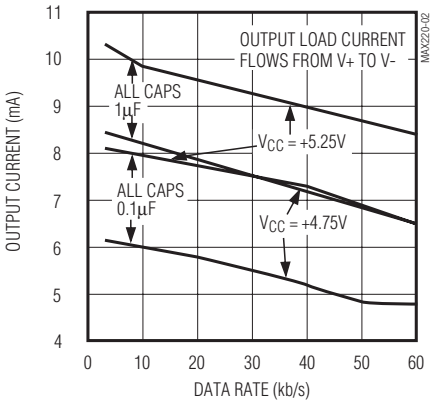
Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243

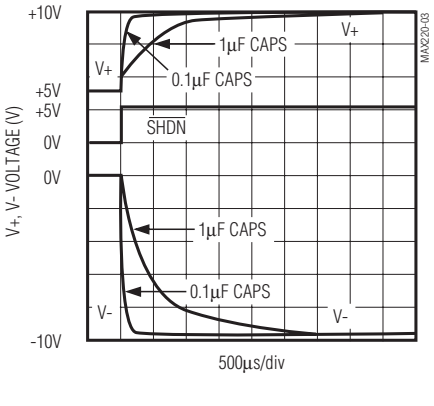
OUTPUT VOLTAGE vs. LOAD CURRENT



AVAILABLE OUTPUT CURRENT vs. DATA RATE



MAX222/MAX242 ON-TIME EXITING SHUTDOWN



+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230–MAX241

(Voltages referenced to GND.)
V_{CC} .....-0.3V to +6V
V+ .....(V_{CC} - 0.3V) to +14V
V- .....+0.3V to -14V
Input Voltages
TIN .....-0.3V to (V_{CC} + 0.3V)
RIN .....±30V
Output Voltages
TOUT .....(V+ + 0.3V) to (V- - 0.3V)
ROUT .....-0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, TOUT to GND .....Continuous
Continuous Power Dissipation (T_A = +70°C)
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C)....800mW
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)....842mW
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C)....889mW
24-Pin Narrow Plastic DIP
(derate 13.33mW/°C above +70°C) 1.07W
24-Pin Plastic DIP (derate 9.09mW/°C above +70°C).....500mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C).....762mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C).....800mW
24-Pin Wide SO (derate 11.76mW/°C above +70°C).....941mW
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
44-Pin Plastic FP (derate 11.11mW/°C above +70°C) 889mW
14-Pin CERDIP (derate 9.09mW/°C above +70°C).....727mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C).....800mW
20-Pin CERDIP (derate 11.11mW/°C above +70°C).....889mW
24-Pin Narrow CERDIP
(derate 12.50mW/°C above +70°C) 1W
24-Pin Sidebraze (derate 20.0mW/°C above +70°C).....1.6W
28-Pin SSOP (derate 9.52mW/°C above +70°C).....762mW
Operating Temperature Ranges
MAX2 _ _ C _ _.....0°C to +70°C
MAX2 _ _ E _ _.....-40°C to +85°C
MAX2 _ _ M _ _.....-55°C to +125°C
Storage Temperature Range .....-65°C to +160°C
Lead Temperature (soldering, 10s) .....+300°C
Soldering Temperature (reflow)
20 PDIP (P20M+1) .....+225°C
24 PDIP (P24M-1) .....+225°C
All other lead(Pb)-free packages.....+260°C
All other packages containing lead(Pb) .....+240°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10%; MAX233/MAX235, V_{CC} = +5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = +5V ±10%; V+ = +7.5V to +13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.) (Note 5)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground		±5.0	±7.3		V
V_{CC} Supply Current	No load, T_A = +25°C	MAX232/233		5	10	mA
		MAX223/230/234–238/240/241		7	15	
		MAX231/239		0.4	1	
V+ Supply Current		MAX231		1.8	5	mA
		MAX239		5	15	
Shutdown Supply Current	T_A = +25°C	MAX223		15	50	µA
		MAX230/235/236/240/241		1	10	
Input Logic-Low Voltage	TIN, EN, $\overline{SHDN}$ (MAX233); $\overline{EN}$, SHDN (MAX230/235–241)				0.8	V
Input Logic-High Voltage	TIN		2.0			V
	EN, $\overline{SHDN}$ (MAX223); $\overline{EN}$, SHDN (MAX230/235/236/240/241)		2.4			
Logic Pullup Current	V_{TIN} = 0V			1.5	200	µA
Receiver Input Voltage Operating Range			-30		+30	V

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241 (continued)

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10%; MAX233/MAX235, V_{CC} = +5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = +5V ±10%; V+ = +7.5V to +13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.) (Note 5)

PARAMETER	CONDITIONS			MIN	TYP	MAX	UNITS
RS-232 Input Logic-Low Voltage	T_A = +25°C, V_{CC} = +5V	Normal operation $V_{\overline{SHDN}}$ = +5V (MAX223) V_{SHDN} = 0V (MAX235/236/240/241)		0.8	1.2		V
		Shutdown (MAX223) $V_{\overline{SHDN}}$ = 0V, V_{EN} = +5V (R4IN, R5IN)		0.6	1.5		
RS-232 Input Logic-High Voltage	T_A = +25°C, V_{CC} = +5V	Normal operation $V_{\overline{SHDN}}$ = 5V (MAX223) V_{SHDN} = 0V (MAX235/236/240/241)			1.7	2.4	V
		Shutdown (MAX223) $V_{\overline{SHDN}}$ = 0V, V_{EN} = +5V (R4IN, R5IN)			1.5	2.4	
RS-232 Input Hysteresis	V_{CC} = +5V, no hysteresis in shutdown			0.2	0.5	1.0	V
RS-232 Input Resistance	T_A = +25°C, V_{CC} = +5V			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 1.6mA (MAX231/232/233, I_{OUT} = 3.2mA)					0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1mA			3.5	V_{CC} - 0.4		V
TTL/CMOS Output Leakage Current	0V ≤ R_{OUT} ≤ V_{CC}; V_{EN} = 0V (MAX223); $V_{\overline{EN}}$ = V_{CC} (MAX235–241)				±0.05	±10	µA
Receiver Output Enable Time	Normal operation	MAX223			600		ns
		MAX235/236/239/240/241			400		
Receiver Output Disable Time	Normal operation	MAX223			900		ns
		MAX235/236/239/240/241			250		
Propagation Delay	RS-232 IN to TTL/CMOS OUT, C_L = 150pF	Normal operation			0.5	10	µs
		$V_{\overline{SHDN}}$ = 0V (MAX223)	t_{PHLS}		4	40	
			t_{PLHS}		6	40	
Transition Region Slew Rate	MAX223/MAX230/MAX234–241, T_A = +25°C, V_{CC} = +5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V			3	5.1	30	V/µs
	MAX231/MAX232/MAX233, T_A = +25°C, V_{CC} = +5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V				4	30	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V			300			Ω
Transmitter Output Short-Circuit Current					±10		mA

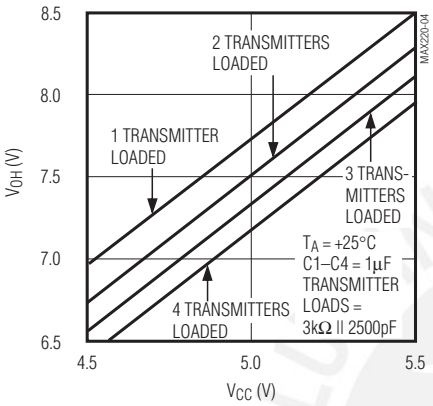
Note 5: All units are production tested at hot except for the MAX240, which is production tested at T_A = +25°C. Specifications over temperature are guaranteed by design.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

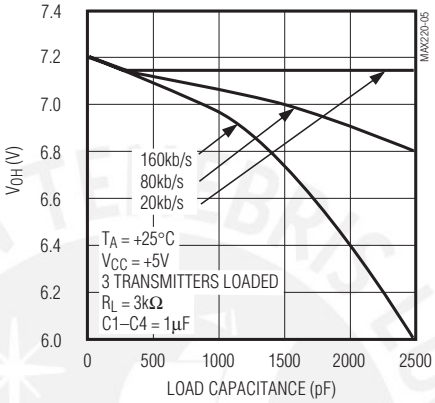
Typical Operating Characteristics

MAX223/MAX230–MAX241

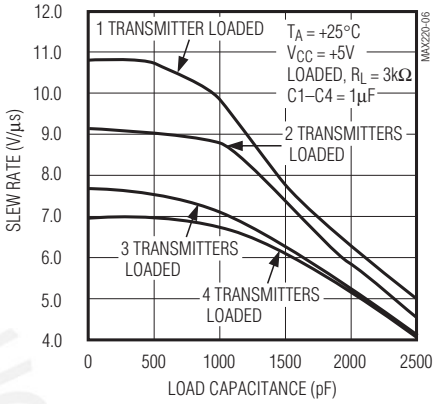
TRANSMITTER OUTPUT VOLTAGE (V_{OH}) vs. V_{CC}



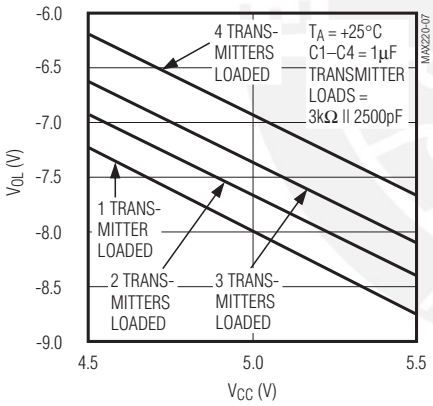
TRANSMITTER OUTPUT VOLTAGE (V_{OH}) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



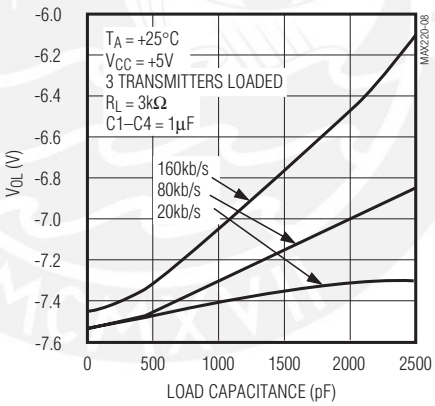
TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



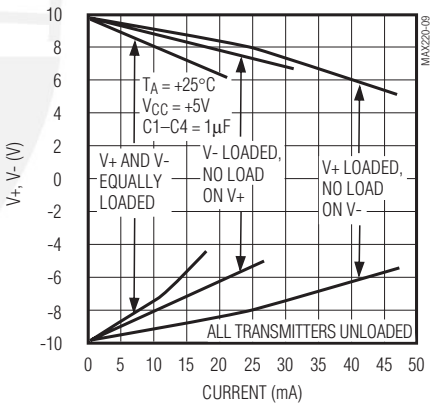
TRANSMITTER OUTPUT VOLTAGE (V_{OL}) vs. V_{CC}



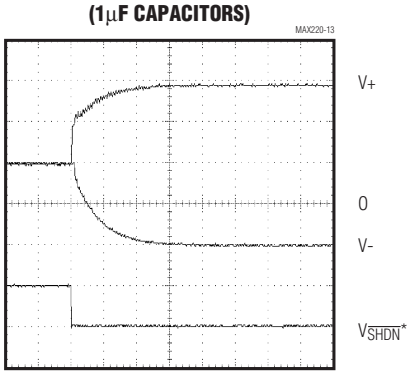
TRANSMITTER OUTPUT VOLTAGE (V_{OL}) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



TRANSMITTER OUTPUT VOLTAGE (V+, V-) vs. LOAD CURRENT



V+, V- WHEN EXITING SHUTDOWN (1µF CAPACITORS)



*SHUTDOWN POLARITY IS REVERSED

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX225/MAX244–MAX249

(Voltages referenced to GND.)
Supply Voltage (V_{CC}) .....-0.3V to +6V
Input Voltages
TIN, $\overline{ENA}$, $\overline{ENB}$, $\overline{ENR}$, $\overline{ENT}$, $\overline{ENRA}$, $\overline{ENRB}$, $\overline{ENTA}$, $\overline{ENTB}$.....-0.3V to (V_{CC} + 0.3V)
RIN .....±25V
TOUT (Note 6).....±15V
ROUT.....-0.3V to (V_{CC} + 0.3V)
Short Circuit Duration (one output at a time)
TOUT to GND .....Continuous
ROUT to GND.....Continuous

Continuous Power Dissipation (T_A = +70°C)
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
40-Pin Plastic DIP (derate 11.11mW/°C above +70°C) ...611mW
44-Pin PLCC (derate 13.33mW/°C above +70°C) 1.07W
Operating Temperature Ranges
MAX225C_ _, MAX24_C_ _ .....0°C to +70°C
MAX225E_ _, MAX24_E_ _ .....-40°C to +85°C
Storage Temperature Range .....-65°C to +160°C
Lead Temperature (soldering, 10s)).....+300°C
Soldering Temperature (reflow)
40 PDIP (P40M-2) .....+225°C
All other lead(Pb)-free packages.....+260°C
All other packages containing lead(Pb) .....+240°C

Note 6: Input voltage measured with transmitter output in a high-impedance state, shutdown, or V_{CC} = 0V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX225/MAX244–MAX249

(MAX225, V_{CC} = +5.0V ±5%; MAX244–MAX249, V_{CC} = +5.0V ±10%, external capacitors C1–C4 = 1µF; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.) (Note 7)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Input Logic-Low Voltage				1.4	0.8	V
Input Logic-High Voltage			2	1.4		V
Logic Pullup/Input Current	Tables 1a–1d	Normal operation		10	50	µA
		Shutdown		±0.01	±1	
Data Rate	Tables 1a–1d, normal operation			120	64	kbps
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±7.5		V
Output Leakage Current (Shutdown)	Tables 1a–1d	$V_{\overline{ENA}}$, $V_{\overline{ENB}}$, $V_{\overline{ENT}}$, $V_{\overline{ENTA}}$, $V_{\overline{ENTB}}$ = V_{CC}, V_{OUT} = ±15V		±0.01	±25	µA
		V_{CC} = 0V, V_{OUT} = ±15V		±0.01	±25	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V (Note 8)		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V		±7	±30		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±25	V
RS-232 Input Logic-Low Voltage	V_{CC} = +5V		0.8	1.3		V
RS-232 Input Logic-High Voltage	V_{CC} = +5V			1.8	2.4	V
RS-232 Input Hysteresis	V_{CC} = +5V		0.2	0.5	1.0	V
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = V_{GND}		-2	-10		mA
	Sinking V_{OUT} = V_{CC}		10	30		
TTL/CMOS Output Leakage Current	Normal operation, outputs disabled, Tables 1a–1d, 0V ≤ V_{OUT} ≤ V_{CC}, $V_{\overline{ENR\_}}$ = V_{CC}			±0.05	±0.10	µA

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ELECTRICAL CHARACTERISTICS—MAX225/MAX244–MAX249 (continued)

(MAX225, V_{CC} = +5.0V ±5%; MAX244–MAX249, V_{CC} = +5.0V ±10%, external capacitors C1–C4 = 1µF; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.) (Note 7)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
POWER SUPPLY AND CONTROL LOGIC						
Supply Voltage Range		MAX225	4.75		5.25	V
		MAX244–MAX249	4.5		5.5	
V_{CC} Supply Current (Normal Operation)	No load	MAX225		10	20	mA
		MAX244–MAX249		11	30	
	3kΩ loads on all outputs	MAX225		40		
		MAX244–MAX249		57		
Shutdown Supply Current	T_A = +25°C			8	25	µA
	T_A = T_{MIN} to T_{MAX}				50	
Control Input	Leakage current				±1	µA
	Logic-low voltage			1.4	0.8	V
	Logic-high voltage		2.4	1.4		
AC CHARACTERISTICS						
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = +5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V		5	10	30	V/µs
Transmitter Propagation Delay TLL to RS-232 (Normal Operation)	t_{PHLT}, Figure 1			1.3	3.5	µs
	t_{PLHT}, Figure 1			1.5	3.5	
Receiver Propagation Delay TLL to RS-232 (Normal Operation)	t_{PHLR}, Figure 2			0.6	1.5	µs
	t_{PLHR}, Figure 2			0.6	1.5	
Receiver Propagation Delay TLL to RS-232 (Low-Power Mode)	t_{PHLS}, Figure 2			0.6	10	µs
	t_{PLHS}, Figure 2			3.0	10	
Transmitter + to - Propagation Delay Difference (Normal Operation)	t_{PHLT} - t_{PLHT}			350		ns
Receiver + to - Propagation Delay Difference (Normal Operation)	t_{PHLR} - t_{PLHR}			350		ns
Receiver-Output Enable Time	t_{ER}, Figure 3			100	500	ns
Receiver-Output Disable Time	t_{DR}, Figure 3			100	500	ns
Transmitter Enable Time	t_{ET}	MAX246–MAX249 (excludes charge-pump startup)		5		µs
		MAX225/MAX245–MAX249 (includes charge-pump startup)		10		ms
Transmitter Disable Time	t_{DT}, Figure 4			100		ns

Note 7: All units production tested at hot. Specifications over temperature are guaranteed by design.

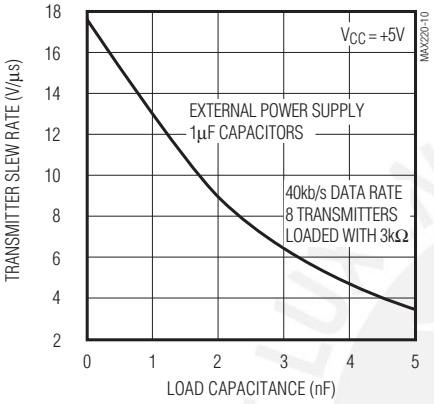
Note 8: The 300Ω minimum specification complies with EIA/TIA-232E, but the actual resistance when in shutdown mode or V_{CC} = 0V is 10MΩ as is implied by the leakage specification.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

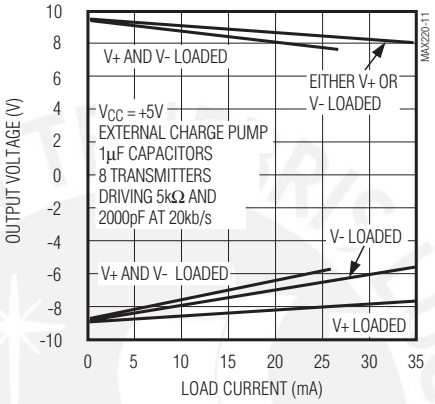
Typical Operating Characteristics

MAX225/MAX244–MAX249

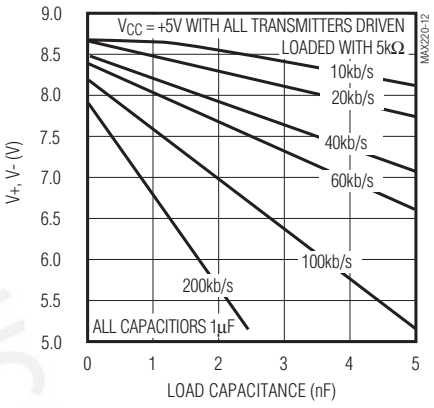
TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



OUTPUT VOLTAGE vs. LOAD CURRENT FOR V+ AND V-



TRANSMITTER OUTPUT VOLTAGE (V+, V-) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



+5V-Powered, Multichannel RS-232 Drivers/Receivers

Test Circuits/Timing Diagrams

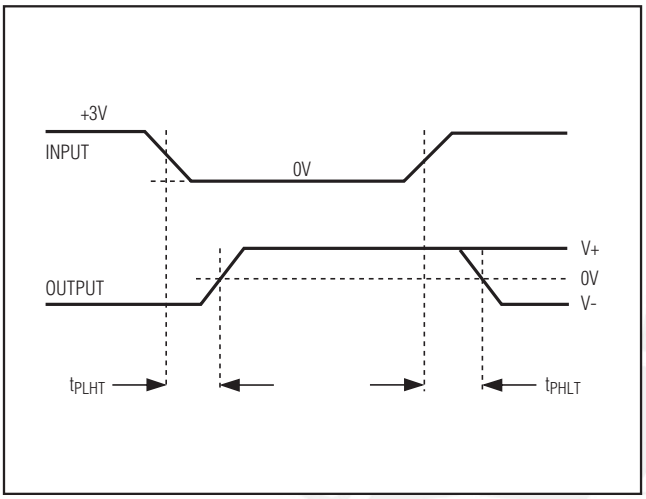


Figure 1. Transmitter Propagation-Delay Timing

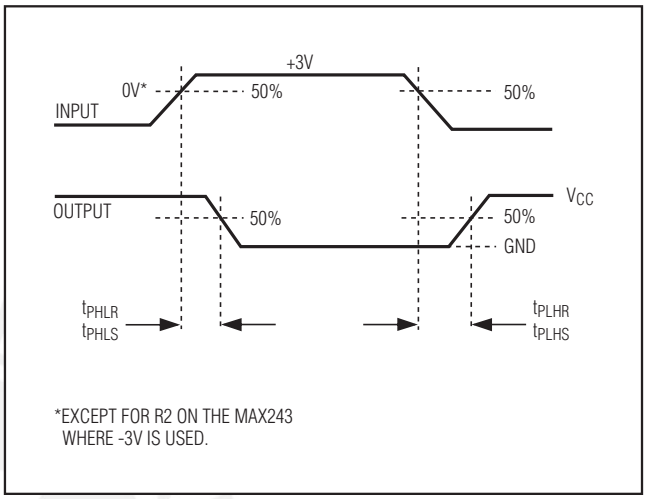


Figure 2. Receiver Propagation-Delay Timing

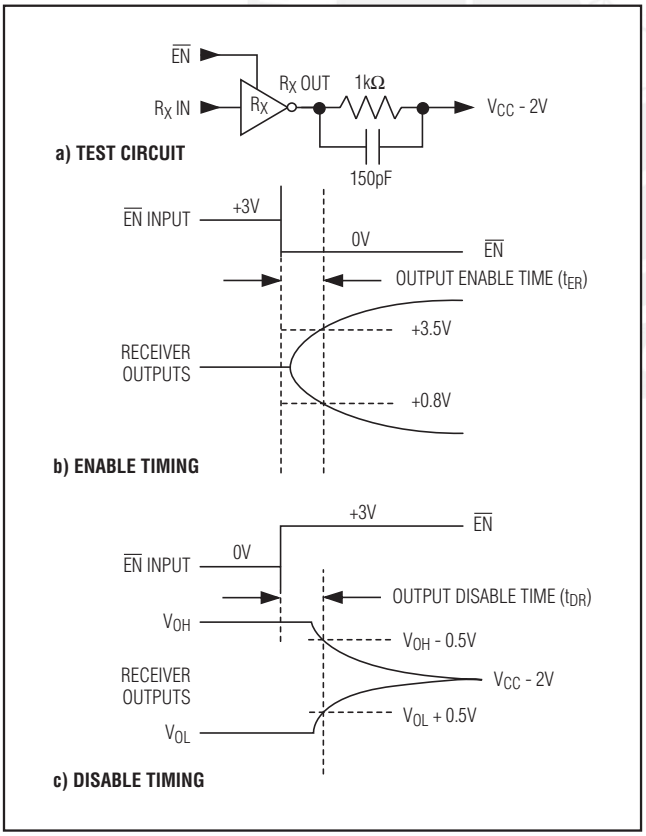


Figure 3. Receiver-Output Enable and Disable Timing

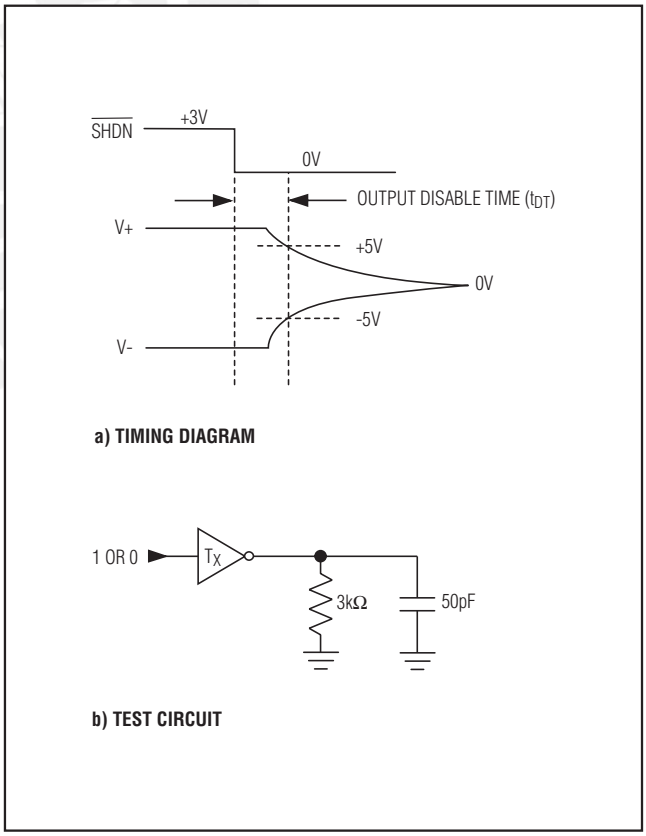


Figure 4. Transmitter-Output Disable Timing

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Control Pin Configuration Tables

Table 1a. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS	RECEIVERS
0	0	Normal Operation	All Active	All Active
0	1	Normal Operation	All Active	All High-Z
1	0	Shutdown	All High-Z	All Low-Power Receive Mode
1	1	Shutdown	All High-Z	All High-Z

Table 1b. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All Active	RA1–RA4 High-Z, RA5 Active	RB1–RB4 High-Z, RB5 Active
1	0	Shutdown	All High-Z	All High-Z	All Low-Power Receive Mode	All Low-Power Receive Mode
1	1	Shutdown	All High-Z	All High-Z	RA1–RA4 High-Z, RA5 Low-Power Receive Mode	RB1–RB4 High-Z, RB5 Low-Power Receive Mode

Table 1c. MAX246 Control Pin Configurations

$\overline{ENA}$	$\overline{ENB}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All High-Z	All Active	RB1–RB4 High-Z, RB5 Active
1	0	Shutdown	All High-Z	All Active	RA1–RA4 High-Z, RA5 Active	All Active
1	1	Shutdown	All High-Z	All High-Z	RA1–RA4 High-Z, RA5 Low-Power Receive Mode	RB1–RB4 High-Z, RA5 Low-Power Receive Mode

Table 1d. MAX247/MAX248/MAX249 Control Pin Configurations

$\overline{ENTA}$	$\overline{ENTB}$	$\overline{ENRA}$	$\overline{ENRB}$	OPERATION STATUS		TRANSMITTERS		RECEIVERS	
					MAX247	TA1–TA4	TB1–TB4	RA1–RA4	RB1–RB5
					MAX248	TA1–TA4	TB1–TB4	RA1–RA4	RB1–RB4
					MAX249	TA1–TA3	TB1–TB3	RA1–RA5	RB1–RB5
0	0	0	0	Normal Operation		All Active	All Active	All Active	All Active
0	0	0	1	Normal Operation		All Active	All Active	All Active	All High-Z, except RB5 stays active on MAX247
0	0	1	0	Normal Operation		All Active	All Active	All High-Z	All Active
0	0	1	1	Normal Operation		All Active	All Active	All High-Z	All High-Z, except RB5 stays active on MAX247
0	1	0	0	Normal Operation		All Active	All High-Z	All Active	All Active
0	1	0	1	Normal Operation		All Active	All High-Z	All Active	All High-Z, except RB5 stays active on MAX247
0	1	1	0	Normal Operation		All Active	All High-Z	All High-Z	All Active
0	1	1	1	Normal Operation		All Active	All High-Z	All High-Z	All High-Z, except RB5 stays active on MAX247
1	0	0	0	Normal Operation		All High-Z	All Active	All Active	All Active
1	0	0	1	Normal Operation		All High-Z	All Active	All Active	All High-Z, except RB5 stays active on MAX247
1	0	1	0	Normal Operation		All High-Z	All Active	All High-Z	All Active
1	0	1	1	Normal Operation		All High-Z	All Active	All High-Z	All High-Z, except RB5 stays active on MAX247
1	1	0	0	Shutdown		All High-Z	All High-Z	Low-Power Receive Mode	Low-Power Receive Mode
1	1	0	1	Shutdown		All High-Z	All High-Z	Low-Power Receive Mode	All High-Z, except RB5 stays active on MAX247
1	1	1	0	Shutdown		All High-Z	All High-Z	All High-Z	Low-Power Receive Mode
1	1	1	1	Shutdown		All High-Z	All High-Z	All High-Z	All High-Z, except RB5 stays active on MAX247

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Detailed Description

The MAX220–MAX249 contain four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, RS-232 receivers, and receiver and transmitter enable control inputs.

Dual Charge-Pump Voltage Converter

The MAX220–MAX249 have two internal charge-pumps that convert +5V to ±10V (unloaded) for RS-232 driver operation. The first converter uses capacitor C1 to double the +5V input to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to -10V on C4 at the V- output.

A small amount of power may be drawn from the +10V (V+) and -10V (V-) outputs to power external circuitry (see the *Typical Operating Characteristics* section), except on the MAX225 and MAX245–MAX247, where these pins are not available. V+ and V- are not regulated, so the output voltage drops with increasing load current. Do not load V+ and V- to a point that violates the minimum ±5V EIA/TIA-232E driver output voltage when sourcing current from V+ and V- to external circuitry.

When using the shutdown feature in the MAX222, MAX225, MAX230, MAX235, MAX236, MAX240, MAX241, and MAX245–MAX249, avoid using V+ and V- to power external circuitry. When these parts are shut down, V- falls to 0V, and V+ falls to +5V. For applications where a +10V external supply is applied to the V+ pin (instead of using the internal charge pump to generate +10V), the C1 capacitor must not be installed and the $\overline{SHDN}$ pin must be connected to V_{CC}. This is because V+ is internally connected to V_{CC} in shutdown mode.

RS-232 Drivers

The typical driver output voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver and V_{CC} = +5V. Output swing is guaranteed to meet the EIA/TIA-232E and V.28 specification, which calls for ±5V minimum driver output levels under worst-case conditions. These include a minimum 3kΩ load, V_{CC} = +4.5V, and maximum operating temperature. Unloaded driver output voltage ranges from (V+ -1.3V) to (V- +0.5V).

Input thresholds are both TTL and CMOS compatible. The inputs of unused drivers can be left unconnected since 400kΩ input pullup resistors to V_{CC} are built in (except for the MAX220). The pullup resistors force the outputs of unused drivers low because all drivers invert. The internal input pullup resistors typically source 12µA, except in shutdown mode where the pullups are disabled. Driver outputs turn off and enter a high-impedance state—where leakage current is typically microamperes (maximum 25µA)—when in shutdown mode, in three-state mode, or when device power is removed. Outputs can be driven to ±15V. The power-supply current typically drops to 8µA in shutdown mode. The MAX220 does not have pullup resistors to force the outputs of the unused drivers low. Connect unused inputs to GND or V_{CC}.

The MAX239 has a receiver three-state control line, and the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver three-state control line and a low-power shutdown control. Table 2 shows the effects of the shutdown control and receiver three-state control on the receiver outputs.

The receiver TTL/CMOS outputs are in a high-impedance, three-state mode whenever the three-state enable line is high (for the MAX225/MAX235/MAX236/MAX239–MAX241), and are also high-impedance whenever the shutdown control line is high.

When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1µA with the driver output pulled to ground. The driver output leakage remains less than 1µA, even if the transmitter output is backdriven between 0V and (V_{CC} + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1kΩ series impedance. The transmitter is also zener clamped to approximately V_{CC} + 6V, with a series impedance of 1kΩ.

The driver output slew rate is limited to less than 30V/µs as required by the EIA/TIA-232E and V.28 specifications. Typical slew rates are 24V/µs unloaded and 10V/µs loaded with 3Ω and 2500pF.

RS-232 Receivers

EIA/TIA-232E and V.28 specifications define a voltage level greater than 3V as a logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232E and V.28 levels.

The receiver inputs withstand an input overvoltage up to ±25V and provide input terminating resistors with

Table 2. Three-State Control of Receivers

PART	SHDN	$\overline{SHDN}$	EN	$\overline{EN(R)}$	RECEIVERS
MAX223	—	Low High High	X Low High	—	High Impedance Active High Impedance
MAX225	—	—	—	Low High	High Impedance Active
MAX235 MAX236 MAX240	Low Low High	—	—	Low High X	High Impedance Active High Impedance

+5V-Powered, Multichannel RS-232 Drivers/Receivers

nominal 5kΩ values. The receivers implement Type 1 interpretation of the fault conditions of V.28 and EIA/TIA-232E.

The receiver input hysteresis is typically 0.5V with a guaranteed minimum of 0.2V. This produces clear output transitions with slow-moving input signals, even with moderate amounts of noise and ringing. The receiver propagation delay is typically 600ns and is independent of input swing direction.

Low-Power Receive Mode

The low-power receive mode feature of the MAX223, MAX242, and MAX245–MAX249 puts the IC into shutdown mode but still allows it to receive information. This is important for applications where systems are periodically awakened to look for activity. Using low-power receive mode, the system can still receive a signal that will activate it on command and prepare it for communication at faster data rates. This operation conserves system power.

Negative Threshold—MAX243

The MAX243 is pin compatible with the MAX232A, differing only in that RS-232 cable fault protection is removed on one of the two receiver inputs. This means that control lines such as CTS and RTS can either be driven or left unconnected without interrupting communication. Different cables are not needed to interface with different pieces of equipment.

The input threshold of the receiver without cable fault protection is -0.8V rather than +1.4V. Its output goes positive only if the input is connected to a control line that is actively driven negative. If not driven, it defaults to the 0 or “OK to send” state. Normally, the MAX243’s other receiver (+1.4V threshold) is used for the data line (TD or RD), while the negative threshold receiver is connected to the control line (DTR, DTS, CTS, RTS, etc.).

Other members of the RS-232 family implement the optional cable fault protection as specified by EIA/TIA-232E specifications. This means a receiver output goes high whenever its input is driven negative, left unconnected, or shorted to ground. The high output tells the serial communications IC to stop sending data. To avoid this, the control lines must either be driven or connected with jumpers to an appropriate positive voltage level.

Shutdown—MAX222–MAX242

On the MAX222, MAX235, MAX236, MAX240, and MAX241, all receivers are disabled during shutdown. On the MAX223 and MAX242, two receivers continue to operate in a reduced power mode when the chip is in shutdown. Under these conditions, the propagation delay increases to about 2.5µs for a high-to-low input transition. When in shutdown, the receiver acts as a CMOS inverter with no hysteresis. The MAX223 and MAX242 also have a receiver output enable input ($\overline{EN}$ for the MAX242 and EN for the MAX223) that allows receiver output control independent of $\overline{SHDN}$ (SHDN for MAX241). With all other devices, $\overline{SHDN}$ (SHDN for MAX241) also disables the receiver outputs.

The MAX225 provides five transmitters and five receivers, while the MAX245 provides ten receivers and eight transmitters. Both devices have separate receiver and transmitter-enable controls. The charge pumps turn off and the devices shut down when a logic high is applied to the ENT input. In this state, the supply current drops to less than 25µA and the receivers continue to operate in a low-power receive mode. Driver outputs enter a high-impedance state (three-state mode). On the MAX225, all five receivers are controlled by the $\overline{ENR}$ input. On the MAX245, eight of the receiver outputs are controlled by the $\overline{ENR}$ input, while the remaining two receivers (RA5 and RB5) are always active. RA1–RA4 and RB1–RB4 are put in a three-state mode when $\overline{ENR}$ is a logic high.

Receiver and Transmitter Enable Control Inputs

The MAX225 and MAX245–MAX249 feature transmitter and receiver enable controls.

The receivers have three modes of operation: full-speed receive (normal active), three-state (disabled), and low-power receive (enabled receivers continue to function at lower data rates). The receiver enable inputs control the full-speed receive and three-state modes. The transmitters have two modes of operation: full-speed transmit (normal active) and three-state (disabled). The transmitter enable inputs also control the shutdown mode. The device enters shutdown mode when all transmitters are disabled. Enabled receivers function in the low-power receive mode when in shutdown.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Tables 1a–1d define the control states. The MAX244 has no control pins and is not included in these tables.

The MAX246 has ten receivers and eight drivers with two control pins, each controlling one side of the device. A logic high at the A-side control input ($\overline{ENA}$) causes the four A-side receivers and drivers to go into a three-state mode. Similarly, the B-side control input ($\overline{ENB}$) causes the four B-side drivers and receivers to go into a three-state mode. As in the MAX245, one A-side and one B-side receiver (RA5 and RB5) remain active at all times. The entire device is put into shutdown mode when both the A and B sides are disabled ($\overline{ENA}$ = $\overline{ENB}$ = +5V).

The MAX247 provides nine receivers and eight drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control four receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs each control four drivers. The ninth receiver (RB5) is always active. The device enters shutdown mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$.

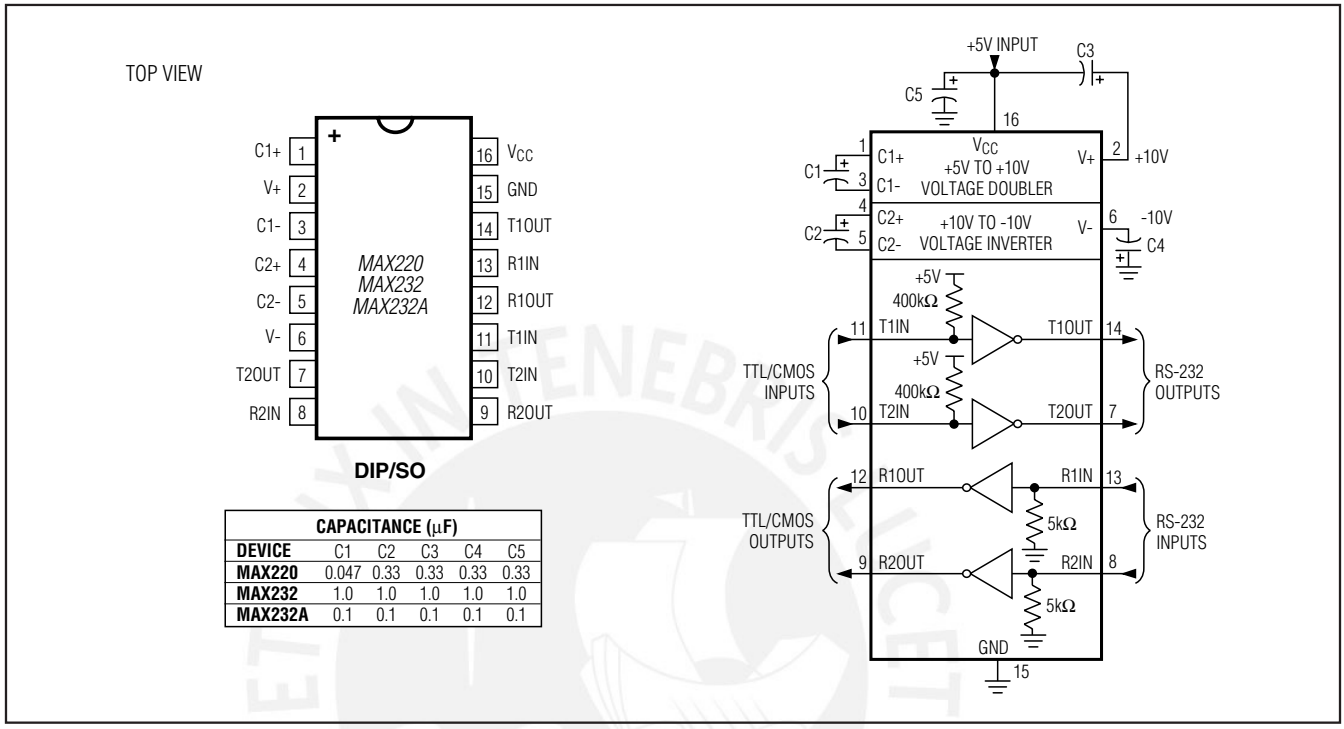
The MAX248 provides eight receivers and eight drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control four receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs control four drivers each. This part does not have an always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$.

The MAX249 provides ten receivers and six drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control five receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs control three drivers each. There is no always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$. In shutdown mode, active receivers operate in a low-power receive mode at data rates up to 20kb/s.

Applications Information

Figures 5 through 25 show pin configurations and typical operating circuits. In applications that are sensitive to power-supply noise, V_{CC} should be decoupled to ground with a capacitor of the same value as C1 and C2 connected as close as possible to the device.

+5V-Powered, Multichannel RS-232 Drivers/Receivers



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

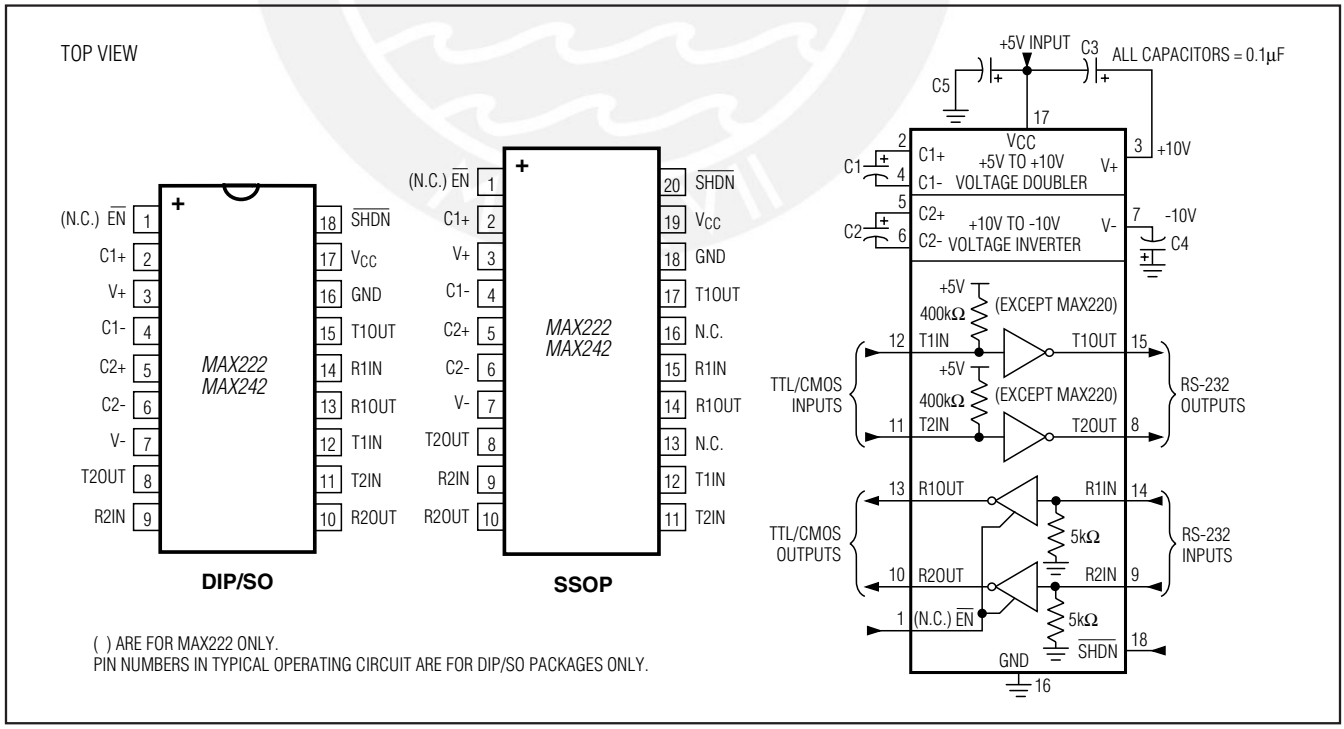
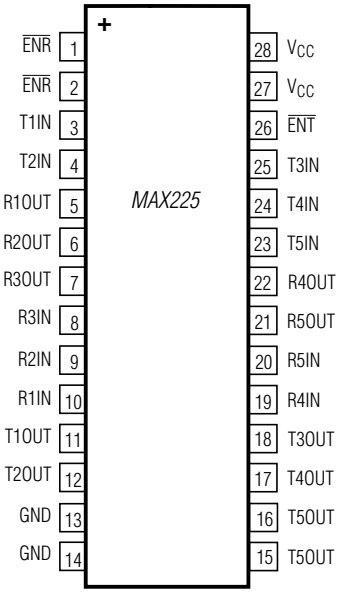


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

TOP VIEW



SO

MAX225 FUNCTIONAL DESCRIPTION

5 RECEIVERS
5 TRANSMITTERS
2 CONTROL PINS
1 RECEIVER ENABLE ($\overline{ENR}$)
1 TRANSMITTER ENABLE ($\overline{ENT}$)

PINS ($\overline{ENR}$, GND, V_{CC}, T5OUT) ARE INTERNALLY CONNECTED.
CONNECT EITHER OR BOTH EXTERNALLY. T5OUT IS A SINGLE DRIVER.

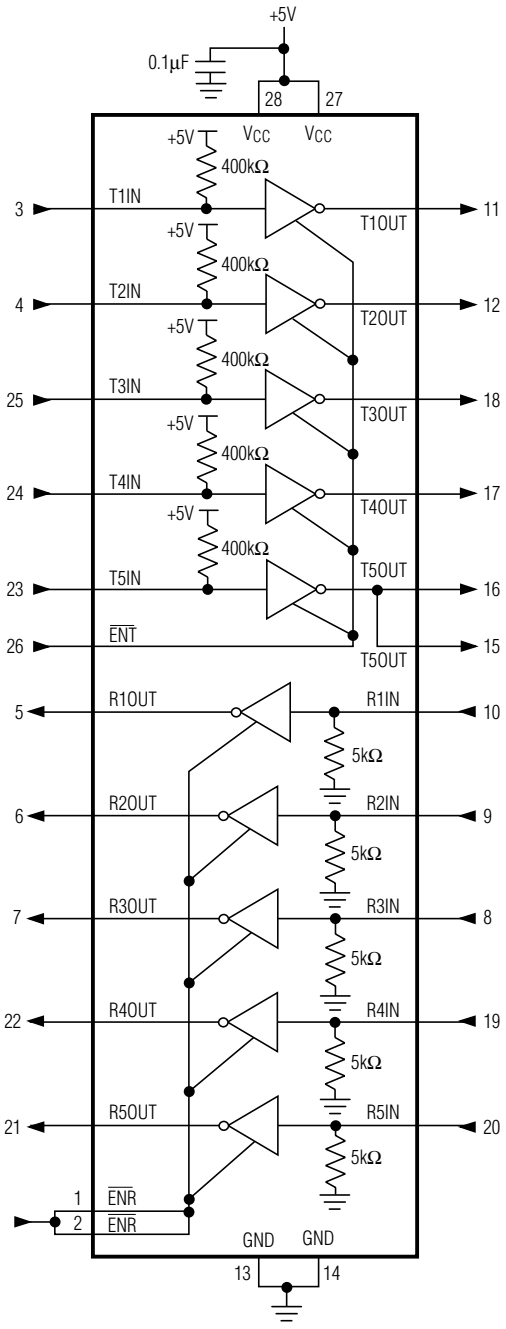


Figure 7. MAX225 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

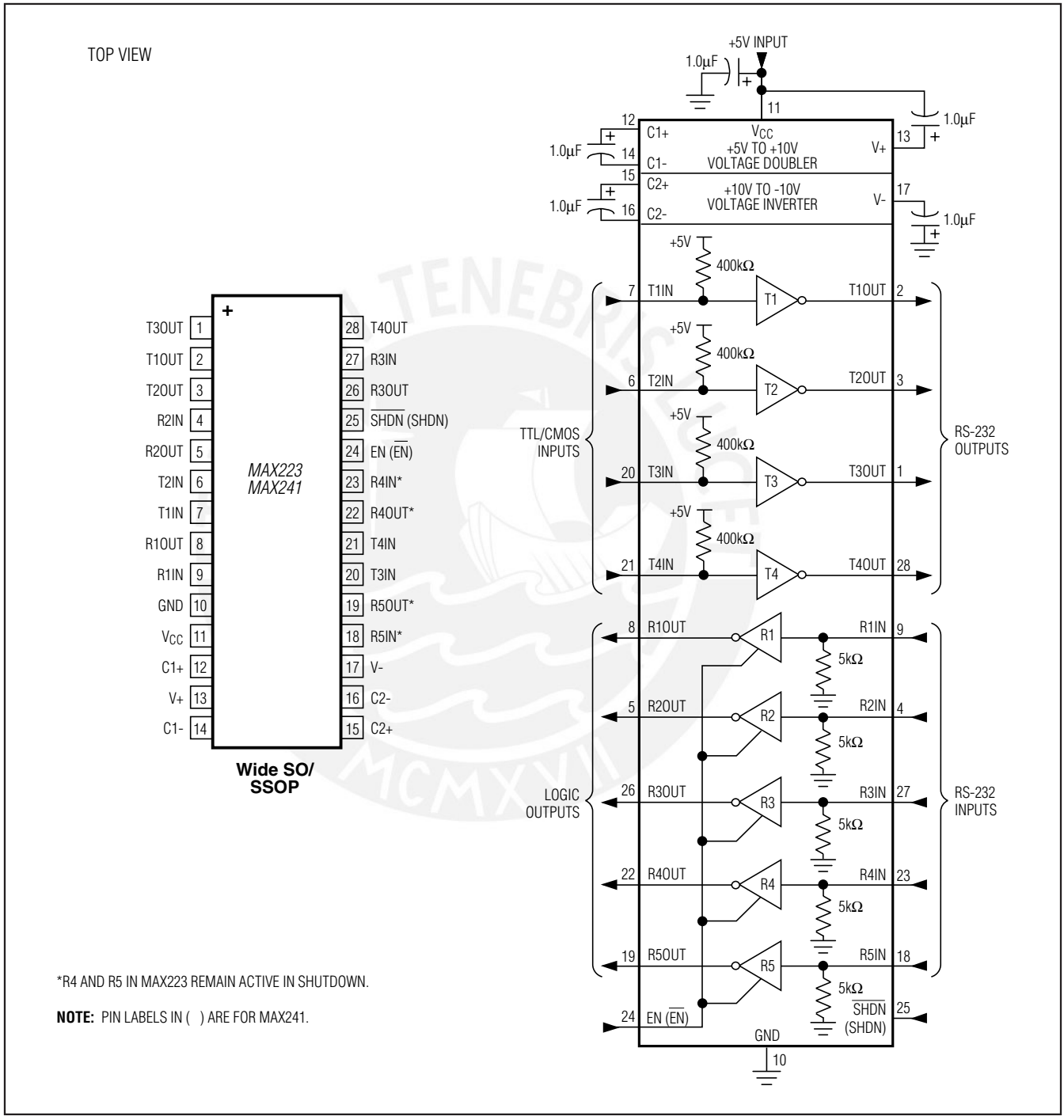


Figure 8. MAX223/MAX241 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

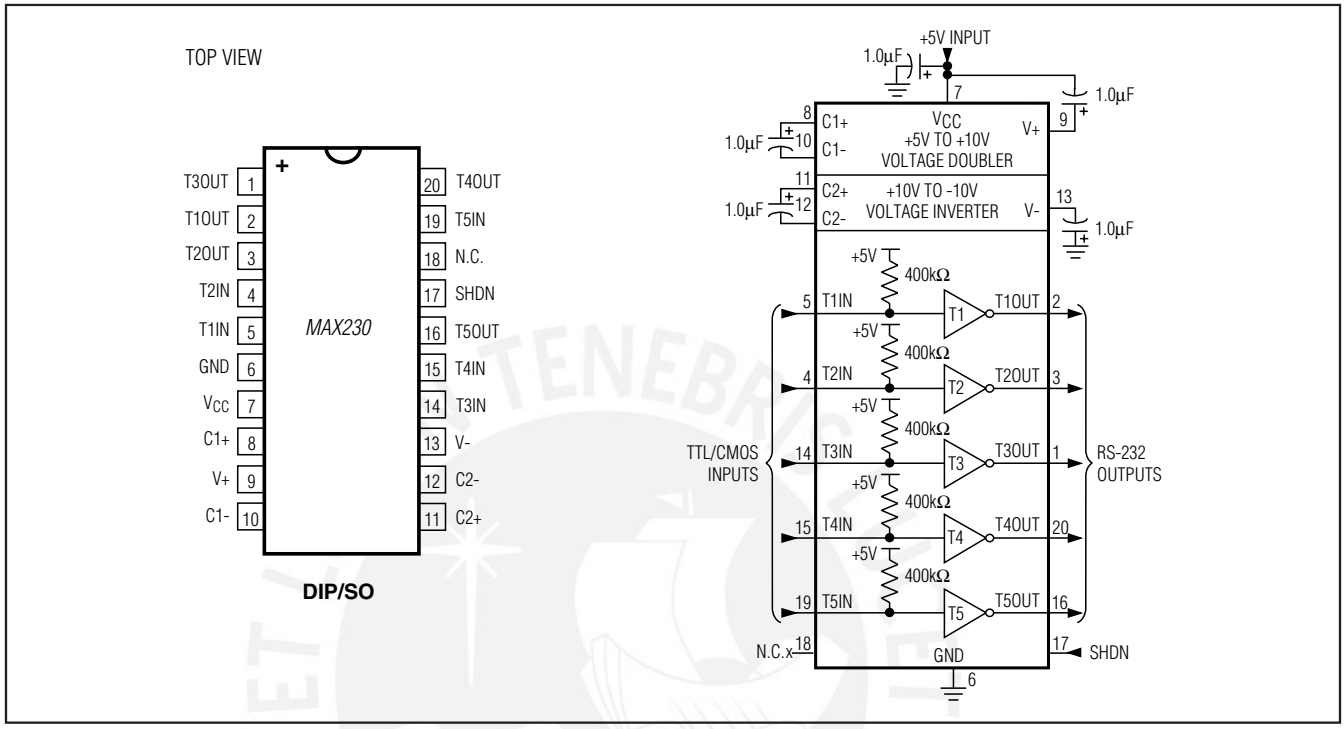


Figure 9. MAX230 Pin Configuration and Typical Operating Circuit

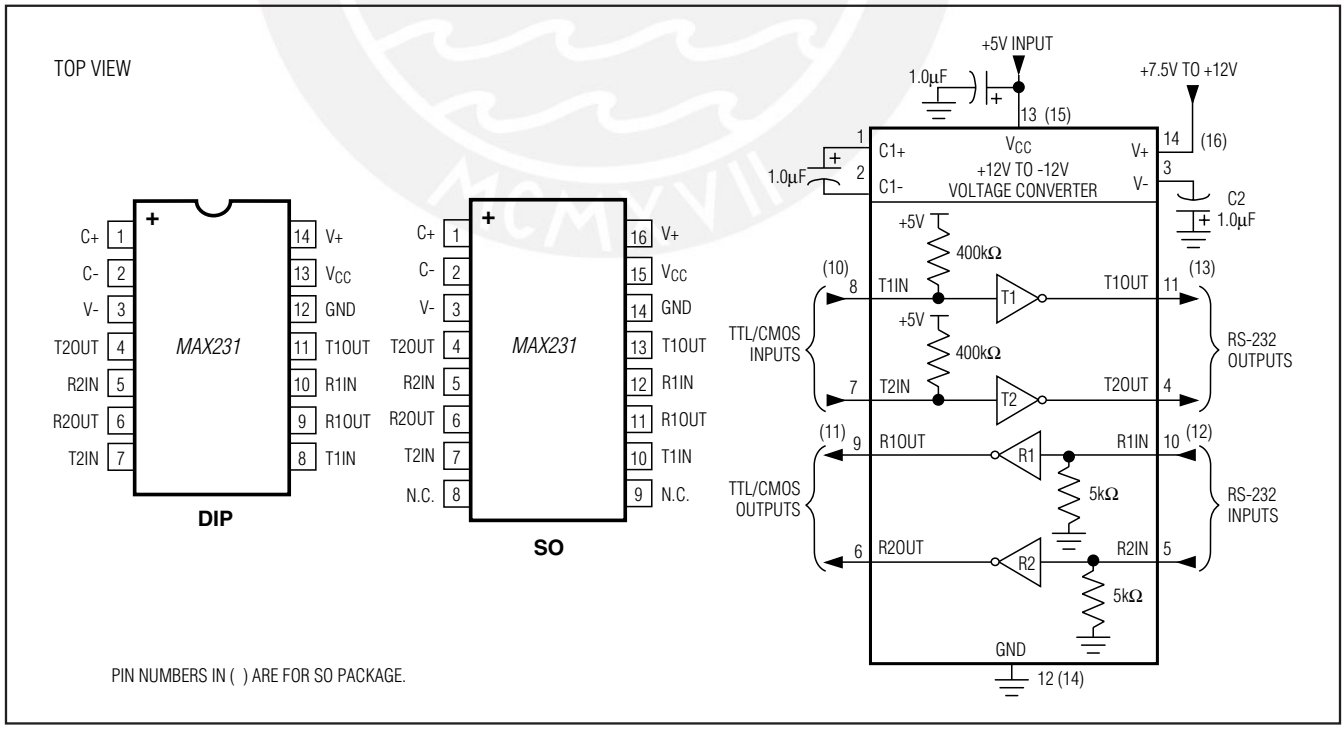


Figure 10. MAX231 Pin Configurations and Typical Operating Circuit

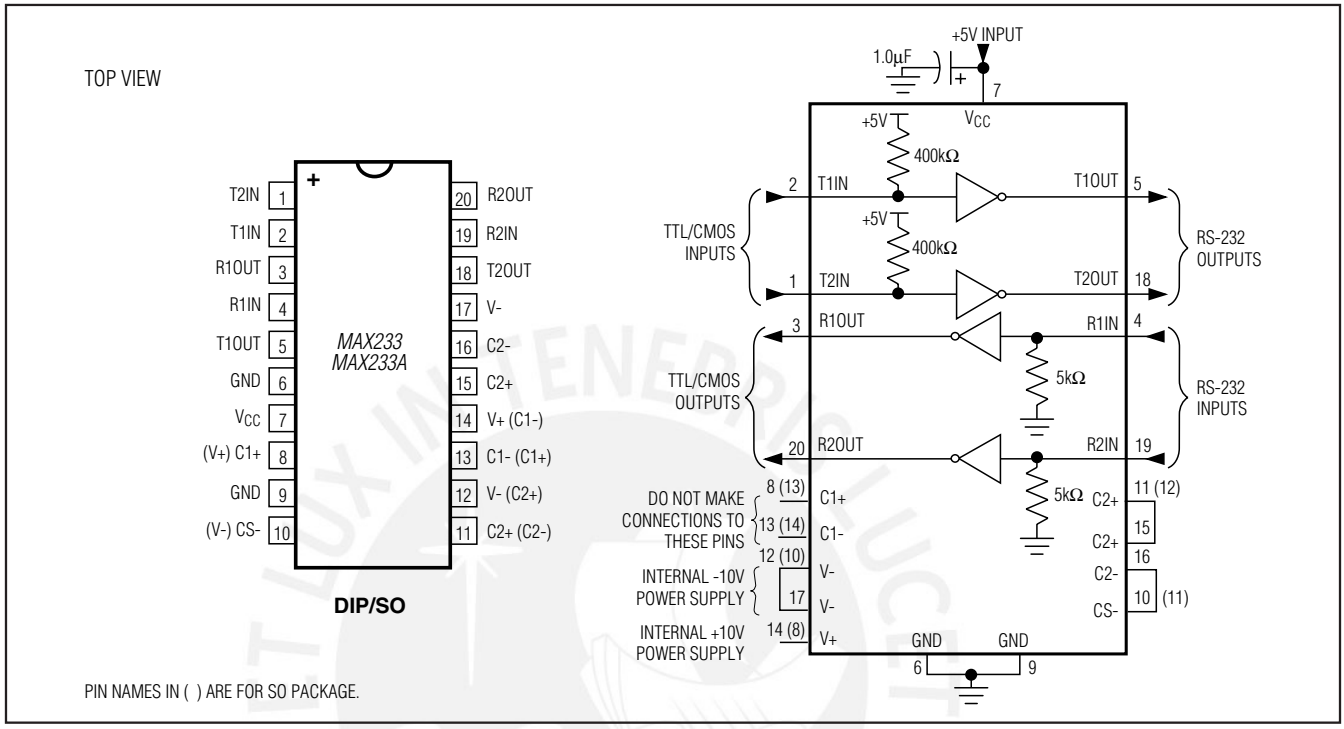


Figure 11. MAX233/MAX233A Pin Configuration and Typical Operating Circuit

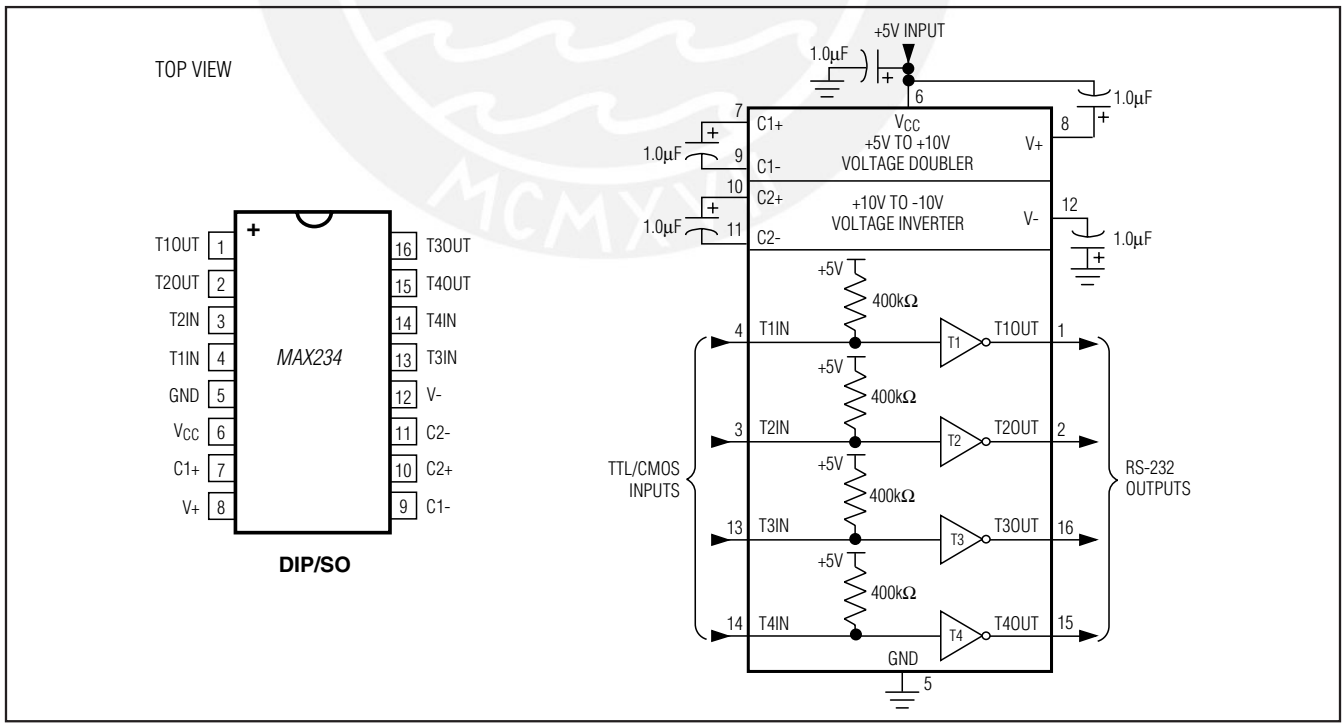


Figure 12. MAX234 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

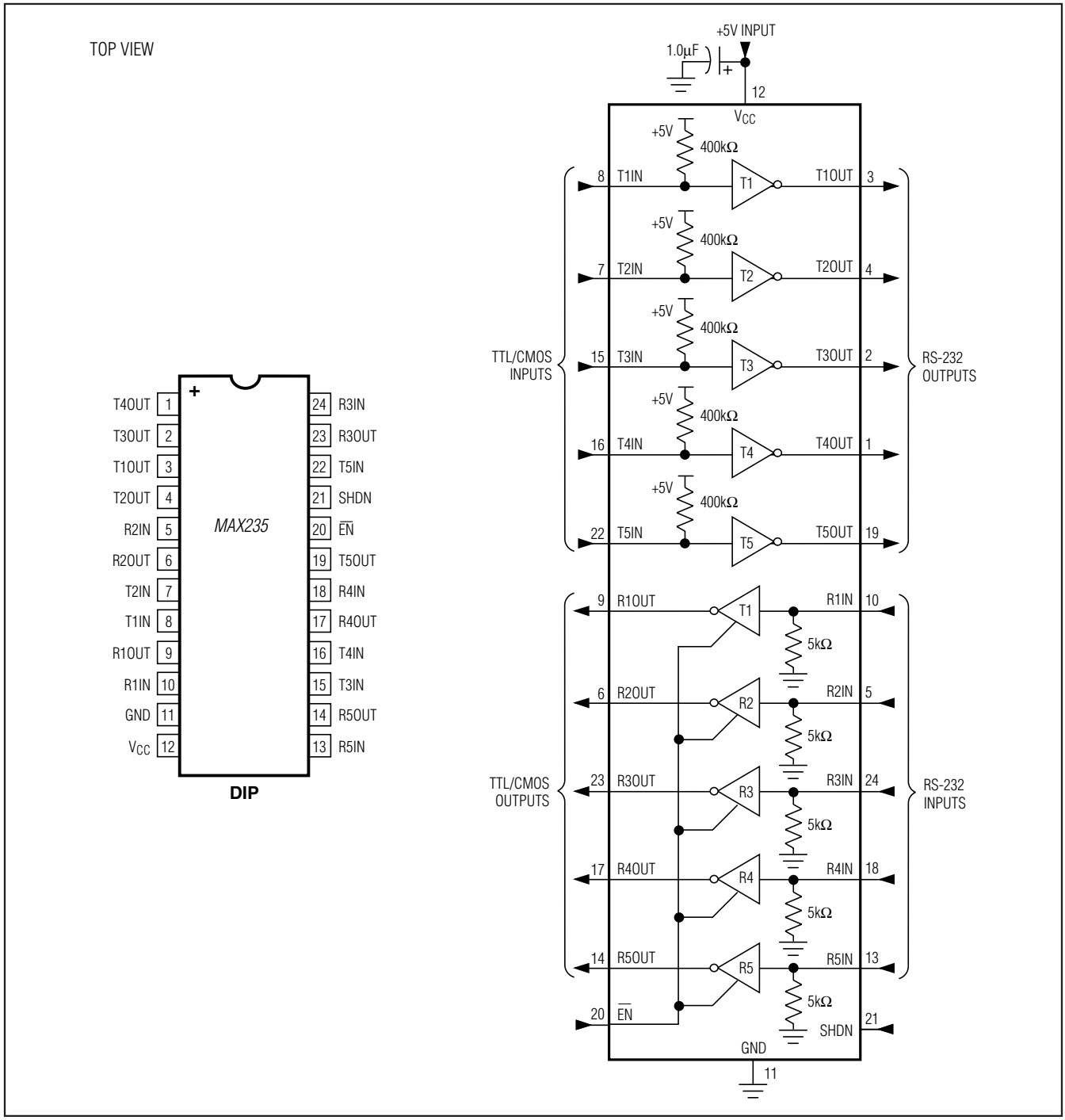


Figure 13. MAX235 Pin Configuration and Typical Operating Circuit

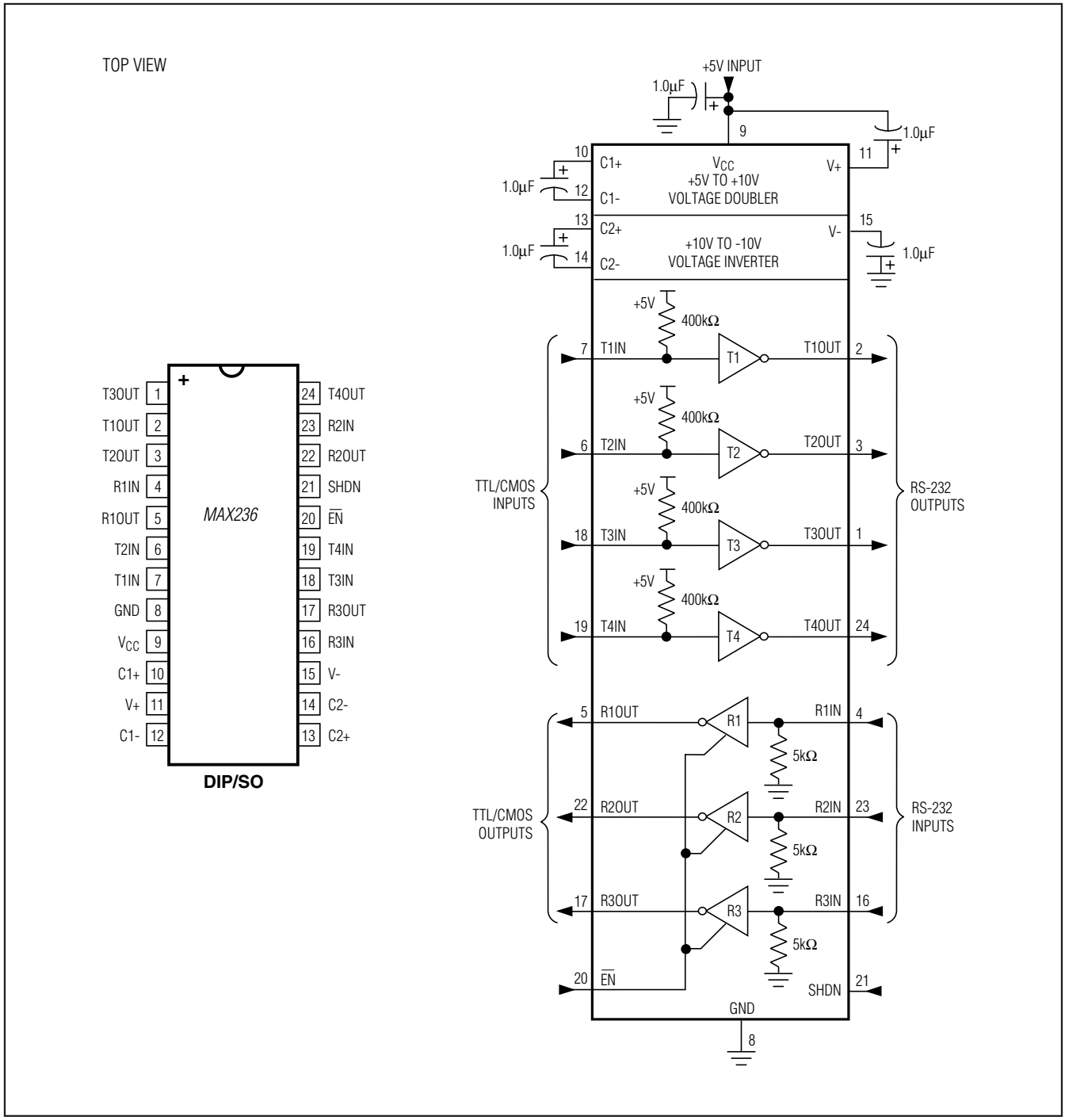


Figure 14. MAX236 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

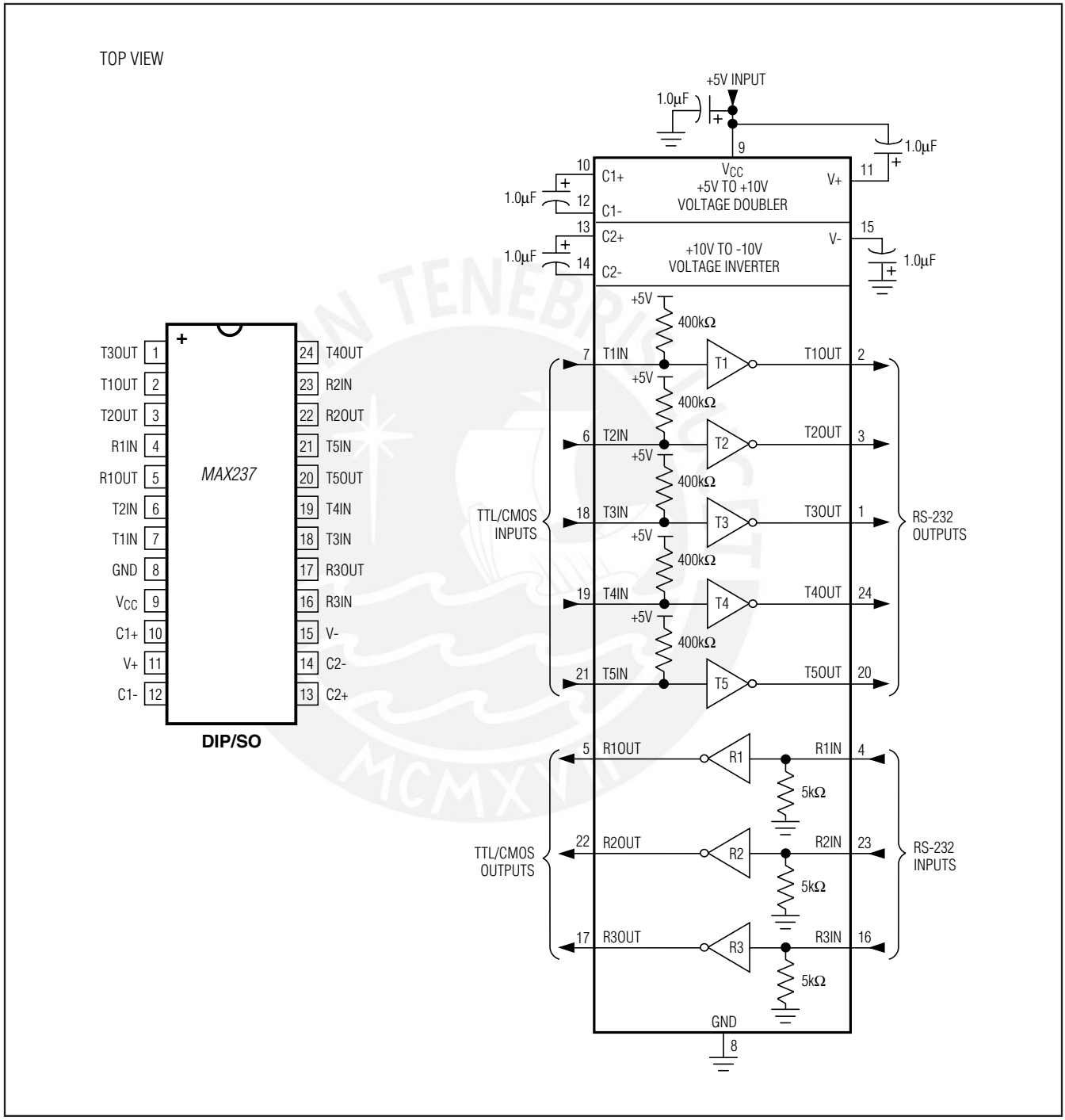


Figure 15. MAX237 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

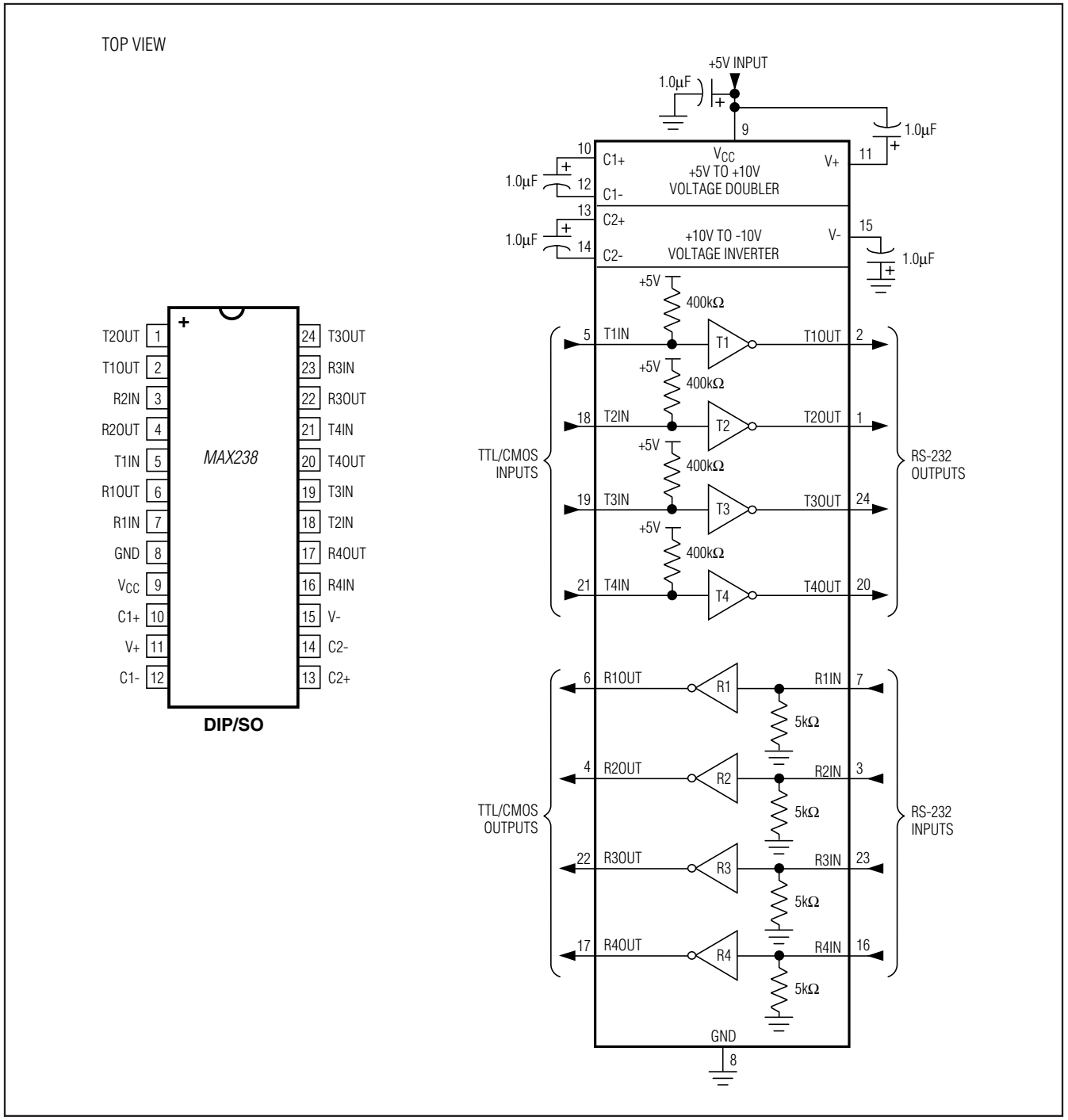
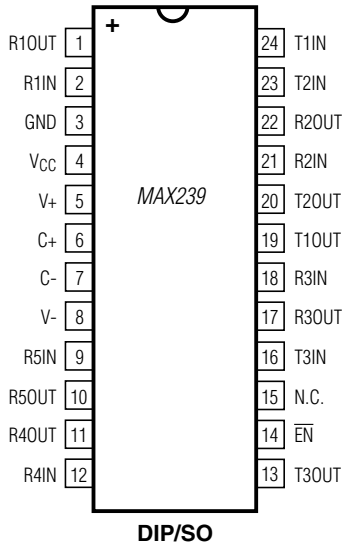


Figure 16. MAX238 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers



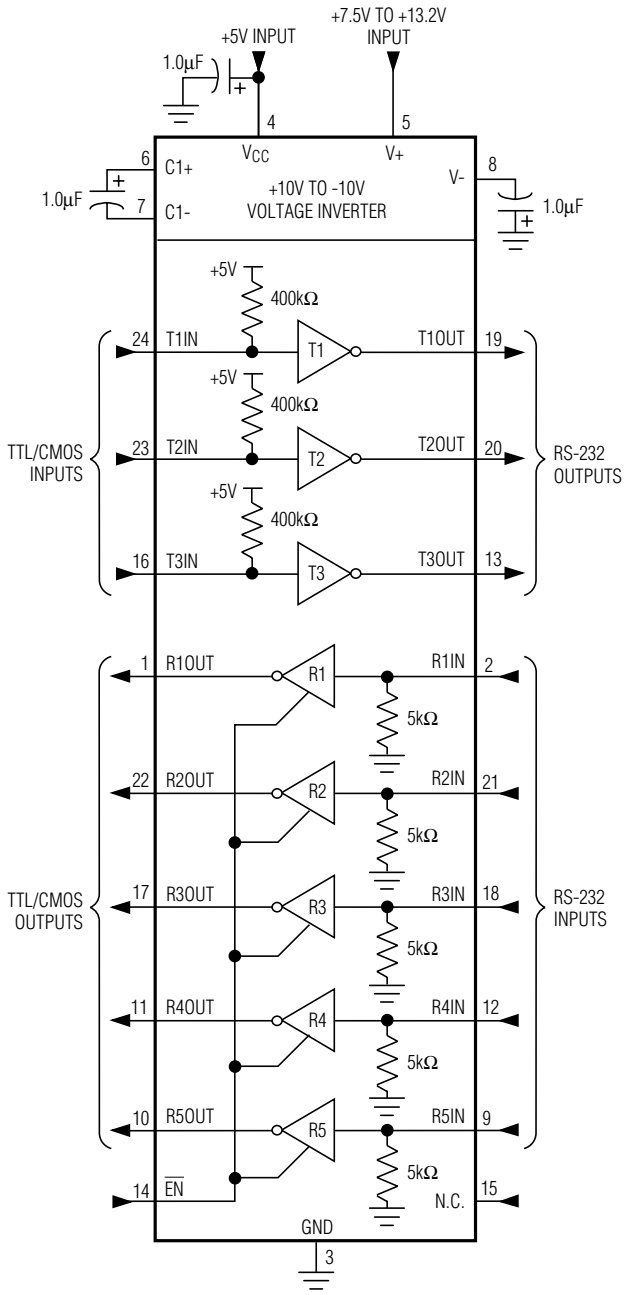


Figure 17. MAX239 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

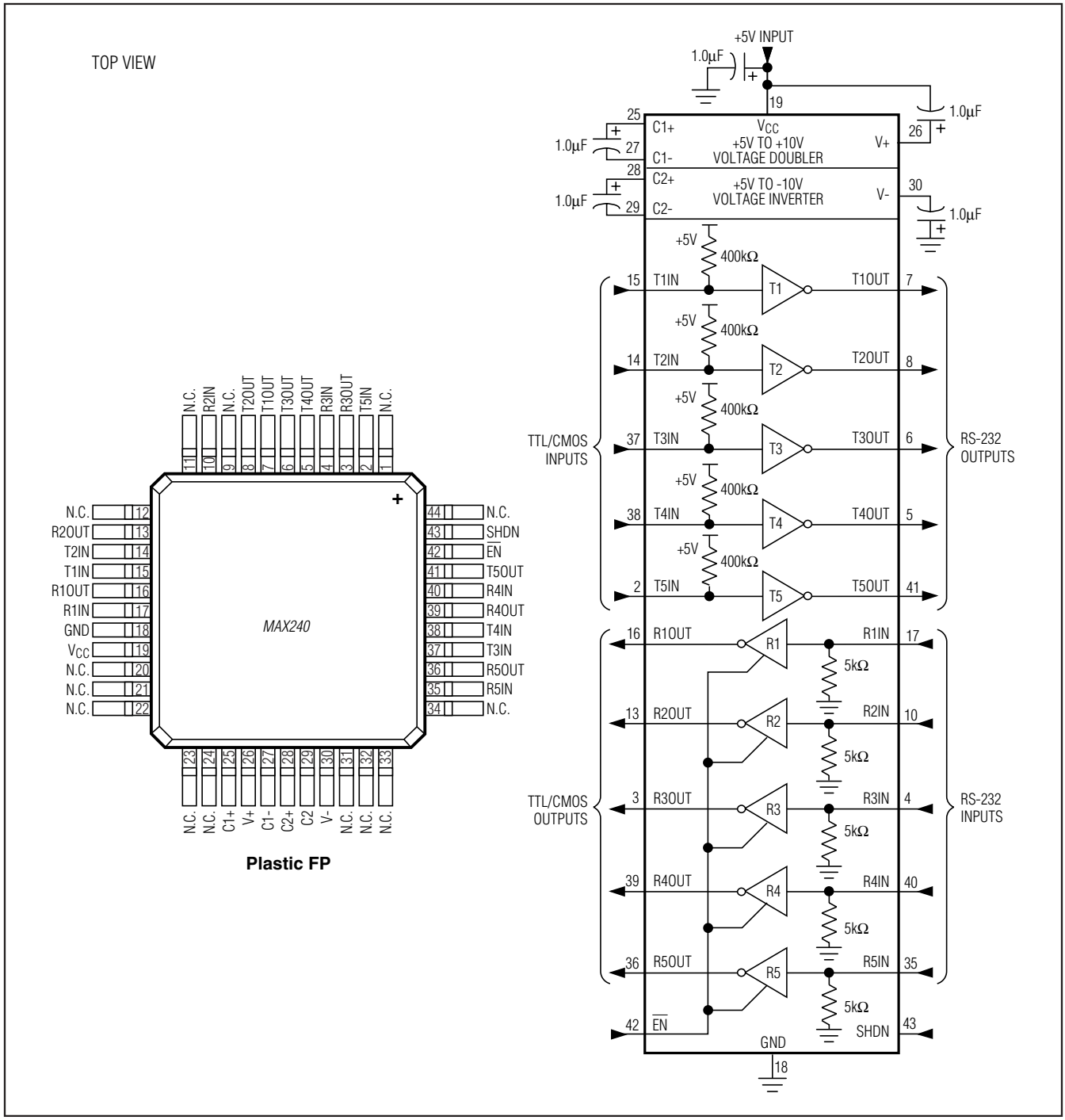
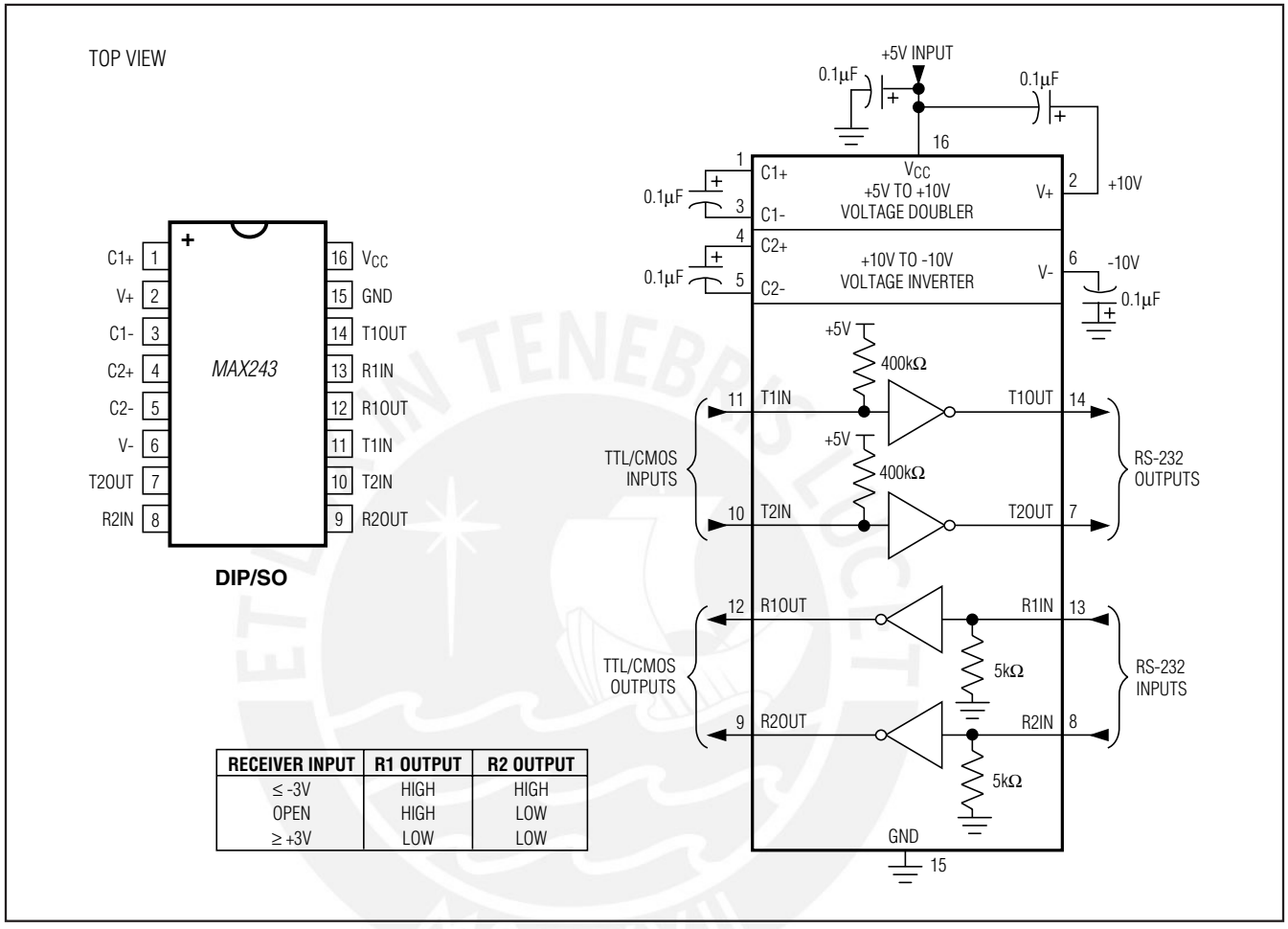


Figure 18. MAX240 Pin Configuration and Typical Operating Circuit

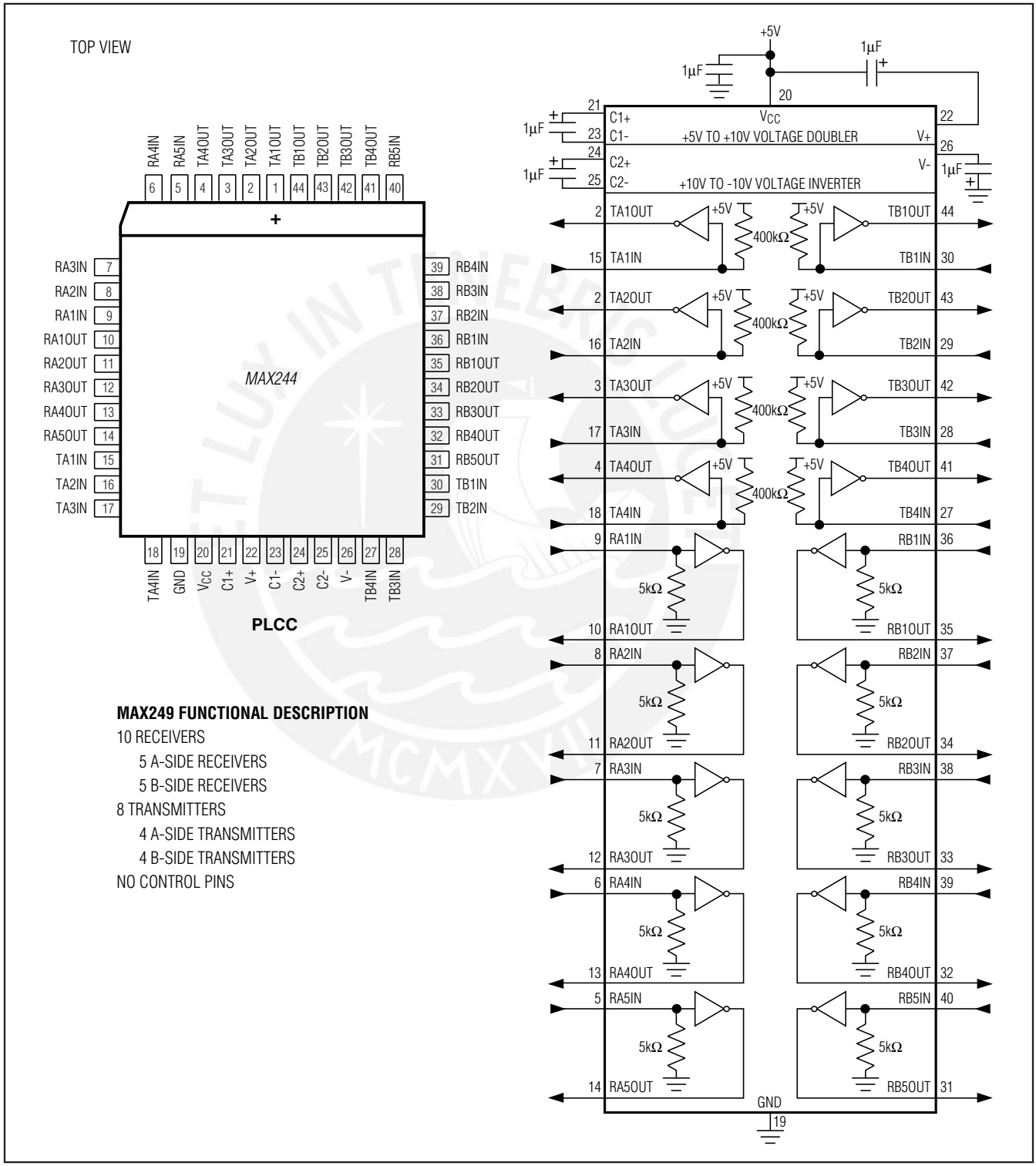
+5V-Powered, Multichannel RS-232 Drivers/Receivers



RECEIVER INPUT	R1 OUTPUT	R2 OUTPUT
≤ -3V	HIGH	HIGH
OPEN	HIGH	LOW
≥ +3V	LOW	LOW

Figure 19. MAX243 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers



MAX249 FUNCTIONAL DESCRIPTION

10 RECEIVERS
- 5 A-SIDE RECEIVERS
- 5 B-SIDE RECEIVERS

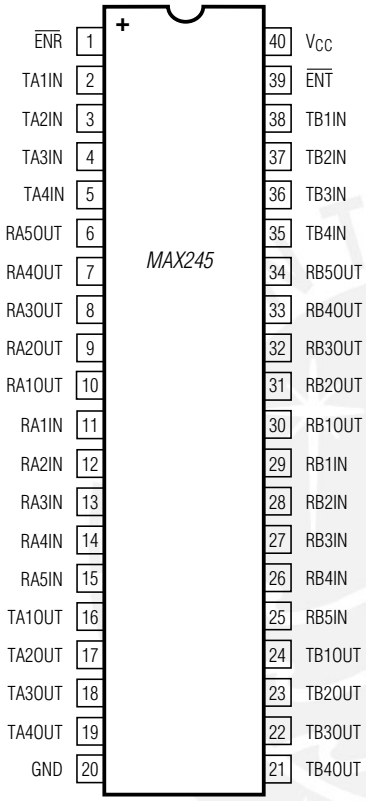
8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

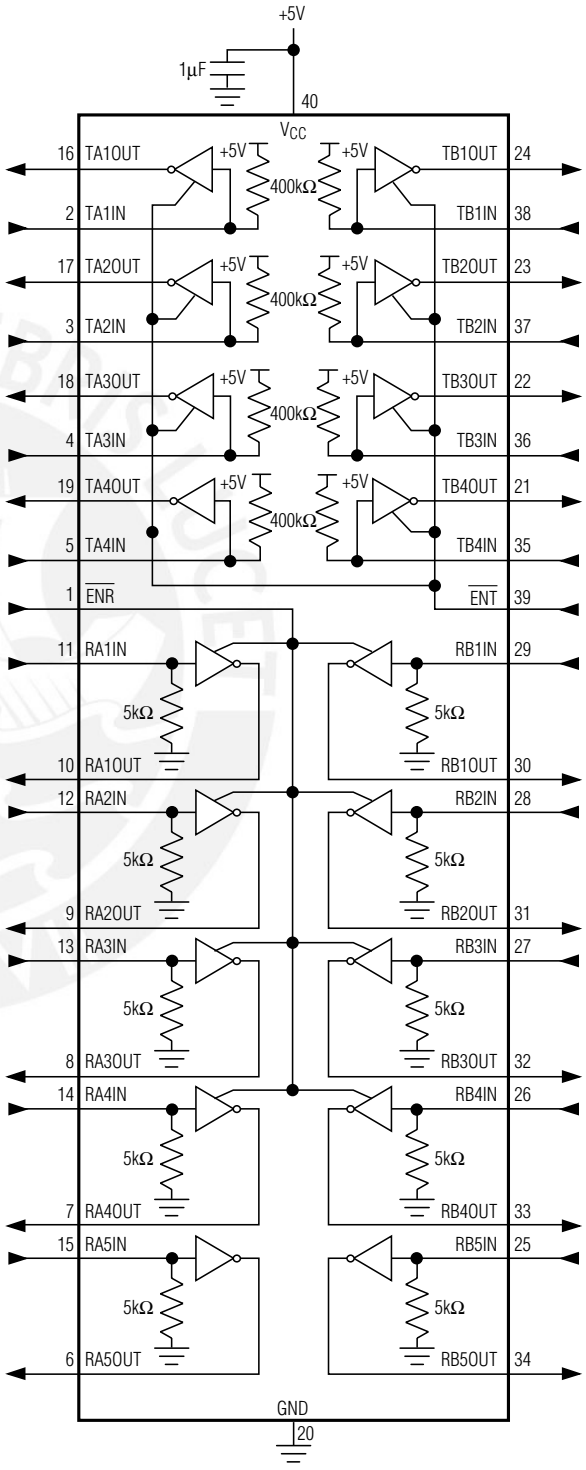
NO CONTROL PINS

Figure 20. MAX244 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

TOP VIEW





MAX245 FUNCTIONAL DESCRIPTION

10 RECEIVERS
- 5 A-SIDE RECEIVERS (R_{A5} ALWAYS ACTIVE)
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

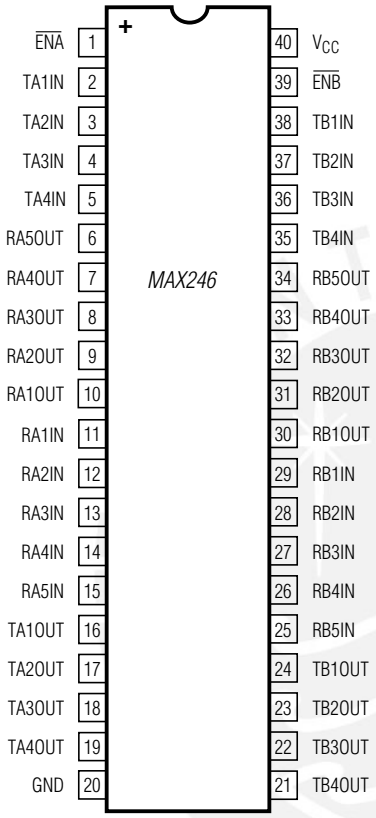
8 TRANSMITTTERS
- 4 A-SIDE TRANSMITTERS

2 CONTROL PINS
- 1 RECEIVER ENABLE ($\overline{ENR}$)
- 1 TRANSMITTER ENABLE ($\overline{ENT}$)

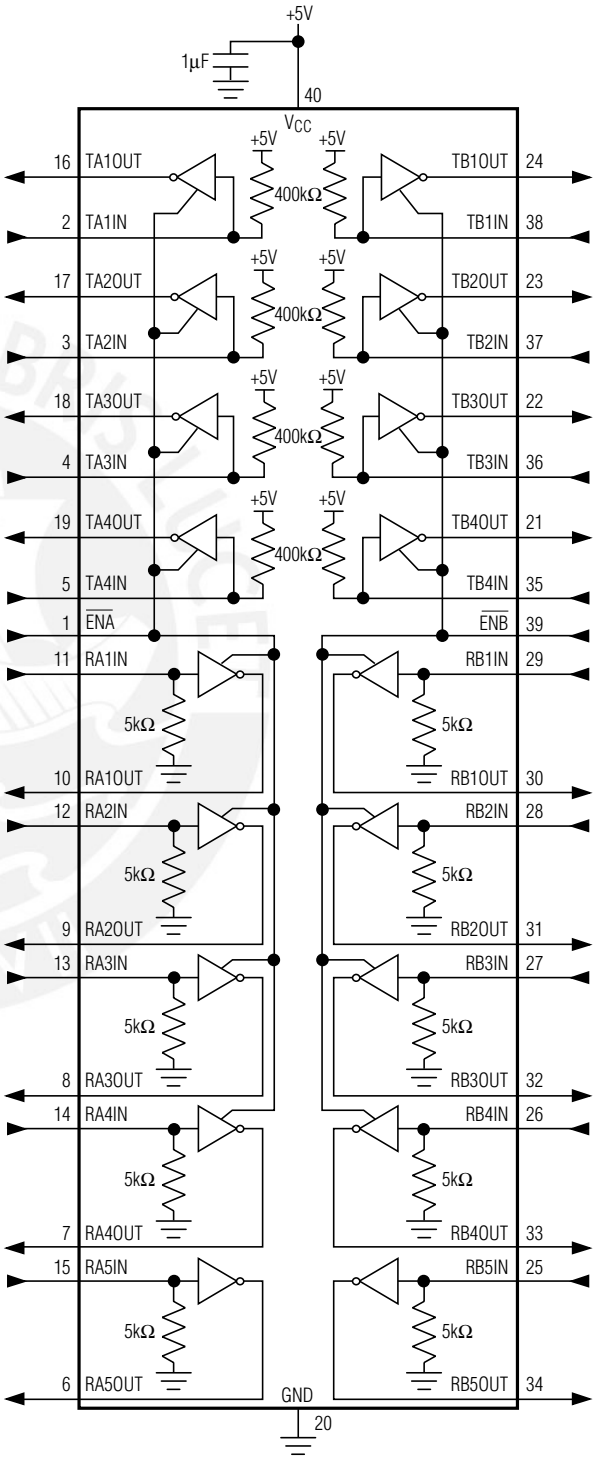
Figure 21. MAX245 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

TOP VIEW



DIP



MAX246 FUNCTIONAL DESCRIPTION

10 RECEIVERS
- 5 A-SIDE RECEIVERS (R_{A5} ALWAYS ACTIVE)
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

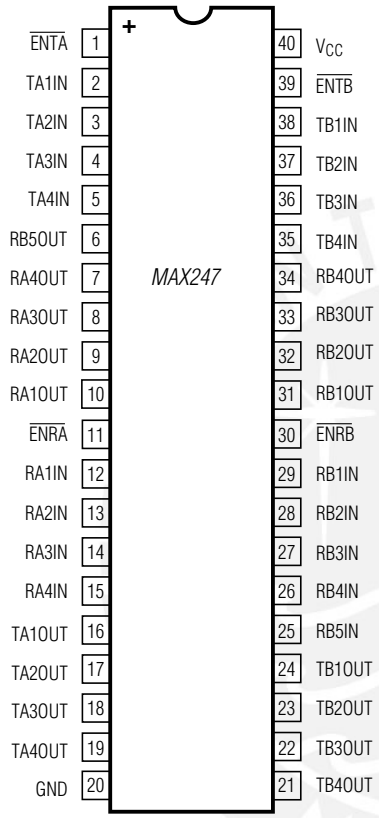
8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

2 CONTROL PINS
- ENABLE A-SIDE ($\overline{ENA}$)
- ENABLE B-SIDE ($\overline{ENB}$)

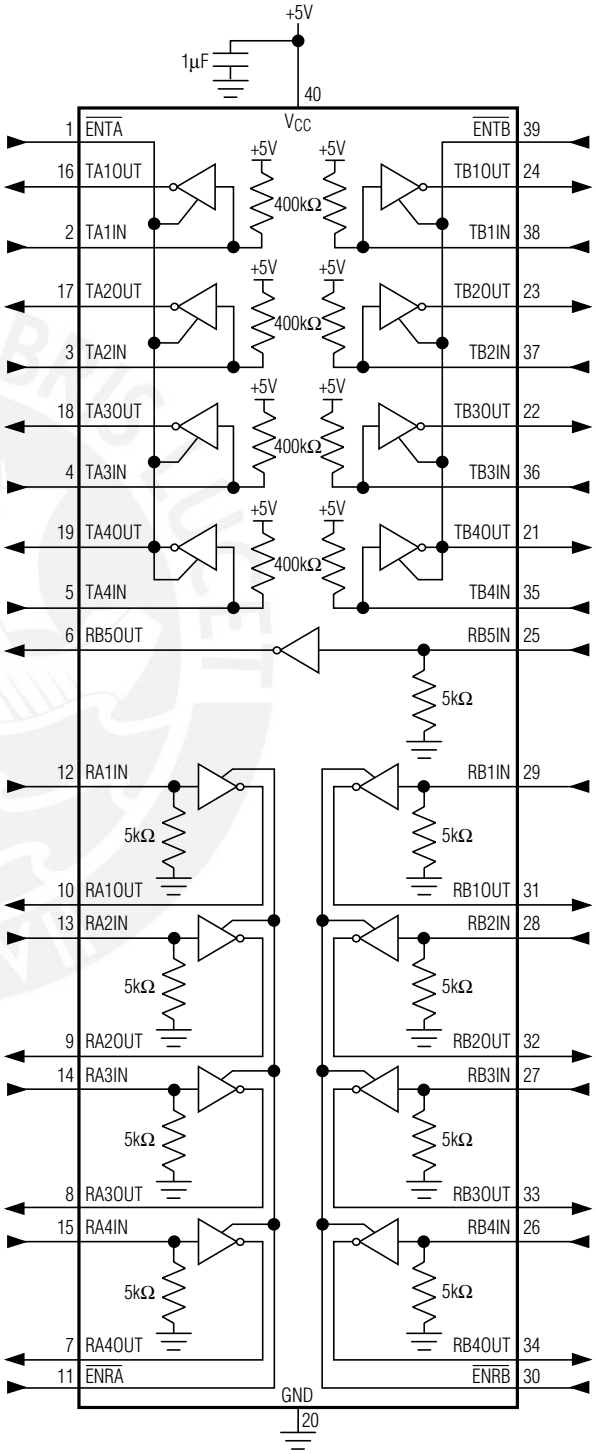
Figure 22. MAX246 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

TOP VIEW



DIP



MAX247 FUNCTIONAL DESCRIPTION

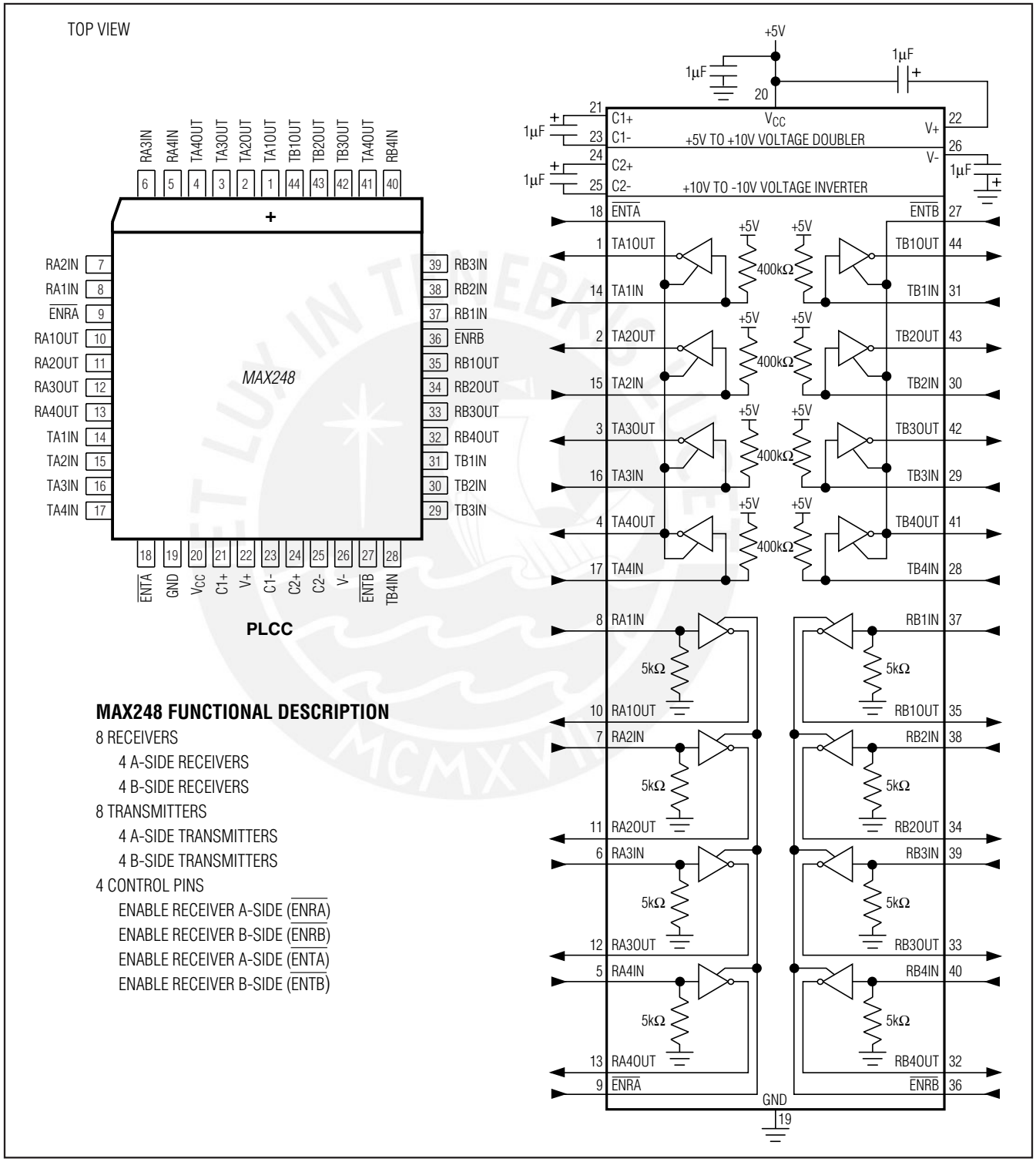
9 RECEIVERS
- 4 A-SIDE RECEIVERS
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

4 CONTROL PINS
- ENABLE RECEIVER A-SIDE ($\overline{ENRA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENRB}$)
- ENABLE RECEIVER A-SIDE ($\overline{ENTA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENTB}$)

Figure 23. MAX247 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers



MAX248 FUNCTIONAL DESCRIPTION

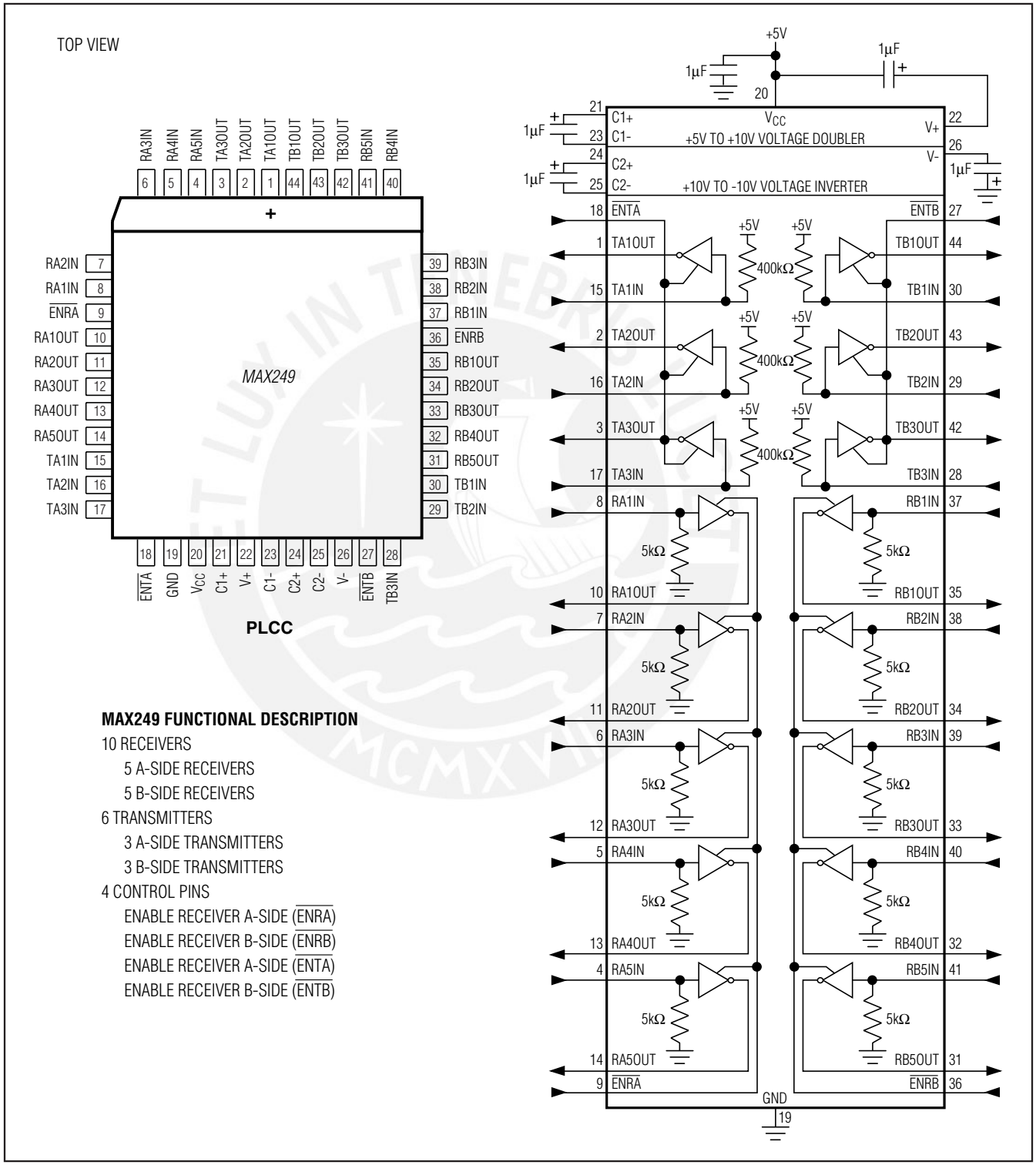
8 RECEIVERS
- 4 A-SIDE RECEIVERS
- 4 B-SIDE RECEIVERS

8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

4 CONTROL PINS
- ENABLE RECEIVER A-SIDE ($\overline{ENRA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENRB}$)
- ENABLE RECEIVER A-SIDE ($\overline{ENTA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENTB}$)

Figure 24. MAX248 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers



MAX249 FUNCTIONAL DESCRIPTION

10 RECEIVERS
- 5 A-SIDE RECEIVERS
- 5 B-SIDE RECEIVERS

6 TRANSMITTERS
- 3 A-SIDE TRANSMITTERS
- 3 B-SIDE TRANSMITTERS

4 CONTROL PINS
- ENABLE RECEIVER A-SIDE ($\overline{ENRA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENRB}$)
- ENABLE RECEIVER A-SIDE ($\overline{ENTA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENTB}$)

Figure 25. MAX249 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Ordering Information (continued)

PART	TEMP RANGE	PIN-PACKAGE
MAX222CPN+	0°C to +70°C	18 Plastic DIP
MAX222CWN+	0°C to +70°C	18 Wide SO
MAX222C/D	0°C to +70°C	Dice*
MAX222EPN+	-40°C to +85°C	18 Plastic DIP
MAX222EWN+	-40°C to +85°C	18 Wide SO
MAX222EJN	-40°C to +85°C	18 CERDIP
MAX222MJN	-55°C to +125°C	18 CERDIP
MAX223CAI+	0°C to +70°C	28 SSOP
MAX223CWI+	0°C to +70°C	28 Wide SO
MAX223C/D	0°C to +70°C	Dice*
MAX223EAI+	-40°C to +85°C	28 SSOP
MAX223EWI+	-40°C to +85°C	28 Wide SO
MAX225CWI+	0°C to +70°C	28 Wide SO
MAX225EWI+	-40°C to +85°C	28 Wide SO
MAX230CPP+	0°C to +70°C	20 Plastic DIP
MAX230CWP+	0°C to +70°C	20 Wide SO
MAX230C/D	0°C to +70°C	Dice*
MAX230EPP+	-40°C to +85°C	20 Plastic DIP
MAX230EWP+	-40°C to +85°C	20 Wide SO
MAX230EJP	-40°C to +85°C	20 CERDIP
MAX230MJP	-55°C to +125°C	20 CERDIP
MAX231CPD+	0°C to +70°C	14 Plastic DIP
MAX231CWE+	0°C to +70°C	16 Wide SO
MAX231CJD	0°C to +70°C	14 CERDIP
MAX231C/D	0°C to +70°C	Dice*
MAX231EPD+	-40°C to +85°C	14 Plastic DIP
MAX231EWE+	-40°C to +85°C	16 Wide SO
MAX231EJD	-40°C to +85°C	14 CERDIP
MAX231MJD	-55°C to +125°C	14 CERDIP
MAX232CPE+	0°C to +70°C	16 Plastic DIP
MAX232CSE+	0°C to +70°C	16 Narrow SO
MAX232CWE+	0°C to +70°C	16 Wide SO
MAX232C/D	0°C to +70°C	Dice*
MAX232EPE+	-40°C to +85°C	16 Plastic DIP
MAX232ESE+	-40°C to +85°C	16 Narrow SO
MAX232EWE+	-40°C to +85°C	16 Wide SO
MAX232EJE	-40°C to +85°C	16 CERDIP
MAX232MJE	-55°C to +125°C	16 CERDIP
MAX232MLP+	-55°C to +125°C	20 LCC
MAX232ACPE+	0°C to +70°C	16 Plastic DIP
MAX232ACSE+	0°C to +70°C	16 Narrow SO
MAX232ACWE+	0°C to +70°C	16 Wide SO
MAX232AC/D	0°C to +70°C	Dice*
MAX232AEPE+	-40°C to +85°C	16 Plastic DIP
MAX232AESE+	-40°C to +85°C	16 Narrow SO
MAX232AEWE+	-40°C to +85°C	16 Wide SO
MAX232AEJE	-40°C to +85°C	16 CERDIP
MAX232AMJE	-55°C to +125°C	16 CERDIP
MAX232AMLP+	-55°C to +125°C	20 LCC
MAX233CPP+	0°C to +70°C	20 Plastic DIP
MAX233EPP+	-40°C to +85°C	20 Plastic DIP
MAX233ACPP+	0°C to +70°C	20 Plastic DIP
MAX233ACWP+	0°C to +70°C	20 Wide SO
MAX233AEPP+	-40°C to +85°C	20 Plastic DIP
MAX233AEWP+	-40°C to +85°C	20 Wide SO
MAX234CPE+	0°C to +70°C	16 Plastic DIP
MAX234CWE+	0°C to +70°C	16 Wide SO
MAX234C/D	0°C to +70°C	Dice*
MAX234EPE+	-40°C to +85°C	16 Plastic DIP
MAX234EWE+	-40°C to +85°C	16 Wide SO
MAX234EJE	-40°C to +85°C	16 CERDIP
MAX234MJE	-55°C to +125°C	16 CERDIP
MAX235CPG+	0°C to +70°C	24 Wide Plastic DIP
MAX235EPG+	-40°C to +85°C	24 Wide Plastic DIP
MAX235EDG	-40°C to +85°C	24 Ceramic SB
MAX235MDG	-55°C to +125°C	24 Ceramic SB
MAX236CNG+	0°C to +70°C	24 Narrow Plastic DIP
MAX236CWG+	0°C to +70°C	24 Wide SO
MAX236C/D	0°C to +70°C	Dice*
MAX236ENG+	-40°C to +85°C	24 Narrow Plastic DIP
MAX236EWG+	-40°C to +85°C	24 Wide SO
MAX236ERG	-40°C to +85°C	24 Narrow CERDIP
MAX236MRG	-55°C to +125°C	24 Narrow CERDIP
MAX237CNG+	0°C to +70°C	24 Narrow Plastic DIP
MAX237CWG+	0°C to +70°C	24 Wide SO
MAX237C/D	0°C to +70°C	Dice*
MAX237ENG+	-40°C to +85°C	24 Narrow Plastic DIP
MAX237EWG+	-40°C to +85°C	24 Wide SO
MAX237ERG	-40°C to +85°C	24 Narrow CERDIP
MAX237MRG	-55°C to +125°C	24 Narrow CERDIP
MAX238CNG+	0°C to +70°C	24 Narrow Plastic DIP
MAX238CWG+	0°C to +70°C	24 Wide SO
MAX238C/D	0°C to +70°C	Dice*

+Denotes a lead(Pb)-free/RoHS-compliant package.
**Contact factory for dice specifications.*

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Ordering Information (continued)

PART	TEMP RANGE	PIN-PACKAGE
MAX238ENG+	-40°C to +85°C	24 Narrow Plastic DIP
MAX238EWG+	-40°C to +85°C	24 Wide SO
MAX238ERG	-40°C to +85°C	24 Narrow CERDIP
MAX238MRG	-55°C to +125°C	24 Narrow CERDIP
MAX239CNG+	0°C to +70°C	24 Narrow Plastic DIP
MAX239CWG+	0°C to +70°C	24 Wide SO
MAX239C/D	0°C to +70°C	Dice*
MAX239ENG+	-40°C to +85°C	24 Narrow Plastic DIP
MAX239EWG+	-40°C to +85°C	24 Wide SO
MAX239ERG	-40°C to +85°C	24 Narrow CERDIP
MAX239MRG	-55°C to +125°C	24 Narrow CERDIP
MAX240CMH+	0°C to +70°C	44 Plastic FP
MAX240C/D	0°C to +70°C	Dice*
MAX241CAI+	0°C to +70°C	28 SSOP
MAX241CWI+	0°C to +70°C	28 Wide SO
MAX241C/D	0°C to +70°C	Dice*
MAX241EAI+	-40°C to +85°C	28 SSOP
MAX241EWI+	-40°C to +85°C	28 Wide SO
MAX242CAP+	0°C to +70°C	20 SSOP
MAX242CPN+	0°C to +70°C	18 Plastic DIP
MAX242CWN+	0°C to +70°C	18 Wide SO
MAX242C/D	0°C to +70°C	Dice*
MAX242EPN+	-40°C to +85°C	18 Plastic DIP
MAX242EWN+	-40°C to +85°C	18 Wide SO
MAX242EJN	-40°C to +85°C	18 CERDIP
MAX242MJN	-55°C to +125°C	18 CERDIP

PART	TEMP RANGE	PIN-PACKAGE
MAX243CPE+	0°C to +70°C	16 Plastic DIP
MAX243CSE+	0°C to +70°C	16 Narrow SO
MAX243CWE+	0°C to +70°C	16 Wide SO
MAX243C/D	0°C to +70°C	Dice*
MAX243EPE+	-40°C to +85°C	16 Plastic DIP
MAX243ESE+	-40°C to +85°C	16 Narrow SO
MAX243EWE+	-40°C to +85°C	16 Wide SO
MAX243EJE	-40°C to +85°C	16 CERDIP
MAX243MJE	-55°C to +125°C	16 CERDIP
MAX244CQH+	0°C to +70°C	44 PLCC
MAX244C/D	0°C to +70°C	Dice*
MAX244EQH+	-40°C to +85°C	44 PLCC
MAX245CPL+	0°C to +70°C	40 Plastic DIP
MAX245C/D	0°C to +70°C	Dice*
MAX245EPL+	-40°C to +85°C	40 Plastic DIP
MAX246CPL+	0°C to +70°C	40 Plastic DIP
MAX246C/D	0°C to +70°C	Dice*
MAX246EPL+	-40°C to +85°C	40 Plastic DIP
MAX247CPL+	0°C to +70°C	40 Plastic DIP
MAX247C/D	0°C to +70°C	Dice*
MAX247EPL+	-40°C to +85°C	40 Plastic DIP
MAX248CQH+	0°C to +70°C	44 PLCC
MAX248C/D	0°C to +70°C	Dice*
MAX248EQH+	-40°C to +85°C	44 PLCC
MAX249CQH+	0°C to +70°C	44 PLCC
MAX249EQH+	-40°C to +85°C	44 PLCC

+Denotes a lead(Pb)-free/RoHS-compliant package.
**Contact factory for dice specifications.*

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Package Information

For the latest package outline information and land patterns, go to www.maxim-ic.com/packages. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
14 PDIP	P14+3	21-0043	—
16 PDIP	P16+1		
16 PDIP	P16+2		
16 PDIP	P16+3		
18 PDIP	P18+5		
20 PDIP	P20+3		
20 PDIP	P20M+1		
24 PDIP	N24+3		
24 PDIP	P24M+1	21-0044	
28 PDIP	P28+2		
40 PDIP	P40+1		
40 PDIP	P40M+2		
14 CERDIP	J14-3	21-0045	
16 CERDIP	J16-3		
18 CERDIP	J18-2		
20 CERDIP	J20-2		
24 CERDIP	R24-4		
16 SO(N)	S16+3	21-0041	90-0097
16 SO(N)	S16+5		
16 SO(W)	W16+1	21-0042	90-0107
16 SO(W)	W16+2		
16 SO(W)	W16+3		
18 SO(W)	W18+1		90-0181
20 SO(W)	W20+3		90-0108
20 SO(W)	W20M+1		
24 SO(W)	W24+2		90-0182
28 SO(W)	W28+1		90-0109
28 SO(W)	W28+2		
28 SO(W)	W28M+1		
20 LCC	L20+3	21-0658	90-0177
20 SSOP	A20+1	21-0056	90-0094
24 SSOP	A24+2		90-0110
28 SSOP	A28+1		90-0095
16 TSSOP	U16+1		90-0117
16 FPCK	F16-3	21-0013	—
44 MQFP	M44+5	21-0826	90-0169
44 PLCC	Q44+1	21-0049	90-0236
44 PLCC	Q44+2		

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
15	1/06	Added part information to the lead temperature in the *Absolute Maximum Ratings* sections	2, 5, 8
16	7/10	Changed multiple packages to lead-free versions; updated/added notes 3, 4, 5, 7, and 8 to the *Electrical Characteristics* table; removed incorrect subscripting from all pin names in the *Electrical Characteristics* table and *Pin Configurations*	1, 2–9, 17–36

19-0122; Rev 8; 10/03

MAXIM

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

General Description

The MAX481, MAX483, MAX485, MAX487–MAX491, and MAX1487 are low-power transceivers for RS-485 and RS-422 communication. Each part contains one driver and one receiver. The MAX483, MAX487, MAX488, and MAX489 feature reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, thus allowing error-free data transmission up to 250kbps. The driver slew rates of the MAX481, MAX485, MAX490, MAX491, and MAX1487 are not limited, allowing them to transmit up to 2.5Mbps.

These transceivers draw between 120µA and 500µA of supply current when unloaded or fully loaded with disabled drivers. Additionally, the MAX481, MAX483, and MAX487 have a low-current shutdown mode in which they consume only 0.1µA. All parts operate from a single 5V supply.

Drivers are short-circuit current limited and are protected against excessive power dissipation by thermal shutdown circuitry that places the driver outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

The MAX487 and MAX1487 feature quarter-unit-load receiver input impedance, allowing up to 128 MAX487/MAX1487 transceivers on the bus. Full-duplex communications are obtained using the MAX488–MAX491, while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are designed for half-duplex applications.

Applications

Low-Power RS-485 Transceivers

Low-Power RS-422 Transceivers

Level Translators

Transceivers for EMI-Sensitive Applications

Industrial-Control Local Area Networks

Next Generation Device Features

- **For Fault-Tolerant Applications**
  MAX3430: ±80V Fault-Protected, Fail-Safe, 1/4 Unit Load, +3.3V, RS-485 Transceiver
  MAX3440E–MAX3444E: ±15kV ESD-Protected, ±60V Fault-Protected, 10Mbps, Fail-Safe, RS-485/J1708 Transceivers
- **For Space-Constrained Applications**
  MAX3460–MAX3464: +5V, Fail-Safe, 20Mbps, Profibus RS-485/RS-422 Transceivers
  MAX3362: +3.3V, High-Speed, RS-485/RS-422 Transceiver in a SOT23 Package
  MAX3280E–MAX3284E: ±15kV ESD-Protected, 52Mbps, +3V to +5.5V, SOT23, RS-485/RS-422, True Fail-Safe Receivers
  MAX3293/MAX3294/MAX3295: 20Mbps, +3.3V, SOT23, RS-855/RS-422 Transmitters
- **For Multiple Transceiver Applications**
  MAX3030E–MAX3033E: ±15kV ESD-Protected, +3.3V, Quad RS-422 Transmitters
- **For Fail-Safe Applications**
  MAX3080–MAX3089: Fail-Safe, High-Speed (10Mbps), Slew-Rate-Limited RS-485/RS-422 Transceivers
- **For Low-Voltage Applications**
  MAX3483E/MAX3485E/MAX3486E/MAX3488E/MAX3490E/MAX3491E: +3.3V Powered, ±15kV ESD-Protected, 12Mbps, Slew-Rate-Limited, True RS-485/RS-422 Transceivers

Ordering Information appears at end of data sheet.

Selection Table

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/ DRIVER ENABLE	QUIESCENT CURRENT (µA)	NUMBER OF TRANSMITTERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8
MAX487	Half	0.25	Yes	Yes	Yes	120	128	8
MAX488	Full	0.25	Yes	No	No	120	32	8
MAX489	Full	0.25	Yes	No	Yes	120	32	14
MAX490	Full	2.5	No	No	No	300	32	8
MAX491	Full	2.5	No	No	Yes	300	32	14
MAX1487	Half	2.5	No	No	Yes	230	128	8

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

ABSOLUTE MAXIMUM RATINGS

Supply Voltage (V_{CC}) 12V
Control Input Voltage ($\overline{RE}$, DE) ..... -0.5V to (V_{CC} + 0.5V)
Driver Input Voltage (DI) ..... -0.5V to (V_{CC} + 0.5V)
Driver Output Voltage (A, B) ..... -8V to +12.5V
Receiver Input Voltage (A, B) ..... -8V to +12.5V
Receiver Output Voltage (RO) ..... -0.5V to (V_{CC} +0.5V)
Continuous Power Dissipation (T_A = +70°C)
8-Pin Plastic DIP (derate 9.09mW/°C above +70°C) 727mW
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C) ..800mW
8-Pin SO (derate 5.88mW/°C above +70°C) 471mW
14-Pin SO (derate 8.33mW/°C above +70°C) 667mW
8-Pin µMAX (derate 4.1mW/°C above +70°C) 830mW
8-Pin CERDIP (derate 8.00mW/°C above +70°C) 640mW
14-Pin CERDIP (derate 9.09mW/°C above +70°C) 727mW
Operating Temperature Ranges
MAX4_ _C_ _/MAX1487C_ A ..... 0°C to +70°C
MAX4_ _E_ _/MAX1487E_ A ..... -40°C to +85°C
MAX4_ _MJ_/MAX1487MJA ..... -55°C to +125°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10sec) ..... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Differential Driver Output (no load)	V_{OD1}					5	V
Differential Driver Output (with load)	V_{OD2}	R = 50Ω (RS-422)		2			V
		R = 27Ω (RS-485), Figure 4		1.5		5	
Change in Magnitude of Driver Differential Output Voltage for Complementary Output States	ΔV_{OD}	R = 27Ω or 50Ω, Figure 4				0.2	V
Driver Common-Mode Output Voltage	V_{OC}	R = 27Ω or 50Ω, Figure 4				3	V
Change in Magnitude of Driver Common-Mode Output Voltage for Complementary Output States	ΔV_{OD}	R = 27Ω or 50Ω, Figure 4				0.2	V
Input High Voltage	V_{IH}	DE, DI, $\overline{RE}$		2.0			V
Input Low Voltage	V_{IL}	DE, DI, $\overline{RE}$				0.8	V
Input Current	I_{IN1}	DE, DI, $\overline{RE}$				±2	µA
Input Current (A, B)	I_{IN2}	DE = 0V; V_{CC} = 0V or 5.25V, all devices except MAX487/MAX1487	V_{IN} = 12V			1.0	mA
			V_{IN} = -7V			-0.8	
		MAX487/MAX1487, DE = 0V, V_{CC} = 0V or 5.25V	V_{IN} = 12V			0.25	mA
			V_{IN} = -7V			-0.2	
Receiver Differential Threshold Voltage	V_{TH}	-7V ≤ V_{CM} ≤ 12V		-0.2		0.2	V
Receiver Input Hysteresis	ΔV_{TH}	V_{CM} = 0V			70		mV
Receiver Output High Voltage	V_{OH}	I_O = -4mA, V_{ID} = 200mV		3.5			V
Receiver Output Low Voltage	V_{OL}	I_O = 4mA, V_{ID} = -200mV				0.4	V
Three-State (high impedance) Output Current at Receiver	I_{OZR}	0.4V ≤ V_O ≤ 2.4V				±1	µA
Receiver Input Resistance	R_{IN}	-7V ≤ V_{CM} ≤ 12V, all devices except MAX487/MAX1487		12			kΩ
		-7V ≤ V_{CM} ≤ 12V, MAX487/MAX1487		48			kΩ

DC ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS			MIN	TYP	MAX	UNITS
No-Load Supply Current (Note 3)	I_{CC}	MAX488/MAX489, DE, DI, $\overline{RE}$ = 0V or V_{CC}				120	250	µA
		MAX490/MAX491, DE, DI, $\overline{RE}$ = 0V or V_{CC}				300	500	
		MAX481/MAX485, $\overline{RE}$ = 0V or V_{CC}	DE = V_{CC}			500	900	
			DE = 0V			300	500	
		MAX1487, $\overline{RE}$ = 0V or V_{CC}	DE = V_{CC}			300	500	
			DE = 0V			230	400	
		MAX483/MAX487, $\overline{RE}$ = 0V or V_{CC}	DE = 5V	MAX483		350	650	
				MAX487		250	400	
			DE = 0V			120	250	
Supply Current in Shutdown	I_{SHDN}	MAX481/483/487, DE = 0V, $\overline{RE}$ = V_{CC}				0.1	10	µA
Driver Short-Circuit Current, V_O = High	I_{OSD1}	-7V ≤ V_O ≤12V (Note 4)			35		250	mA
Driver Short-Circuit Current, V_O = Low	I_{OSD2}	-7V ≤ V_O ≤12V (Note 4)			35		250	mA
Receiver Short-Circuit Current	I_{OSR}	0V ≤ V_O ≤ V_{CC}			7		95	mA

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Driver Input to Output	t_{PLH}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF		10	30	60	ns
	t_{PHL}			10	30	60	
Driver Output Skew to Output	t_{SKEW}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF			5	10	ns
Driver Rise or Fall Time	t_R, t_F	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	MAX481, MAX485, MAX1487	3	15	40	ns
			MAX490C/E, MAX491C/E	5	15	25	
			MAX490M, MAX491M	3	15	40	
Driver Enable to Output High	t_{ZH}	Figures 7 and 9, C_L = 100pF, S2 closed			40	70	ns
Driver Enable to Output Low	t_{ZL}	Figures 7 and 9, C_L = 100pF, S1 closed			40	70	ns
Driver Disable Time from Low	t_{LZ}	Figures 7 and 9, C_L = 15pF, S1 closed			40	70	ns
Driver Disable Time from High	t_{HZ}	Figures 7 and 9, C_L = 15pF, S2 closed			40	70	ns
Receiver Input to Output	t_{PLH}, t_{PHL}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	MAX481, MAX485, MAX1487	20	90	200	ns
			MAX490C/E, MAX491C/E	20	90	150	
			MAX490M, MAX491M	20	90	200	
$\lvert t_{PLH} - t_{PHL} \rvert$ Differential Receiver Skew	t_{SKD}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF			13		ns
Receiver Enable to Output Low	t_{ZL}	Figures 5 and 11, C_{RL} = 15pF, S1 closed			20	50	ns
Receiver Enable to Output High	t_{ZH}	Figures 5 and 11, C_{RL} = 15pF, S2 closed			20	50	ns
Receiver Disable Time from Low	t_{LZ}	Figures 5 and 11, C_{RL} = 15pF, S1 closed			20	50	ns
Receiver Disable Time from High	t_{HZ}	Figures 5 and 11, C_{RL} = 15pF, S2 closed			20	50	ns
Maximum Data Rate	f_{MAX}			2.5			Mbps
Time to Shutdown	t_{SHDN}	MAX481 (Note 5)		50	200	600	ns

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487 (continued)

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Enable from Shutdown to Output High (MAX481)	$t_{ZH(SHDN)}$	Figures 7 and 9, C_L = 100pF, S2 closed		40	100	ns
Driver Enable from Shutdown to Output Low (MAX481)	$t_{ZL(SHDN)}$	Figures 7 and 9, C_L = 100pF, S1 closed		40	100	ns
Receiver Enable from Shutdown to Output High (MAX481)	$t_{ZH(SHDN)}$	Figures 5 and 11, C_L = 15pF, S2 closed, A - B = 2V		300	1000	ns
Receiver Enable from Shutdown to Output Low (MAX481)	$t_{ZL(SHDN)}$	Figures 5 and 11, C_L = 15pF, S1 closed, B - A = 2V		300	1000	ns

SWITCHING CHARACTERISTICS—MAX483, MAX487/MAX488/MAX489

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Input to Output	t_{PLH}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	250	800	2000	ns
	t_{PHL}		250	800	2000	
Driver Output Skew to Output	t_{SKEW}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF		100	800	ns
Driver Rise or Fall Time	t_R, t_F	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	250		2000	ns
Driver Enable to Output High	t_{ZH}	Figures 7 and 9, C_L = 100pF, S2 closed	250		2000	ns
Driver Enable to Output Low	t_{ZL}	Figures 7 and 9, C_L = 100pF, S1 closed	250		2000	ns
Driver Disable Time from Low	t_{LZ}	Figures 7 and 9, C_L = 15pF, S1 closed	300		3000	ns
Driver Disable Time from High	t_{HZ}	Figures 7 and 9, C_L = 15pF, S2 closed	300		3000	ns
Receiver Input to Output	t_{PLH}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	250		2000	ns
	t_{PHL}		250		2000	
| t_{PLH} - t_{PHL} | Differential Receiver Skew	t_{SKD}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF		100		ns
Receiver Enable to Output Low	t_{ZL}	Figures 5 and 11, C_{RL} = 15pF, S1 closed		20	50	ns
Receiver Enable to Output High	t_{ZH}	Figures 5 and 11, C_{RL} = 15pF, S2 closed		20	50	ns
Receiver Disable Time from Low	t_{LZ}	Figures 5 and 11, C_{RL} = 15pF, S1 closed		20	50	ns
Receiver Disable Time from High	t_{HZ}	Figures 5 and 11, C_{RL} = 15pF, S2 closed		20	50	ns
Maximum Data Rate	f_{MAX}	t_{PLH}, t_{PHL} < 50% of data period	250			kbps
Time to Shutdown	t_{SHDN}	MAX483/MAX487 (Note 5)	50	200	600	ns
Driver Enable from Shutdown to Output High	$t_{ZH(SHDN)}$	MAX483/MAX487, Figures 7 and 9, C_L = 100pF, S2 closed			2000	ns
Driver Enable from Shutdown to Output Low	$t_{ZL(SHDN)}$	MAX483/MAX487, Figures 7 and 9, C_L = 100pF, S1 closed			2000	ns
Receiver Enable from Shutdown to Output High	$t_{ZH(SHDN)}$	MAX483/MAX487, Figures 5 and 11, C_L = 15pF, S2 closed			2500	ns
Receiver Enable from Shutdown to Output Low	$t_{ZL(SHDN)}$	MAX483/MAX487, Figures 5 and 11, C_L = 15pF, S1 closed			2500	ns

NOTES FOR ELECTRICAL/SWITCHING CHARACTERISTICS

Note 1: All currents into device pins are positive; all currents out of device pins are negative. All voltages are referenced to device ground unless otherwise specified.

Note 2: All typical specifications are given for V_{CC} = 5V and T_A = +25°C.

Note 3: Supply current specification is valid for loaded transmitters when DE = 0V.

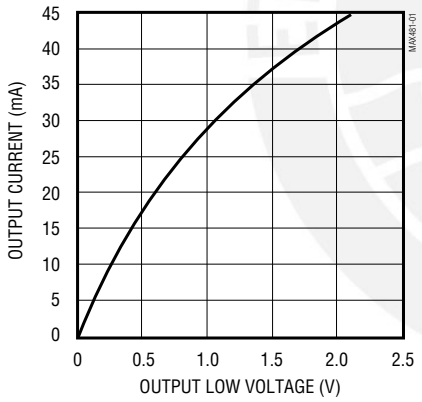
Note 4: Applies to peak current. See *Typical Operating Characteristics.*

Note 5: The MAX481/MAX483/MAX487 are put into shutdown by bringing $\overline{RE}$ high and DE low. If the inputs are in this state for less than 50ns, the parts are guaranteed not to enter shutdown. If the inputs are in this state for at least 600ns, the parts are guaranteed to have entered shutdown. See *Low-Power Shutdown Mode* section.

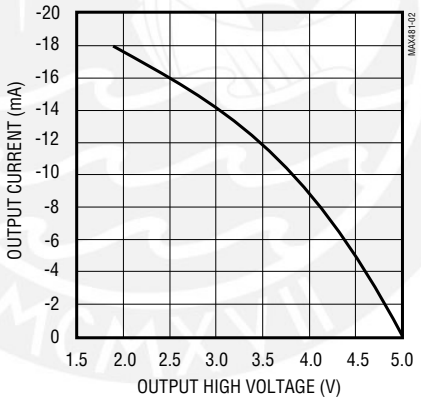
Typical Operating Characteristics

(V_{CC} = 5V, T_A = +25°C, unless otherwise noted.)

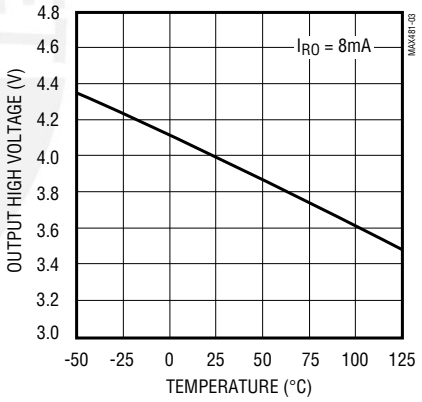
OUTPUT CURRENT vs. RECEIVER OUTPUT LOW VOLTAGE



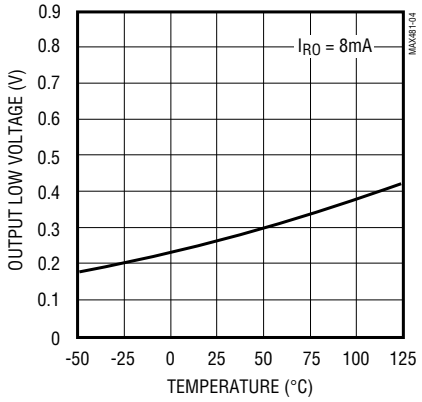
OUTPUT CURRENT vs. RECEIVER OUTPUT HIGH VOLTAGE



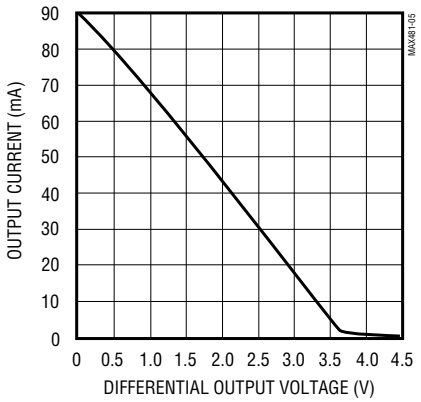
RECEIVER OUTPUT HIGH VOLTAGE vs. TEMPERATURE



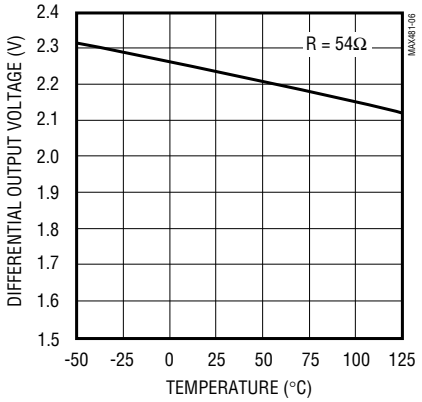
RECEIVER OUTPUT LOW VOLTAGE vs. TEMPERATURE



DRIVER OUTPUT CURRENT vs. DIFFERENTIAL OUTPUT VOLTAGE



DRIVER DIFFERENTIAL OUTPUT VOLTAGE vs. TEMPERATURE

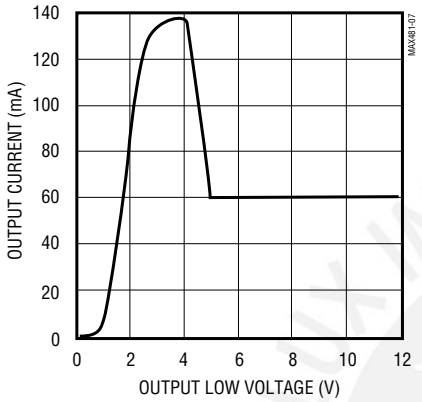


Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

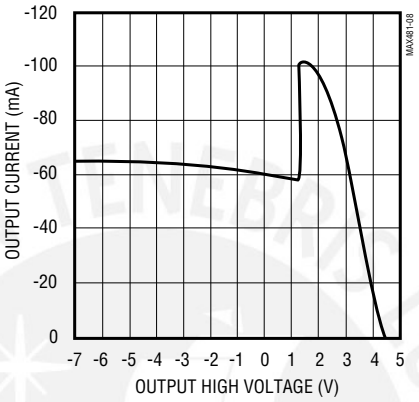
Typical Operating Characteristics (continued)

(V_{CC} = 5V, T_A = +25°C, unless otherwise noted.)

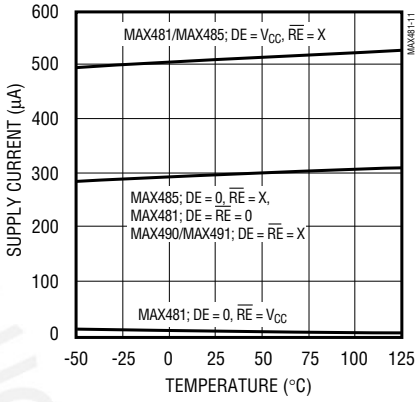
OUTPUT CURRENT vs. DRIVER OUTPUT LOW VOLTAGE



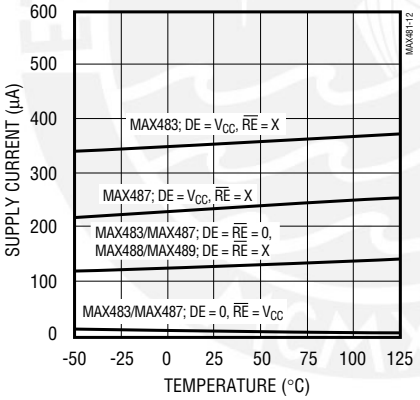
OUTPUT CURRENT vs. DRIVER OUTPUT HIGH VOLTAGE



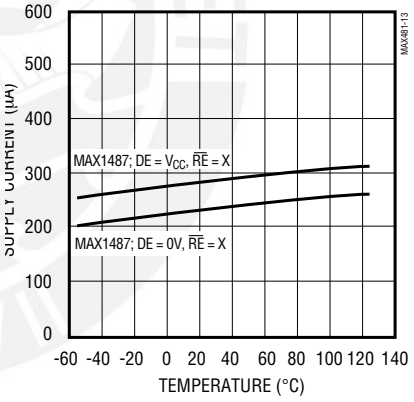
MAX481/MAX485/MAX490/MAX491 SUPPLY CURRENT vs. TEMPERATURE



MAX483/MAX487–MAX489 SUPPLY CURRENT vs. TEMPERATURE



MAX1487 SUPPLY CURRENT vs. TEMPERATURE



Pin Description

PIN					NAME	FUNCTION
MAX481/MAX483/ MAX485/MAX487/ MAX1487		MAX488/ MAX490		MAX489/ MAX491		
DIP/SO	µMAX	DIP/SO	µMAX	DIP/SO		
1	3	2	4	2	RO	Receiver Output: If A > B by 200mV, RO will be high; If A < B by 200mV, RO will be low.
2	4	—	—	3	$\overline{RE}$	Receiver Output Enable. RO is enabled when $\overline{RE}$ is low; RO is high impedance when $\overline{RE}$ is high.
3	5	—	—	4	DE	Driver Output Enable. The driver outputs, Y and Z, are enabled by bringing DE high. They are high impedance when DE is low. If the driver outputs are enabled, the parts function as line drivers. While they are high impedance, they function as line receivers if $\overline{RE}$ is low.
4	6	3	5	5	DI	Driver Input. A low on DI forces output Y low and output Z high. Similarly, a high on DI forces output Y high and output Z low.
5	7	4	6	6, 7	GND	Ground
—	—	5	7	9	Y	Noninverting Driver Output
—	—	6	8	10	Z	Inverting Driver Output
6	8	—	—	—	A	Noninverting Receiver Input and Noninverting Driver Output
—	—	8	2	12	A	Noninverting Receiver Input
7	1	—	—	—	B	Inverting Receiver Input and Inverting Driver Output
—	—	7	1	11	B	Inverting Receiver Input
8	2	1	3	14	V_{CC}	Positive Supply: 4.75V ≤ V_{CC} ≤ 5.25V
—	—	—	—	1, 8, 13	N.C.	No Connect—not internally connected

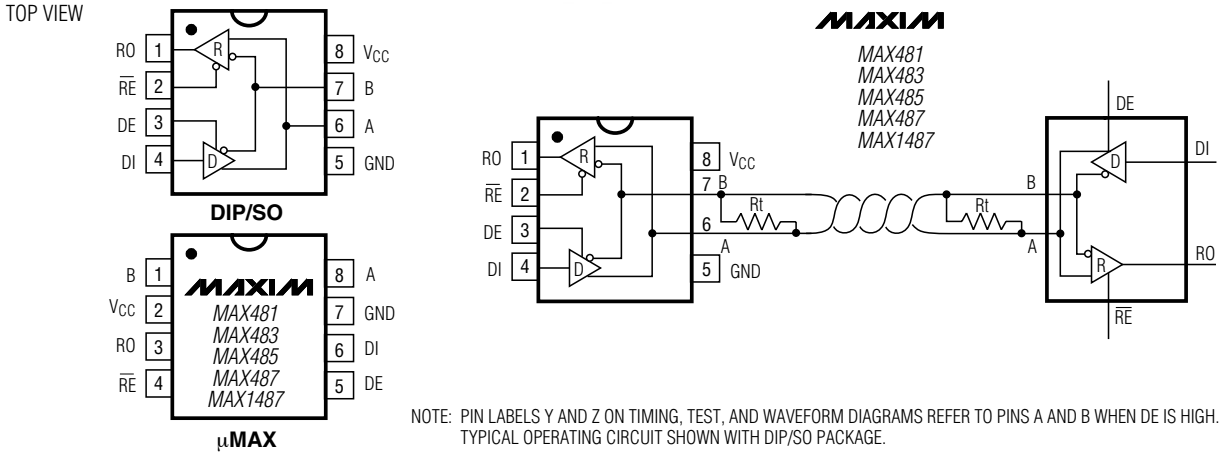


Figure 1. MAX481/MAX483/MAX485/MAX487/MAX1487 Pin Configuration and Typical Operating Circuit

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

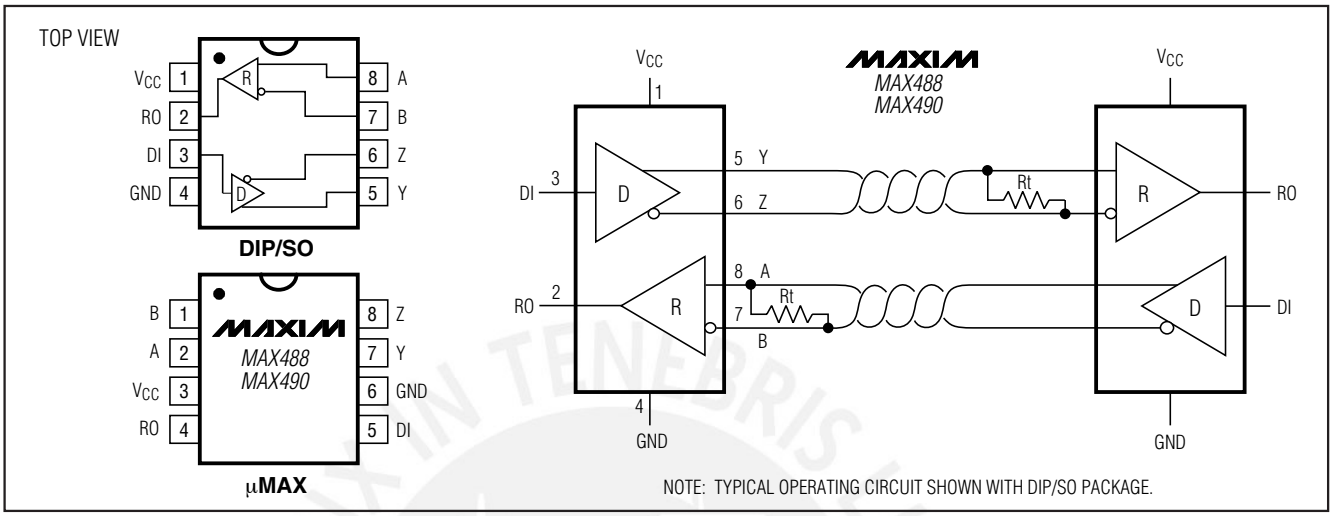


Figure 2. MAX488/MAX490 Pin Configuration and Typical Operating Circuit

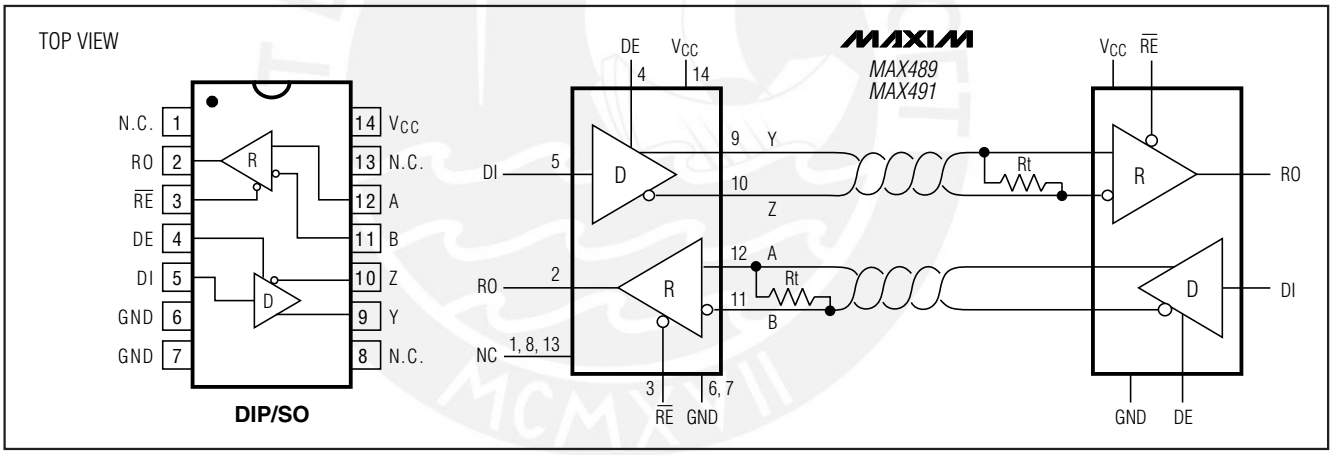


Figure 3. MAX489/MAX491 Pin Configuration and Typical Operating Circuit

Applications Information

The MAX481/MAX483/MAX485/MAX487–MAX491 and MAX1487 are low-power transceivers for RS-485 and RS-422 communications. The MAX481, MAX485, MAX490, MAX491, and MAX1487 can transmit and receive at data rates up to 2.5Mbps, while the MAX483, MAX487, MAX488, and MAX489 are specified for data rates up to 250kbps. The MAX488–MAX491 are full-duplex transceivers while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are half-duplex. In addition, Driver Enable (DE) and Receiver Enable ($\overline{RE}$) pins are included on the MAX481, MAX483, MAX485, MAX487, MAX489, MAX491, and MAX1487. When disabled, the driver and receiver outputs are high impedance.

MAX487/MAX1487: 128 Transceivers on the Bus

The 48kΩ, 1/4-unit-load receiver input impedance of the MAX487 and MAX1487 allows up to 128 transceivers on a bus, compared to the 1-unit load (12kΩ input impedance) of standard RS-485 drivers (32 transceivers maximum). Any combination of MAX487/MAX1487 and other RS-485 transceivers with a total of 32 unit loads or less can be put on the bus. The MAX481/MAX483/MAX485 and MAX488–MAX491 have standard 12kΩ Receiver Input impedance.

Test Circuits

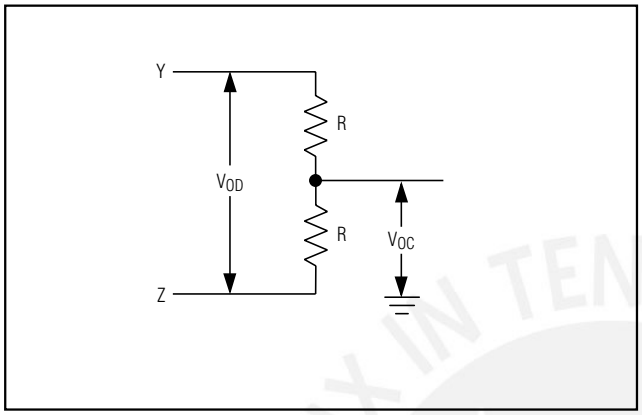


Figure 4. Driver DC Test Load

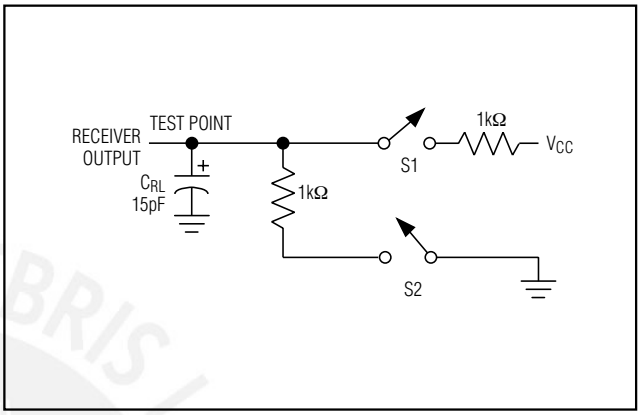


Figure 5. Receiver Timing Test Load

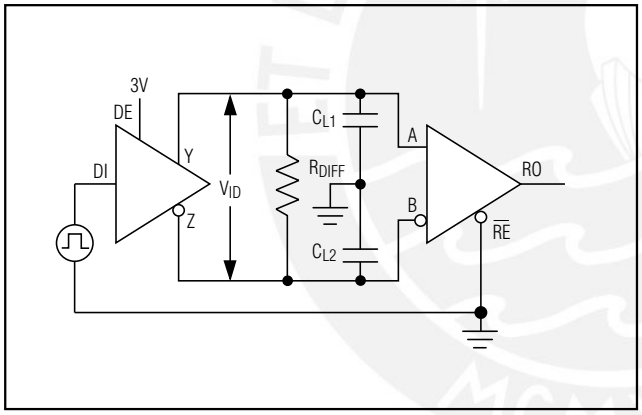


Figure 6. Driver/Receiver Timing Test Circuit

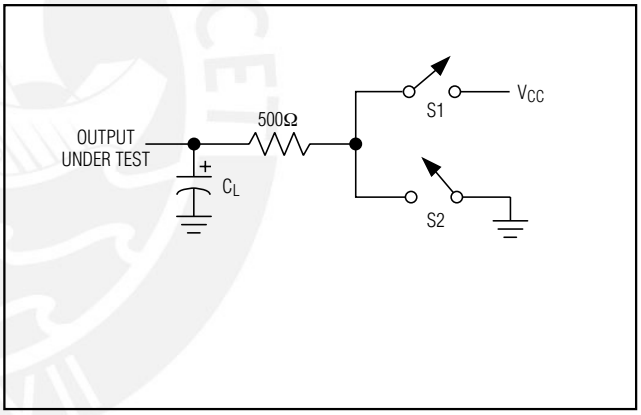


Figure 7. Driver Timing Test Load

MAX483/MAX487/MAX488/MAX489: Reduced EMI and Reflections

The MAX483 and MAX487–MAX489 are slew-rate limited, minimizing EMI and reducing reflections caused by improperly terminated cables. Figure 12 shows the driver output waveform and its Fourier analysis of a 150kHz signal transmitted by a MAX481, MAX485, MAX490, MAX491, or MAX1487. High-frequency harmonics with large amplitudes are evident. Figure 13 shows the same information displayed for a MAX483, MAX487, MAX488, or MAX489 transmitting under the same conditions. Figure 13's high-frequency harmonics have much lower amplitudes, and the potential for EMI is significantly reduced.

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Switching Waveforms

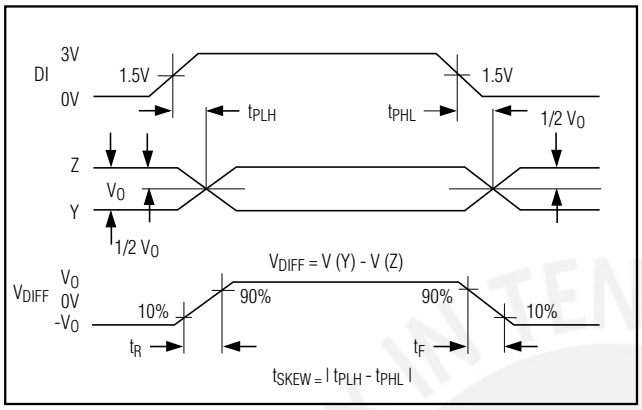


Figure 8. Driver Propagation Delays

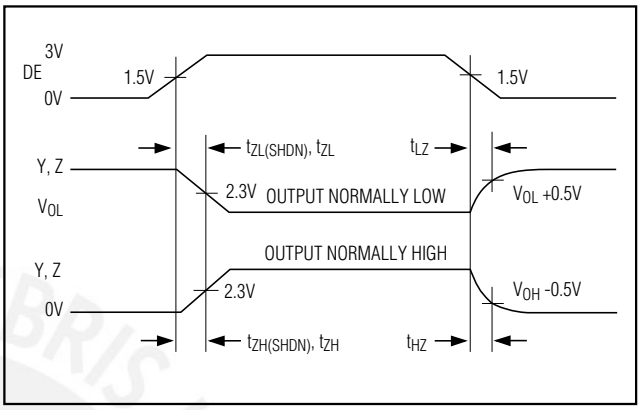


Figure 9. Driver Enable and Disable Times (except MAX488 and MAX490)

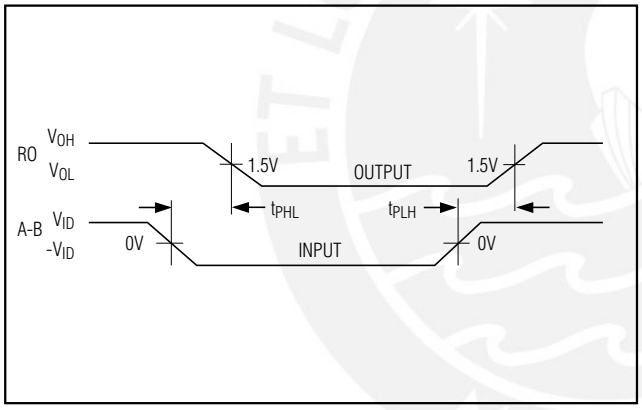


Figure 10. Receiver Propagation Delays

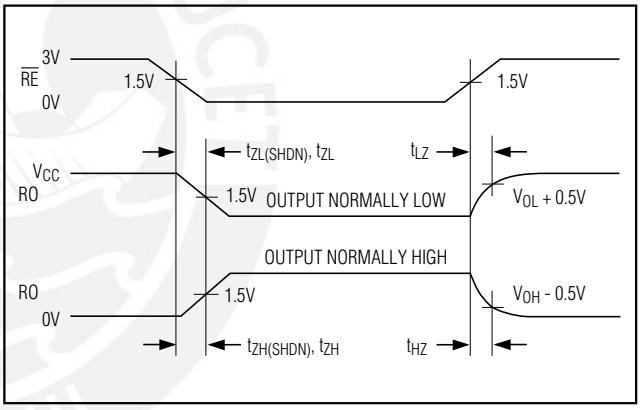


Figure 11. Receiver Enable and Disable Times (except MAX488 and MAX490)

Function Tables (MAX481/MAX483/MAX485/MAX487/MAX1487)

Table 1. Transmitting

INPUTS			OUTPUTS	
$\overline{RE}$	DE	DI	Z	Y
X	1	1	0	1
X	1	0	1	0
0	0	X	High-Z	High-Z
1	0	X	High-Z*	High-Z*

X = Don't care
High-Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX487

Table 2. Receiving

INPUTS			OUTPUT
$\overline{RE}$	DE	A-B	RO
0	0	≥ +0.2V	1
0	0	≤ -0.2V	0
0	0	Inputs open	1
1	0	X	High-Z*

X = Don't care
High-Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX487

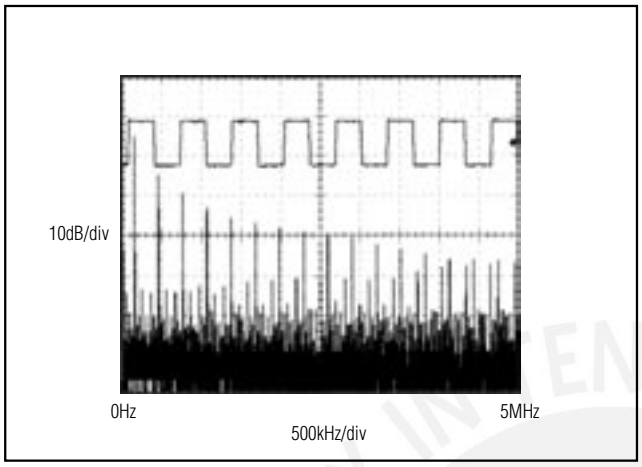


Figure 12. Driver Output Waveform and FFT Plot of MAX481/ MAX485/MAX490/MAX491/MAX1487 Transmitting a 150kHz Signal

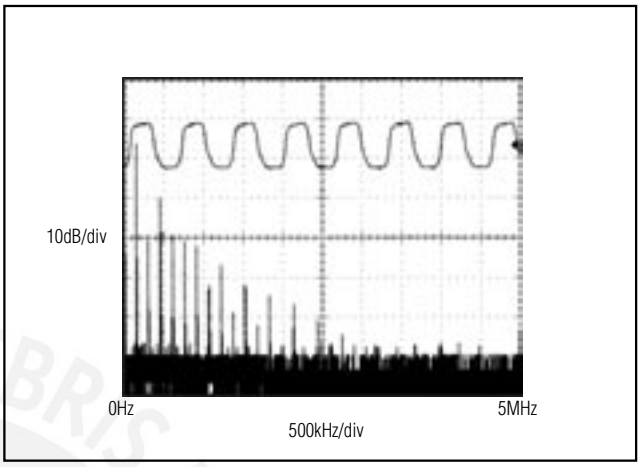


Figure 13. Driver Output Waveform and FFT Plot of MAX483/ MAX487–MAX489 Transmitting a 150kHz Signal

Low-Power Shutdown Mode (MAX481/MAX483/MAX487)

A low-power shutdown mode is initiated by bringing both $\overline{RE}$ high and DE low. The devices will not shut down unless both the driver and receiver are disabled. In shutdown, the devices typically draw only 0.1µA of supply current.

$\overline{RE}$ and DE may be driven simultaneously; the parts are guaranteed not to enter shutdown if $\overline{RE}$ is high and DE is low for less than 50ns. If the inputs are in this state for at least 600ns, the parts are guaranteed to enter shutdown.

For the MAX481, MAX483, and MAX487, the t_{ZH} and t_{ZL} enable times assume the part was not in the low-power shutdown state (the MAX485/MAX488–MAX491 and MAX1487 can not be shut down). The $t_{ZH(SHDN)}$ and $t_{ZL(SHDN)}$ enable times assume the parts were shut down (see *Electrical Characteristics*).

It takes the drivers and receivers longer to become enabled from the low-power shutdown state ($t_{ZH(SHDN)}$, $t_{ZL(SHDN)}$) than from the operating mode (t_{ZH}, t_{ZL}). (The parts are in operating mode if the $\overline{RE}$, DE inputs equal a logical 0,1 or 1,1 or 0, 0.)

Driver Output Protection

Excessive output current and power dissipation caused by faults or by bus contention are prevented by two mechanisms. A foldback current limit on the output stage provides immediate protection against short circuits over the whole common-mode voltage range (see *Typical Operating Characteristics*). In addition, a thermal shutdown circuit forces the driver outputs into a high-impedance state if the die temperature rises excessively.

Propagation Delay

Many digital encoding schemes depend on the difference between the driver and receiver propagation delay times. Typical propagation delays are shown in Figures 15–18 using Figure 14's test circuit.

The difference in receiver delay times, $|t_{PLH} - t_{PHL}|$, is typically under 13ns for the MAX481, MAX485, MAX490, MAX491, and MAX1487 and is typically less than 100ns for the MAX483 and MAX487–MAX489.

The driver skew times are typically 5ns (10ns max) for the MAX481, MAX485, MAX490, MAX491, and MAX1487, and are typically 100ns (800ns max) for the MAX483 and MAX487–MAX489.

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

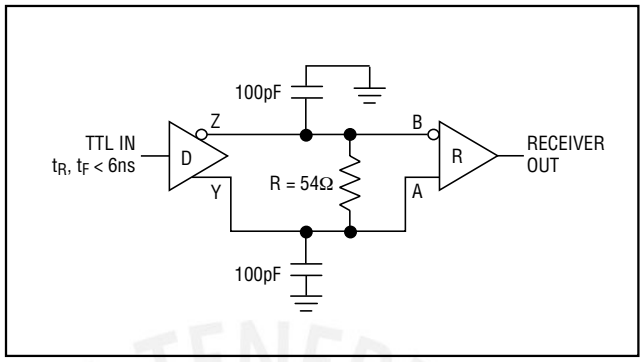


Figure 14. Receiver Propagation Delay Test Circuit

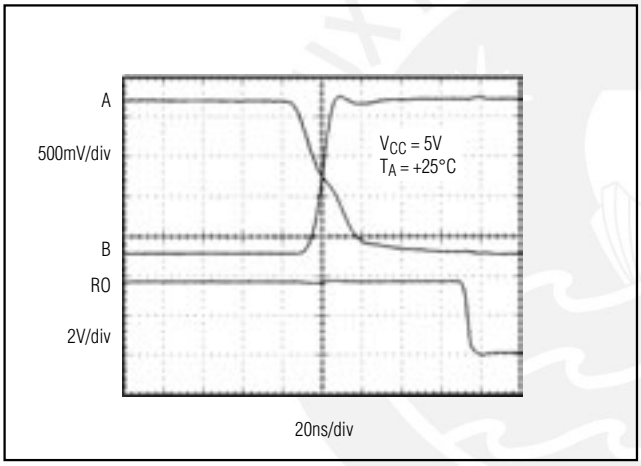


Figure 15. MAX481/MAX485/MAX490/MAX491/MAX1487 Receiver t_{PHL}

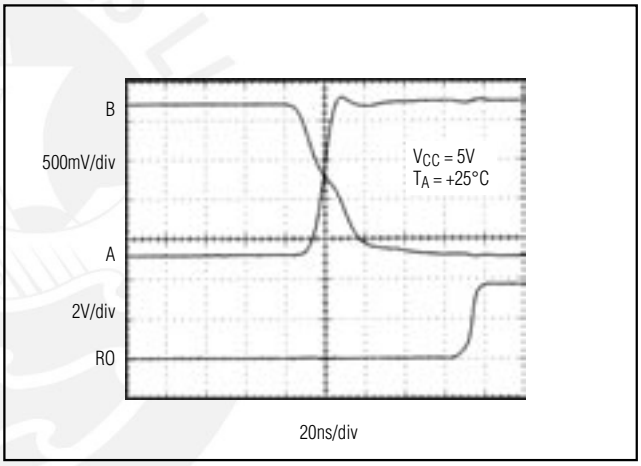


Figure 16. MAX481/MAX485/MAX490/MAX491/MAX1487 Receiver t_{PLH}

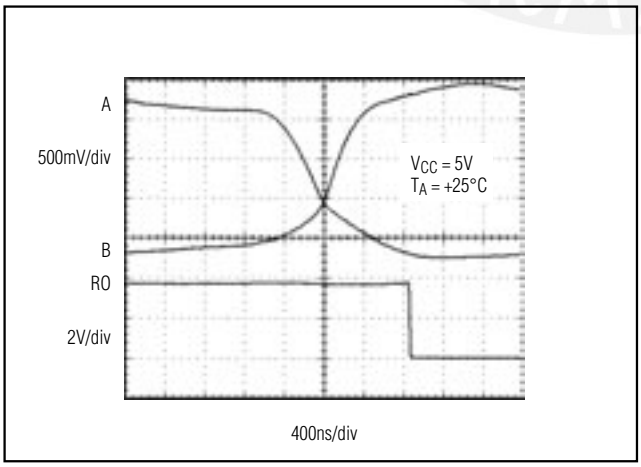


Figure 17. MAX483, MAX487–MAX489 Receiver t_{PHL}

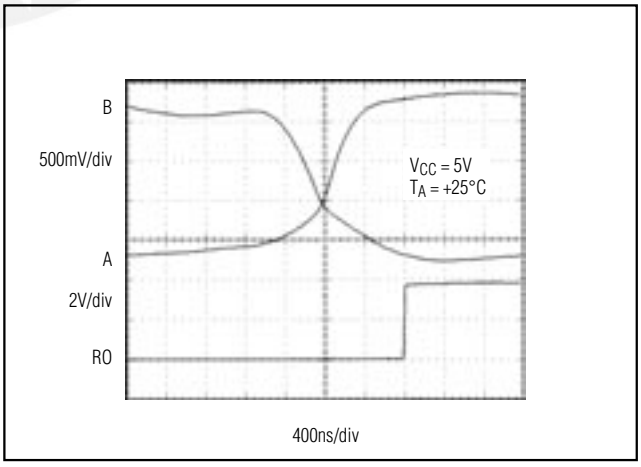


Figure 18. MAX483, MAX487–MAX489 Receiver t_{PLH}

Line Length vs. Data Rate

The RS-485/RS-422 standard covers line lengths up to 4000 feet. For line lengths greater than 4000 feet, see Figure 23.

Figures 19 and 20 show the system differential voltage for the parts driving 4000 feet of 26AWG twisted-pair wire at 110kHz into 120Ω loads.

Typical Applications

The MAX481, MAX483, MAX485, MAX487–MAX491, and MAX1487 transceivers are designed for bidirectional data communications on multipoint bus transmission lines.

Figures 21 and 22 show typical network applications circuits. These parts can also be used as line repeaters, with cable lengths longer than 4000 feet, as shown in Figure 23.

To minimize reflections, the line should be terminated at both ends in its characteristic impedance, and stub lengths off the main line should be kept as short as possible. The slew-rate-limited MAX483 and MAX487–MAX489 are more tolerant of imperfect termination.

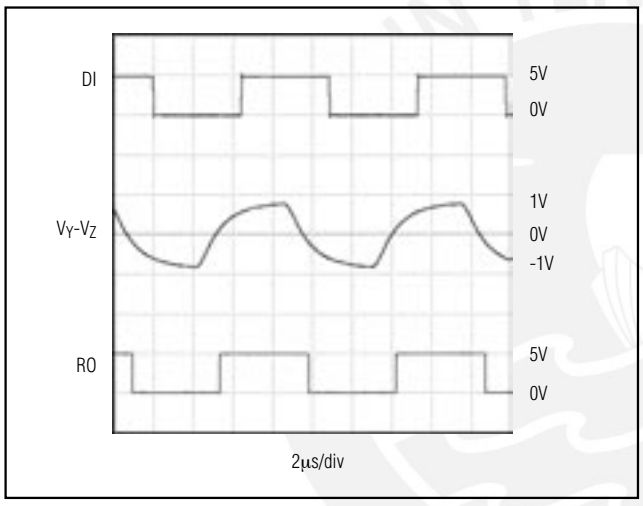


Figure 19. MAX481/MAX485/MAX490/MAX491/MAX1487 System Differential Voltage at 110kHz Driving 4000ft of Cable

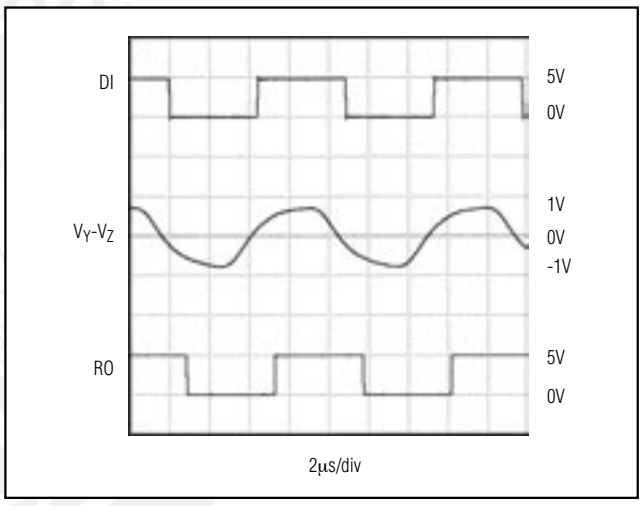


Figure 20. MAX483, MAX487–MAX489 System Differential Voltage at 110kHz Driving 4000ft of Cable

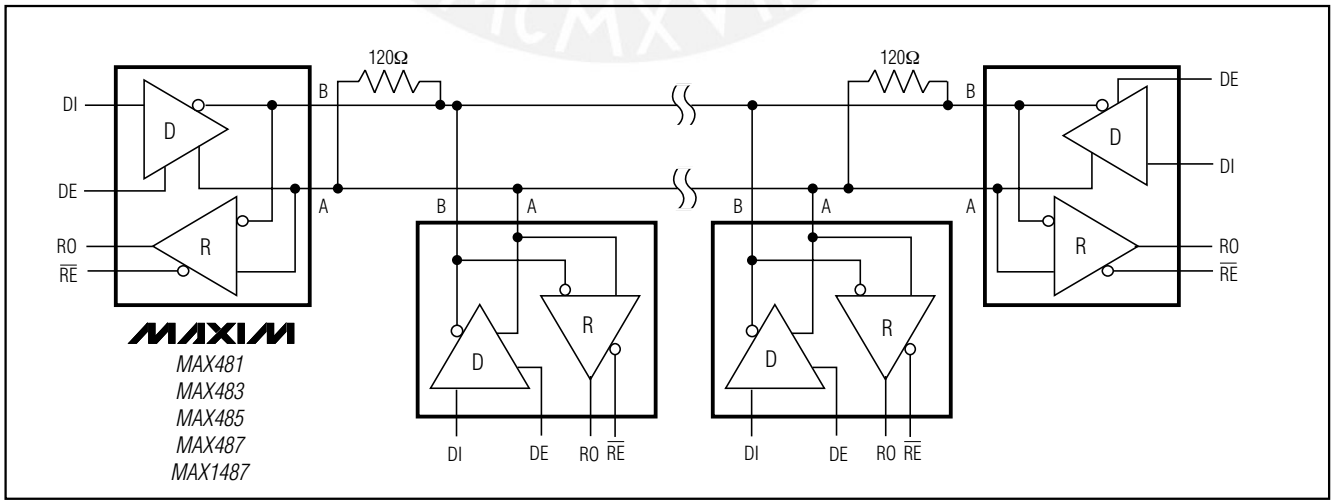


Figure 21. MAX481/MAX483/MAX485/MAX487/MAX1487 Typical Half-Duplex RS-485 Network

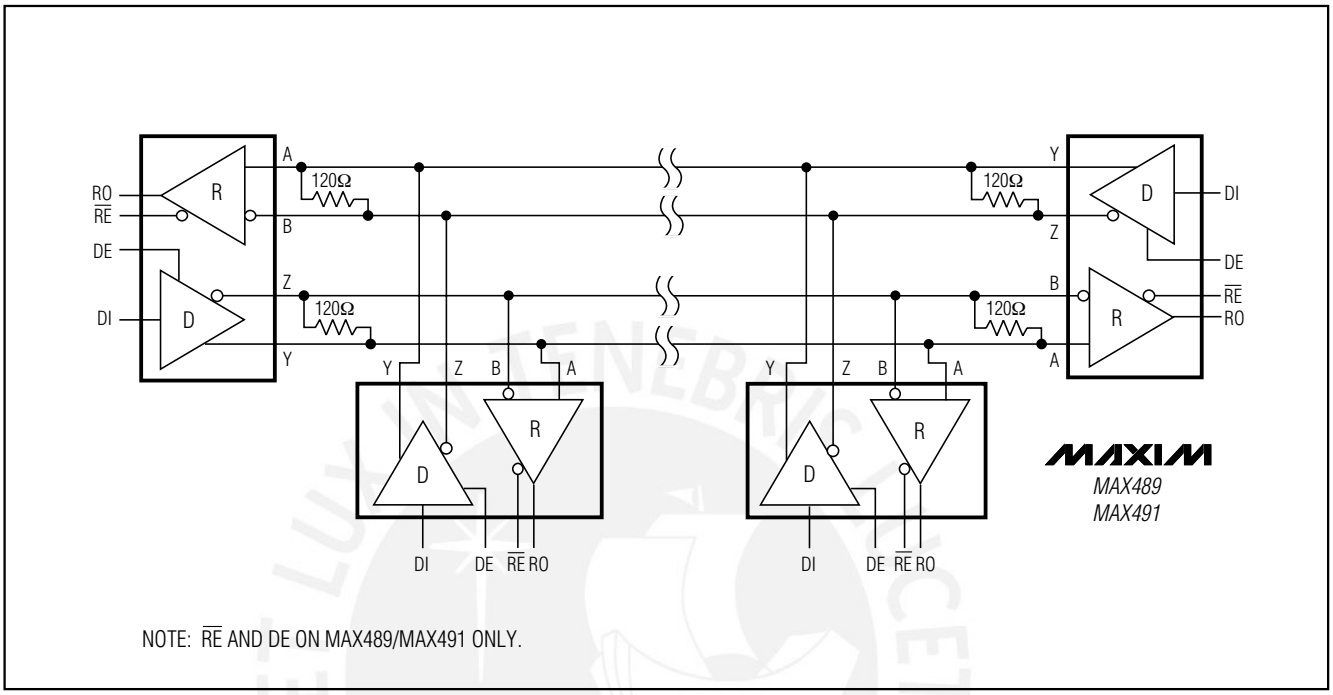


Figure 22. MAX488–MAX491 Full-Duplex RS-485 Network

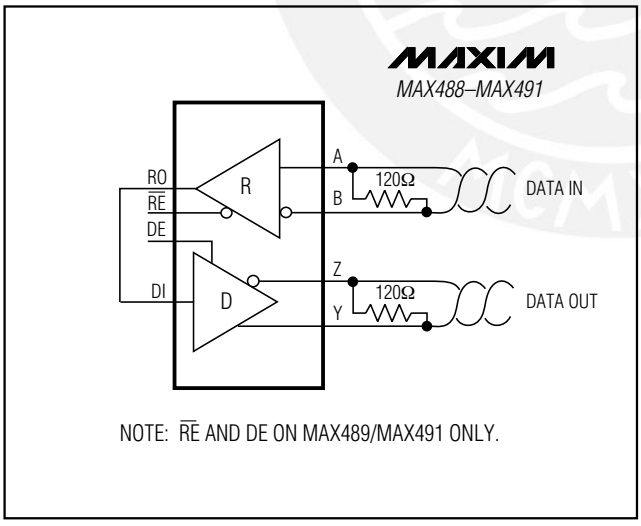


Figure 23. Line Repeater for MAX488–MAX491

Isolated RS-485

For isolated RS-485 applications, see the MAX253 and MAX1480 data sheets.

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX481CPA	0°C to +70°C	8 Plastic DIP
MAX481CSA	0°C to +70°C	8 SO
MAX481CUA	0°C to +70°C	8 µMAX
MAX481C/D	0°C to +70°C	Dice*
MAX481EPA	-40°C to +85°C	8 Plastic DIP
MAX481ESA	-40°C to +85°C	8 SO
MAX481MJA	-55°C to +125°C	8 CERDIP
MAX483CPA	0°C to +70°C	8 Plastic DIP
MAX483CSA	0°C to +70°C	8 SO
MAX483CUA	0°C to +70°C	8 µMAX
MAX483C/D	0°C to +70°C	Dice*
MAX483EPA	-40°C to +85°C	8 Plastic DIP
MAX483ESA	-40°C to +85°C	8 SO
MAX483MJA	-55°C to +125°C	8 CERDIP
MAX485CPA	0°C to +70°C	8 Plastic DIP
MAX485CSA	0°C to +70°C	8 SO
MAX485CUA	0°C to +70°C	8 µMAX
MAX485C/D	0°C to +70°C	Dice*
MAX485EPA	-40°C to +85°C	8 Plastic DIP
MAX485ESA	-40°C to +85°C	8 SO
MAX485MJA	-55°C to +125°C	8 CERDIP
MAX487CPA	0°C to +70°C	8 Plastic DIP
MAX487CSA	0°C to +70°C	8 SO
MAX487CUA	0°C to +70°C	8 µMAX
MAX487C/D	0°C to +70°C	Dice*
MAX487EPA	-40°C to +85°C	8 Plastic DIP
MAX487ESA	-40°C to +85°C	8 SO
MAX487MJA	-55°C to +125°C	8 CERDIP
MAX488CPA	0°C to +70°C	8 Plastic DIP
MAX488CSA	0°C to +70°C	8 SO
MAX488CUA	0°C to +70°C	8 µMAX
MAX488C/D	0°C to +70°C	Dice*
MAX488EPA	-40°C to +85°C	8 Plastic DIP
MAX488ESA	-40°C to +85°C	8 SO
MAX488MJA	-55°C to +125°C	8 CERDIP
MAX489CPD	0°C to +70°C	14 Plastic DIP
MAX489CSD	0°C to +70°C	14 SO
MAX489C/D	0°C to +70°C	Dice*
MAX489EPD	-40°C to +85°C	14 Plastic DIP
MAX489ESD	-40°C to +85°C	14 SO
MAX489MJD	-55°C to +125°C	14 CERDIP

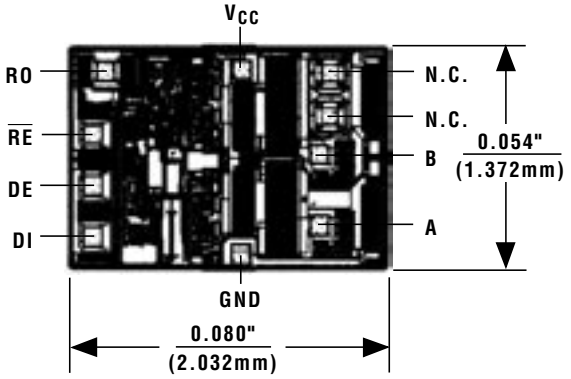
Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX490CPA	0°C to +70°C	8 Plastic DIP
MAX490CSA	0°C to +70°C	8 SO
MAX490CUA	0°C to +70°C	8 µMAX
MAX490C/D	0°C to +70°C	Dice*
MAX490EPA	-40°C to +85°C	8 Plastic DIP
MAX490ESA	-40°C to +85°C	8 SO
MAX490MJA	-55°C to +125°C	8 CERDIP
MAX491CPD	0°C to +70°C	14 Plastic DIP
MAX491CSD	0°C to +70°C	14 SO
MAX491C/D	0°C to +70°C	Dice*
MAX491EPD	-40°C to +85°C	14 Plastic DIP
MAX491ESD	-40°C to +85°C	14 SO
MAX491MJD	-55°C to +125°C	14 CERDIP
MAX1487CPA	0°C to +70°C	8 Plastic DIP
MAX1487CSA	0°C to +70°C	8 SO
MAX1487CUA	0°C to +70°C	8 µMAX
MAX1487C/D	0°C to +70°C	Dice*
MAX1487EPA	-40°C to +85°C	8 Plastic DIP
MAX1487ESA	-40°C to +85°C	8 SO
MAX1487MJA	-55°C to +125°C	8 CERDIP

** Contact factory for dice specifications.*

Chip Topographies

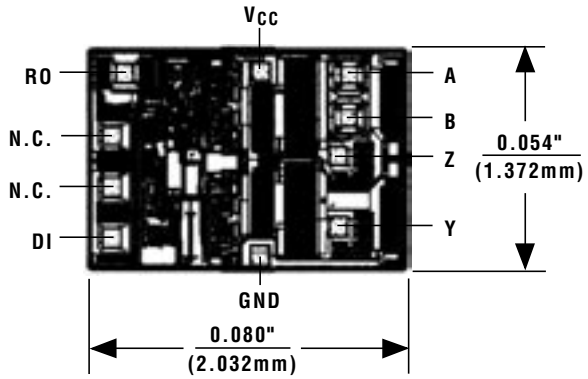
MAX481/MAX483/MAX485/MAX487/MAX1487



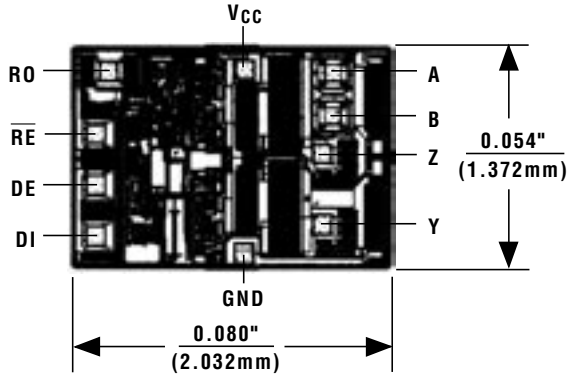
Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Chip Topographies (continued)

MAX488/MAX490



MAX489/MAX491

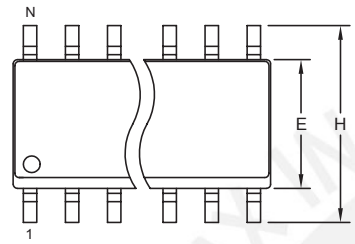


TRANSISTOR COUNT: 248
SUBSTRATE CONNECTED TO GND

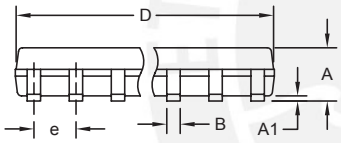
Package Information

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to **www.maxim-ic.com/packages**.)

SOICN .EPS



TOP VIEW



FRONT VIEW

SIDE VIEW

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.053	0.069	1.35	1.75
A1	0.004	0.010	0.10	0.25
B	0.014	0.019	0.35	0.49
C	0.007	0.010	0.19	0.25
e	0.050 BSC		1.27 BSC	
E	0.150	0.157	3.80	4.00
H	0.228	0.244	5.80	6.20
L	0.016	0.050	0.40	1.27

VARIATIONS:

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX	N	MS012
D	0.189	0.197	4.80	5.00	8	AA
D	0.337	0.344	8.55	8.75	14	AB
D	0.386	0.394	9.80	10.00	16	AC

NOTES:
1. D&E DO NOT INCLUDE MOLD FLASH.
2. MOLD FLASH OR PROTRUSIONS NOT TO EXCEED 0.15mm (.006").
3. LEADS TO BE COPLANAR WITHIN 0.10mm (.004").
4. CONTROLLING DIMENSION: MILLIMETERS.
5. MEETS JEDEC MS012.
6. N = NUMBER OF PINS.

DALLAS SEMICONDUCTOR MAXIM

PROPRIETARY INFORMATION

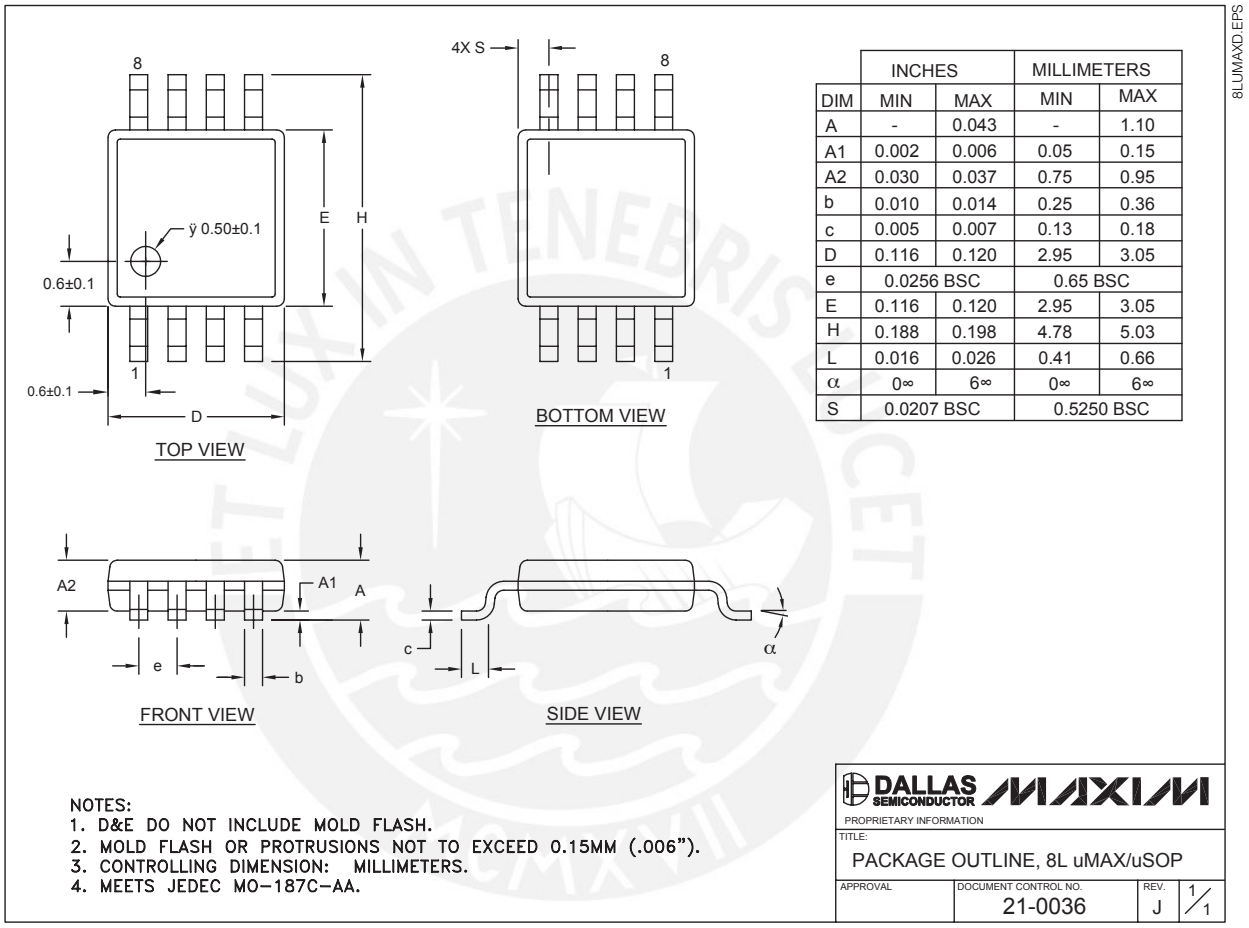
TITLE: PACKAGE OUTLINE, .150" SOIC

APPROVAL	DOCUMENT CONTROL NO.	REV.	
	21-0041	B	1/1

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Package Information (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to **www.maxim-ic.com/packages**.)

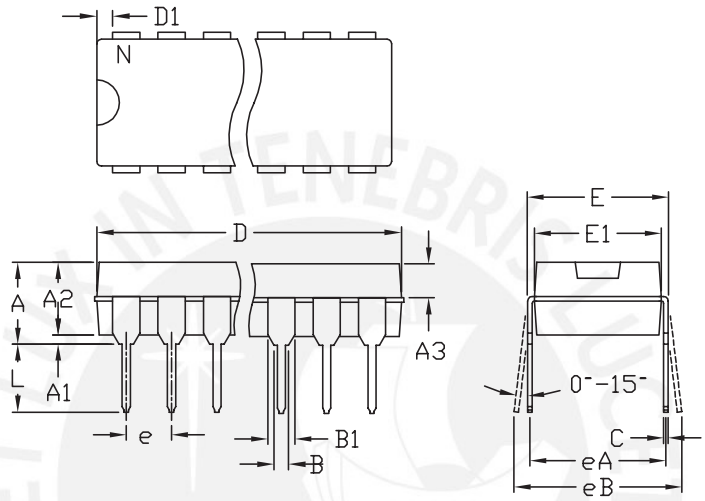


	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	-	0.043	-	1.10
A1	0.002	0.006	0.05	0.15
A2	0.030	0.037	0.75	0.95
b	0.010	0.014	0.25	0.36
c	0.005	0.007	0.13	0.18
D	0.116	0.120	2.95	3.05
e	0.0256 BSC		0.65 BSC	
E	0.116	0.120	2.95	3.05
H	0.188	0.198	4.78	5.03
L	0.016	0.026	0.41	0.66
α	0∞	6∞	0∞	6∞
S	0.0207 BSC		0.5250 BSC	

Package Information (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to **www.maxim-ic.com/packages**.)

PDIPN.EPS



	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	---	0.180	---	4.572
A1	0.015	---	0.38	---
A2	0.125	0.175	3.18	4.45
A3	0.055	0.080	1.40	2.03
B	0.015	0.022	0.381	0.56
B1	0.045	0.065	1.14	1.65
C	0.008	0.014	0.2	0.355
D1	0.005	0.080	0.13	2.03
E	0.300	0.325	7.62	8.26
E1	0.240	0.310	6.10	7.87
e	0.100 BSC.		2.54 BSC.	
eA	0.300 BSC.		7.62 BSC.	
eB	0.400 BSC.		10.16 BSC.	
L	0.115	0.150	2.921	3.81

	INCHES		MILLIMETERS			
	MIN	MAX	MIN	MAX	N	MS001
D	0.348	0.390	8.84	9.91	8	AB
D	0.735	0.765	18.67	19.43	14	AC
D	0.745	0.765	18.92	19.43	16	AA
D	0.885	0.915	22.48	23.24	18	AD
D	1.015	1.045	25.78	26.54	20	AE
D	1.14	1.265	28.96	32.13	24	AF
D	1.360	1.380	34.54	35.05	28	*5

NOTES:
1. D&E DO NOT INCLUDE MOLD FLASH
2. MOLD FLASH OR PROTRUSIONS NOT TO EXCEED .15mm (.006")
3. CONTROLLING DIMENSION: MILLIMETER
4. MEETS JEDEC MS001-XX AS SHOWN IN ABOVE TABLE
5. SIMILIAR TO JEDEC MO-058AB
6. N = NUMBER OF PINS

MAXIM 120 SAN GABRIEL DR SUNNYVALE CA 94086 FAX (408) 737 7194
PROPRIETARY INFORMATION

TITLE: PACKAGE FAMILY OUTLINE: PDIP .300" | 1/1 | DOCUMENT CONTROL NUMBER 21-0043 D REV

MOP-AL204A

Parallel Display Specifications

Revision 1.0

Revision History

Revision	Description	Author
1.0	Initial Release	Clark

Contents

Features

The Matrix Orbital Parallel display series offers a low cost display solution utilizing an industry standard communication interface for simple integration into a wide variety of new and existing applications. The Light Emitting Diode backlight with configurable brightness and voltage controlled contrast allows the MOP Liquid Crystal Display line to offer a professional display solution with low power impact for any project. The standard alphanumeric font set also allows up to eight custom characters to be saved in display Random Access Memory for a custom design touch.

Hardware

Drawing

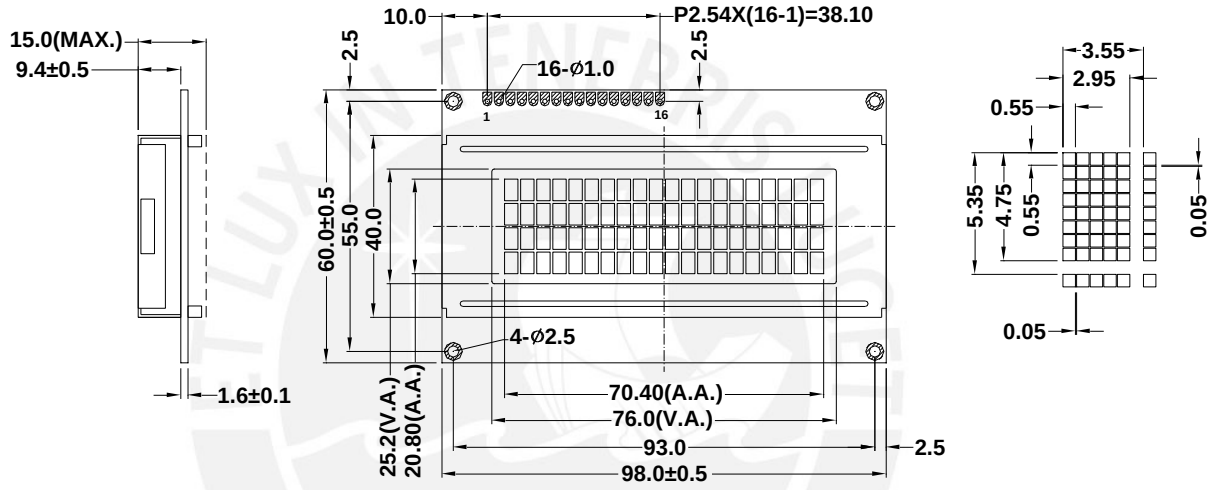


Figure 1: MOP-AL204A Mechanical Drawing

Interface

Table 1: Display Control

Pin	Symbol	Description
1	V_{SS}	Ground
2	V_{DD}	Supply Voltage for Logic
3	V_0	Supply Voltage for LCD (Contrast)
4	RS	Register Select
5	R/W	Read/Write
6	CE	Chip Enable
15	LED(+)	Anode of LED Backlight
16	LED(-)	Cathode of LED Backlight

Table 2: Parallel Data

Pin	Symbol	Description
7	DB0	*Data bit 0
8	DB1	*Data bit 1
9	DB2	*Data bit 2
10	DB3	*Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7

***Note:** Not used in 4-bit mode

Instructions

Outline

The MOP line is controlled using a standard HD44780 compliant controller. The display is enabled by pulling the Chip Enable (CE) pin high, communication to and from the device is controlled using the Read/Write (R/W) input, and one of two available 8-bit registers are selected via the Register Select (RS) line. Using Register Select, either the Instruction Register (IR) or Data Register (DR) is selected by toggling RS low or high respectively.

While executing from the IR, the display will pull the Most Significant Bit of the data bus, DB7, high. While this Busy Flag (BF) is set, any instructions sent to the unit will be ignored. The status of this flag and the current position of the Address Counter (AC) can be obtained by performing a read operation on the instruction register at any time.

Table 3: Register Selection

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	Write data to DDRAM or CGRAM (DR to DDRAM or CGRAM)
1	1	Read data from DDRAM or CGRAM (DDRAM or CGRAM to DR)

When writing for the DR, one of two locations can be chosen using the AC. The value provided to the AC when executing a set address command differentiates these locations. The AC is automatically decremented or incremented after a read or a write.

DDRAM provides eighty bytes of display memory to all displays. Memory outside the bounds of the display area can be used as general RAM. DDRAM addressing begins at the top left of the display with a value of 0, addresses then increment from left to right then down once a row is filled.

Table 4: One Line Addressing

Position	1	2	...	80
DDRAM Address	00	01	...	4F

Table 5: Two Line Addressing

Position	1	2	...	40
DDRAM Address	00	01	...	27
	40	41	...	67

Table 6: Four Line Addressing

Position	1	2	...	20
DDRAM Address	00	01	...	13
	40	41	...	53
	14	15	...	27
	54	55	...	67

CGRAM provides eight custom characters that can be created by writing to CGRAM locations then displayed using the first eight CGROM character codes, as seen in the character ROM table below.

Characters are sent to the display by performing a write operation on the DR using the correct character address within CGROM. Instructions are issued by writing to the IR; a complete list is available below.

Instruction Table

Table 7: Parallel Instruction Table

Instruction	Instruction Code										Description
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to all DDRAM locations, set DDRAM address to "00H", return cursor to its original position, and set I/D to "1".
Return Home	0	0	0	0	0	0	0	0	1	—	Set DDRAM address to "00H" and return cursor to its original position if shifted. The contents of DDRAM are not changed.
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display. DDRAM and CRAM addresses are incremented and cursor moves right when I/D is set to "1", the opposite is true when reset to "0". Setting SH to "1" causes the entire display to shift affecting only DDRAM.
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor (C), and blinking of cursor (B) on/off control bit. Setting D, C, or B to "1" will cause the display, underline cursor, or blinking cursor to turn on, the opposite is true for reset.
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	—	—	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data. Setting S/L to "1" will shift the screen horizontally while the opposite will move the cursor through all screen positions. Setting R/L to "1" will shift right immediately. AC and DDRAM are not altered.
Function Set	0	0	0	0	1	DL	N	F	—	—	Set interface data length, numbers of display line and, display font type. Setting DL to "1" specifies 8-bit mode, "0" 4-bit. Setting N to "1" permits a multi-line display, "0" a single. Resetting F to "0" indicates a 5x8 dot character.
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Read the status of the display controller through the BF Bit. The contents of address counter can also be read.
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM), location is determined by the AC. AC and display shift are adjusted as specified.
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM), location is determined by the AC, set command is recommended previous to this. Only AC is adjusted.

Character ROM

The character generator ROM stores up to two hundred fifty-six 5×8 dot character patterns from 8-bit character codes. The first eight characters are reserved for custom characters saved in CGRAM.

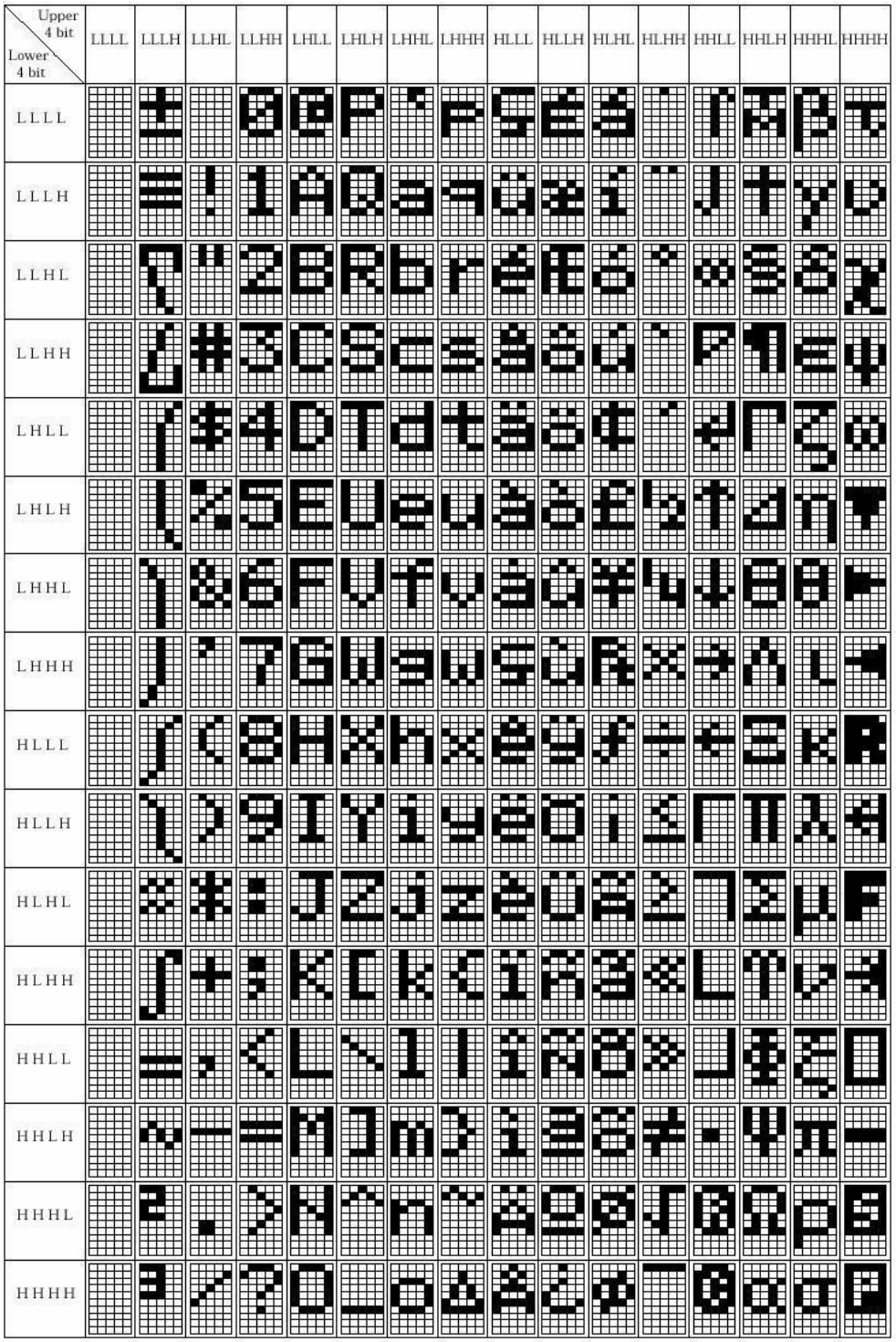


Figure 2: European Character Set

Character RAM

CGRAM allows the creation of up to eight 5x8 character patterns. Eight bytes are assigned to each character address, the least significant five bits of which represent the five pixel columns. Pixels are activated by setting the bit in their position in CGRAM to "1".

Each character has eight addresses in CGRAM corresponding to each of its eight pixel rows. The highest three bits represent the character address in DDRAM. The lowest three bits of this address represent the row positions beginning with 0 at the top. The last row will be logically OR'd with the cursor when it is active.

Finally, each character can be referenced in DDRAM and written to the screen using its eight bit address.

Table 8:Relationship between CGRAM Addresses, Character Codes (DDRAM Data) and Character Patterns (CGRAM Data)

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)
7 6 5 4 3 2 1 0 (High … Low)	5 4 3 2 1 0 (High … Low)	7 6 5 4 3 2 1 0 (High … Low)
0 0 0 0 * 0 0 0	0 0 0	* * * 1 1 1 1 0 / 1 0 0 0 1 / 1 0 0 0 1 / 1 1 1 1 0 / 1 0 1 0 0 / 1 0 0 1 0 / 1 0 0 0 1 / * * * 0 0 0 0 0 — Character pattern (1); Cursor position
0 0 0 0 * 0 0 1	0 0 1	* * * 1 0 0 0 1 / 0 1 0 1 0 / 1 1 1 1 1 / 0 0 1 0 0 / 1 1 1 1 1 / 0 0 1 0 0 / 0 0 1 0 0 / * * * 0 0 0 0 0 — Character pattern (2); Cursor position
	0 0 0 / 0 0 1	* * *
0 0 0 0 * 1 1 1	1 1 1 / 1 0 0 / 1 0 1 / 1 1 0 / 1 1 1	* * *

Note: * Indicates no effect.

Timing Characteristics

Table 9: Read and Write Operation Specifications

		Write			Read			
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Enable cycle time	t_{cycE}	1200	—	—	1200	—	—	ns
Enable pulse width (high level)	PW_{EH}	140	—	—	140	—	—	ns
Enable rise/fall time	t_{Er},t_{Ef}	—	—	25	—	—	25	ns
Address set-up time (RS, R/W to E)	t_{AS}	0	—	—	0	—	—	ns
Address hold time	t_{AH}	10	—	—	10	—	—	ns
Data set-up time	t_{DS}	40	—	—	—	—	100	ns
Data hold time	t_H	10	—	—	10	—	—	ns

Conditions: Ta=25°C, VDD=5.0± 0.5V

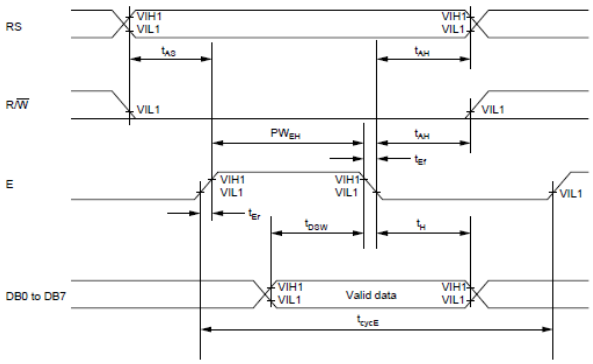


Figure 3: Write Timing Waveform

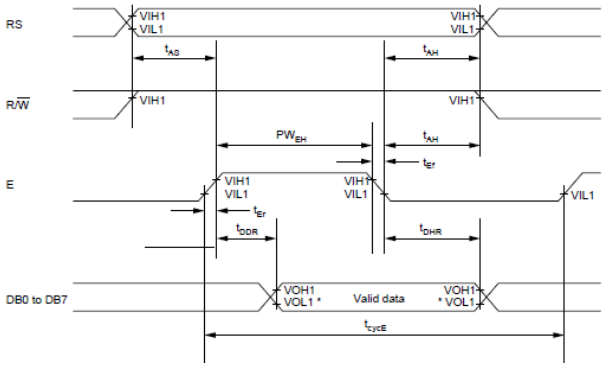


Figure 4: Read Timing Waveform

Initialization

Before beginning any application, it is recommended that all display settings be initialized. Below are algorithms for initializing the display in both 8-bit and 4-bit communication modes.

Before the first wait condition, please allow Vcc to rise to 2.7V then wait 40ms. During the three function set commands that follow, note that the busy flag cannot be checked; it becomes available in the last block. The unit will always expect a total of 8 bits to be sent, so note the structure used in four bit mode. The last initialization block will set the number of lines and character font as specified, turn the display off, issue the display clear command, and finally set the entry mode as desired.

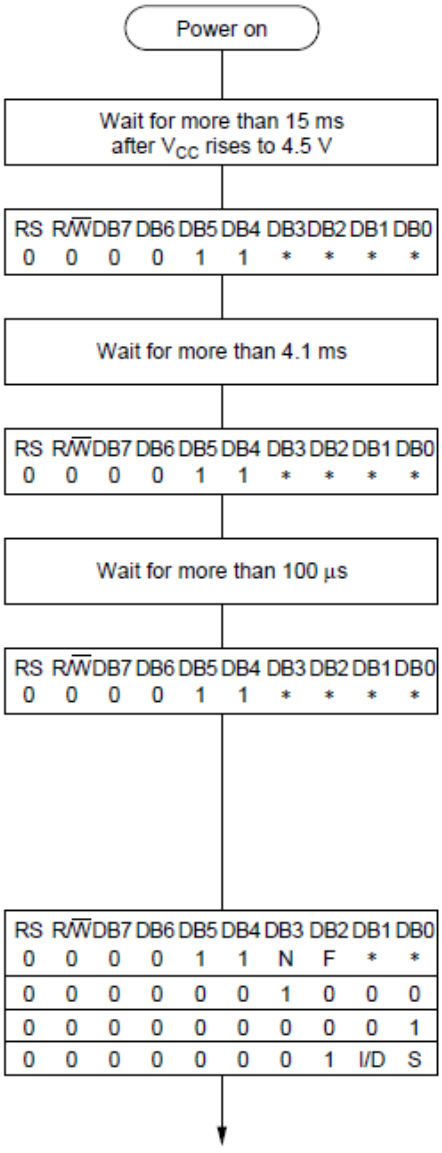


Figure 5: 8-bit Initialization

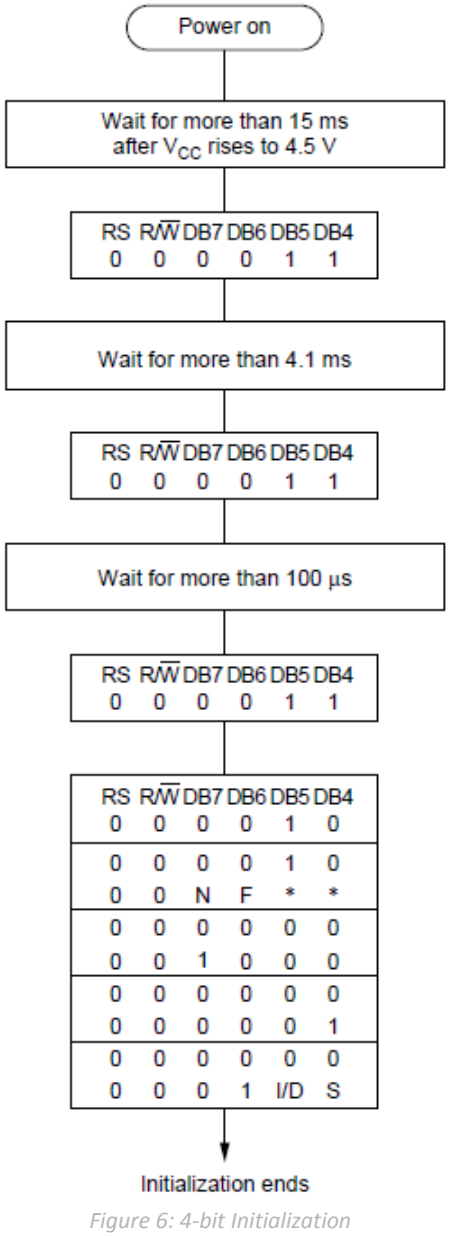


Figure 6: 4-bit Initialization

Note: * Indicates do not care condition.

Specifications

Electrical

Table 10: Electrical Characteristics

Item	Symbol	Min	Typ	Max	Unit
Supply Voltage For Logic	V_{DD}	4.5	5.0	5.5	V
Supply Voltage For LCD (Contrast)	V_0	-13.5	—	V_{DD}	V
Input High Voltage	V_{IH}	0.7 V_{DD}	—	V_{DD}	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.3 V_{DD}	V
Supply Current (V_{DD}=5V)	I_{DD}	0.7	1.75	1.5	mA
Supply Voltage of Red Backlight (36 Die)	V_{LED}	3.5	3.9	4.1	V
Supply Current of Red Backlight (36Die)	I_{LED}	0	—	180	mA
Supply Voltage of Yellow-Green Backlight (36 Die)	V_{LED}	4.0	4.2	4.4	V
Supply Current of Yellow-Green Backlight (36 Die)	I_{LED}	0	—	180	mA
Supply Voltage of White Backlight (2 Die)	V_{LED}	3.8	4.0	4.2	V
Supply Current of White Backlight (2 Die)	I_{LED}	0	—	30	mA

Optical

Table 11: Display Characteristics

Item	Dimension	Unit
Number of Characters	20 Characters x 4 Lines	—
Module dimension	98.0 x 60.0 x 15.0	mm
View area	76.0 x 25.2	mm
Active area	70.40 x 20.80	mm
Character size	2.95 x 4.75	mm
Character pitch	3.55 x 5.35	mm
Dot size	0.55 x 0.55	mm
Dot pitch	0.60 x 0.60	mm
LCD type	STN	
Duty	1/16	
View direction	12 o'clock	

Table 12: Viewing Characteristics

Item	Symbol	Min	Typ	Max	Unit
View Angle	(V)θ	-20	—	35	deg
	(H)ϕ	-30	—	30	deg
Contrast Ratio	CR	—	3	—	—
Response Time	T rise	—	—	250	ms
	T fall	—	—	250	ms

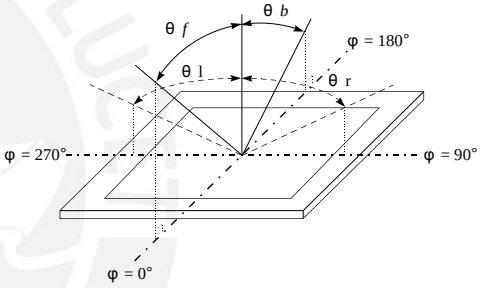


Figure 7: Viewing Angle Definition

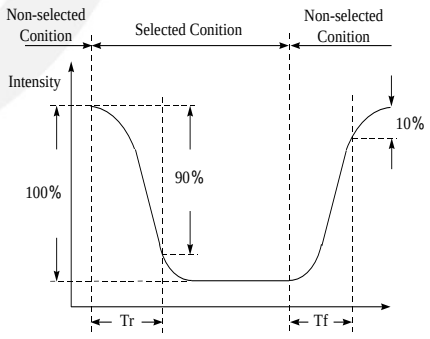


Figure 8: Display Response Time

Environmental

Table 13: Environmental Specifications

Item	Symbol	Min	Max	Unit
Operating Temp.	Top	-20	70	℃
Storage Temp.	Tstr	-30	80	℃

Note: Maximum 90% non-condensing humidity.

Troubleshooting

Power

For your MOP Display to function correctly, appropriate power must be applied, often as indicated by the backlight illuminating or a darkening of the character spaces. Please refer to the power diagram below and reference all voltages to the specifications provided.

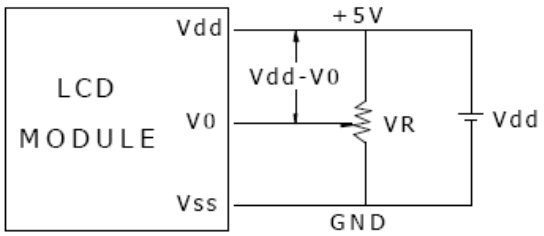


Figure 9: Single Supply Configuration

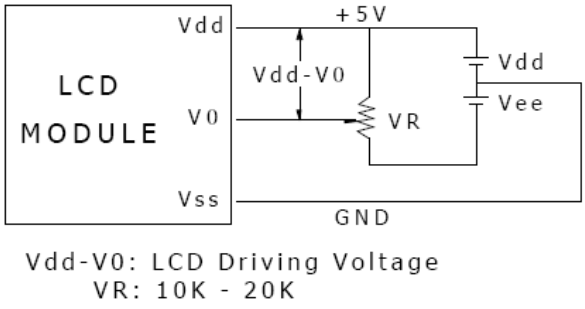


Figure 10: Dual Supply Configuration

Display

If your display is powered successfully, the backlight or contrast should be evident. A lack of text could be the result of a high contrast voltage, lower V_0. Also, ensure the expected DDRAM addresses are shown by moving the display to the home position.

Communication

When communication of either text or commands is interrupted, check all data and control pins for continuity. Ensure the display has been initialized correctly before sending information using the appropriate initialization algorithm. For 4-bit mode ensure D4-D7 are used. Finally, slow down communication and refer to timing diagrams and specifications for proper control flow.

Precautions

- Do not make extra holes on the display, modify its shape, or change the components.
- Avoid applying excessive electrical shock to the module.
- Do not drop, bend, twist, or disassemble the display.
- Avoid operation outside absolute maximum ratings.
- Solder only to the I/O terminals provided.
- Store in an anti-static container within a clean environment.

Ordering

Part Numbering Scheme

Table 14: Parallel Part Numbering Scheme

MOP	-	A	L	20	4	A	-	B	G	F	W	-	2	5	E	-	3	I	N
1	-	2	3	4	5	6	-	7	8	9	10	-	11	12	13	-	14	15	16

Options

Table 15: Parallel Part Options

#	Designator	Options
1	Product Line	MOP: Matrix Orbital Parallel Display
2	Display Type	A: Alphanumeric
3	Screen Type	L: Liquid Crystal Display
4	Display Columns	8: Eight Character Columns 16: Sixteen Character Columns 20: Twenty Character Columns 24: Twenty-Four Character Columns 40: Forty Character Columns
5	Display Rows	2: Two Character Rows 4: Four Character Rows
6	Display Form Factor	A: A Form Factor B: B Form Factor C: C Form Factor F: F Form Factor
7	IC Package	B: Chip on Board
8	LCD Glass Type	B: STN Positive Blue F: FFSTN Negative G: STN Positive Grey T: FSTN Negative W: FSTN Positive Y: STN Positive Yellow
9	Polarizer Style	F: Transflective T: Transmissive
10	Backlight Colour	R: Red Y: Yellow-Green W: White
11	Viewing Angle	1: 6:00 2: 12:00
12	Controller	5: S6A0069 Compatible
13	Character Set	E: European J: Japanese
14	Input Voltage	3: 5.0V
15	Temperature Range	I: Industrial
16	Negative Voltage Generation	N: None Provided

Contact

Sales
Phone: 403.229.2737
Email: sales@matrixorbital.ca

Support
Phone: 403.204.3750
Email: support@matrixorbital.ca

Online
Purchasing: www.matrixorbital.com
Support: www.matrixorbital.ca

WORLD-BEAM® Q12

Miniature self-contained photoelectric sensors in universal housing

BANNER®

more sensors more solutions

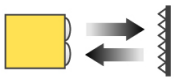
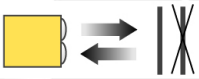
Features

Standard Model **Chemical-Resistant Model**

- Bright, visible red (640 nm) light source
- Standard models available with 4-wire 2 m (6.5') or 9 m (30') cable or 3 or 4-wire 150 mm (6") pigtail with Pico-style M8 threaded connector
- Solid-state, bipolar outputs: one current sourcing (PNP) and one current sinking (NPN) standard on 4-wire models
- Single output solid-state PNP or NPN standard on Q3 models
- Light Operate (L.O.) or Dark Operate (D.O.), depending on model
- Models available with PFA chemical-resistant jacket (1200 psi washdown rated) for use in harsh environments (see *Chemical-Resistant Models* on page 3).
- Compact 8 mm (0.31") housing mounts almost anywhere
- Crosstalk-avoidance circuitry for multiple-sensor applications
- LED status indicators for Power ON, Output Overload, Signal Received, and Marginal Signal

Standard Models

Sensing Mode		Model*	Range	Output
Opposed	640 nm Visible Red	**Q126E (emitter)**	2 m (6.5')	N/A
	Effective Beam: 5.7 mm (0.22") 	**Q12AB6R**		Bipolar LO
		Q12RB6R		Bipolar DO
		Q12AP6RQ3		1 PNP LO
		Q12RP6RQ3		1 PNP DO
		Q12AN6RQ3		1 NPN LO
		Q12RN6RQ3		1 NPN DO
Polarized Retro	640 nm Visible Red 	**Q12AB6LP**	1 m** (40")	Bipolar LO
		Q12RB6LP		Bipolar DO
		Q12AP6LPQ3		1 PNP LO
		Q12RP6LPQ3		1 PNP DO
		Q12AN6LPQ3		1 NPN LO
		Q12RN6LPQ3		1 NPN DO

Sensing Mode		Model*	Range	Output
Retro	640 nm Visible Red 	**Q12AB6LV**	1.5 m** (59")	Bipolar LO
		Q12RB6LV		Bipolar DO
		Q12AP6LVQ3		1 PNP LO
		Q12RP6LVQ3		1 PNP DO
		Q12AN6LVQ3		1 NPN LO
		Q12RN6LVQ3		1 NPN DO
Fixed-Field	Performance based on use of 90% reflectance white test card.			
	640 nm Visible Red 	**Q12AB6FF15**	15 mm (0.6") cutoff; 10 mm (0.4") focus	Bipolar LO
		Q12RB6FF15		Bipolar DO
		Q12AP6FF15Q3		1 PNP LO
		Q12RP6FF15Q3		1 PNP DO
		Q12AN6FF15Q3		1 NPN LO
		Q12RN6FF15Q3		1 NPN DO
		Q12AB6FF30	30 mm (1.2") cutoff; 16 mm (0.63") focus	Bipolar LO
		Q12RB6FF30		Bipolar DO
		Q12AP6FF30Q3		1 PNP LO
		Q12RP6FF30Q3		1 PNP DO
		Q12AN6FF30Q3		1 NPN LO
		Q12RN6FF30Q3		1 NPN DO
		Q12AB6FF50	50 mm (2") cutoff 16 mm (0.63") focus	Bipolar LO
		Q12RB6FF50		Bipolar DO
		Q12AP6FF50Q3		1 PNP LO
		Q12RP6FF50Q3		1 PNP DO
		Q12AN6FF50Q3		1 NPN LO
		Q12RN6FF50Q3		1 NPN DO

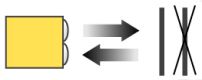
* **Q3 models:** 3-pin Pico-style (M8 threaded) 150 mm (6") pigtail QD. Not available for bipolar models.

Models with no suffix have standard 2 m (6.5') cables.

- For 9 m (30') cable, add suffix "**W/30**" to the model number (e.g., **Q126E W/30**).
- For 4-pin Pico-style (M8 threaded) 150 mm (6") pigtail QD, add suffix **Q** to the model number (e.g. **Q126EQ**).
- For 4-pin Euro-style (M12 threaded) 150 mm (6") pigtail QD, add suffix **Q5** to the model number (e.g. **Q126EQ5**).

Retroreflective range is specified using one model **BRT-60X40C retroreflector. Actual sensing range may be more or less than specified, depending upon efficiency and reflective area of the retroreflector(s) used.

Chemical-Resistant Models

Sensing Mode		Model*	Range	Output
Opposed	640 nm Visible Red	Q126ECR	1.5 m (4.9')	N/A
	Effective Beam: 5.7 mm (0.22") 	Q12AB6RCR		Bipolar LO
		Q12RB6RCR		Bipolar DO
Fixed-Field	Performance based on use of 90% reflectance white test card.			
	640 nm Visible Red 	Q12AB6FF15CR	13 mm (0.5") cutoff; 8 mm (0.3") focus	Bipolar LO
		Q12RB6FF15CR		Bipolar DO
		Q12AB6FF30CR	28 mm (1.1") cutoff; 14 mm (0.6") focus	Bipolar LO
		Q12RB6FF30CR		Bipolar DO
		Q12AB6FF50CR	48 mm (1.9") cutoff; 14 mm (0.6") focus	Bipolar LO
		Q12RB6FF50CR		Bipolar DO

*Only standard 2 m (6.5') cables are available for chemical-resistant models.

Indicator Features

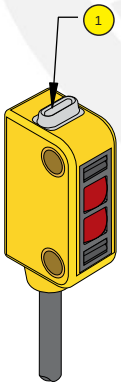


Figure 1. Features

- 1. Yellow and Green LEDs
  - Green ON steady: power to sensor is ON
  - Green flashing: output is overloaded
  - Yellow ON steady: received signal
  - Yellow flashing: marginal signal

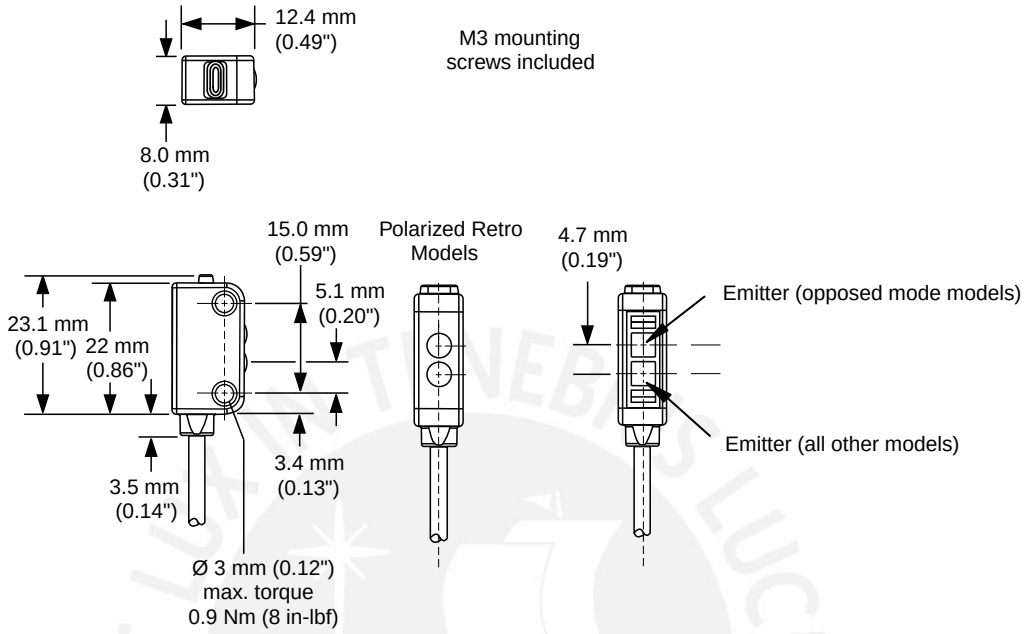
Chemical-Resistant models: LEDs are visible through translucent PFA jacket. Rated to 1200 psi washdown.

Specifications

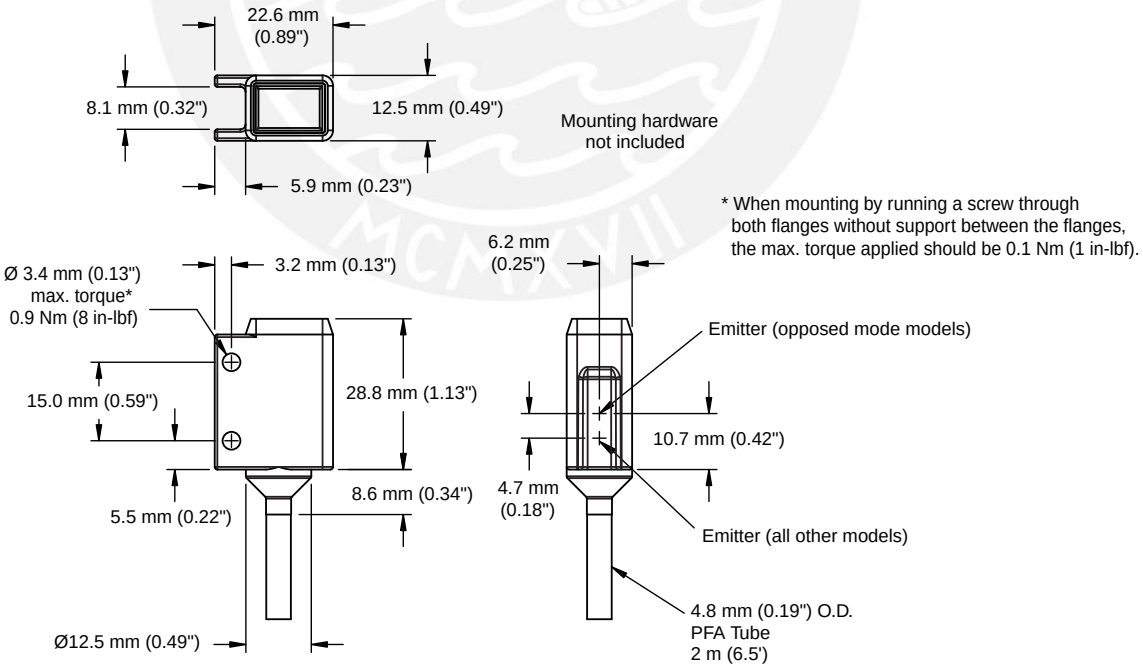
Feature	Description		
Sensing Beam	640 nm visible red		
Supply Voltage and Current	10 to 30V dc (10% max. ripple) @ 20 mA max current		
Supply Protection Circuitry	Protected against reverse polarity and transient voltages		
Output Configuration	Bipolar (1 NPN and 1 PNP) solid-state output or Single output (PNP or NPN), LO or DO, depending on model		
Output Ratings	50 mA total across all output(s) with overload and short circuit protection		
	OFF-state leakage current:	**NPN:** 200 µA	**PNP:** 10 µA
	ON-state saturation voltage:	**NPN:** 1.25V @ 50 mA	**PNP:** 1.45V @ 50 mA
Output Protection Circuitry	Protected against false pulse on power-up, short-circuit protected		
Output Response Time	**Opposed Mode:** 1.3 ms ON; 900 µs OFF		
	All Other Modes: 700 µs ON/OFF		
	NOTE: 120 ms delay on power-up; outputs do not conduct during this time.		
Repeatability	175 microseconds		
Switching Frequency	**Opposed Mode:** 385 Hz		
	All Other Modes: 715 Hz		
Indicators	One Yellow and one Green LED (see Figure 1)		
Construction	**Polarized Retro Models:** Thermoplastic elastomer housing with glass lens		
	All Other Standard Models: Thermoplastic elastomer housing with polycarbonate lens		
	Chemical-Resistant Models: Housing encased in PFA jacket; cable encased in 3/16" O.D. PFA tubing		
Environmental Rating	**Standard Models:** IEC IP67		
	Chemical-Resistant Models: IEC IP67 (NEMA6) and PW12 1200 psi washdown per NEMA ICS5, Annex F-2002		
Connections	**Standard Models:** 2 m (6.5') or 9 m (30') attached PVC cable, or 150 mm (6") pigtail with M8 or M12 threaded connection		
	Chemical-Resistant Models: 2 m (6.5') cable encased in 3/16" O.D. PFA tubing		
Operating Conditions	**Operating temperature:** -20° to +55° C (-4° to +131° F)		
	Storage temperature: -30° to +75° C (-22° to +167° F)		
	Relative humidity: 95% max @ +50° C (+122° F) non-condensing		
Certifications	CE cULus		

Dimensions

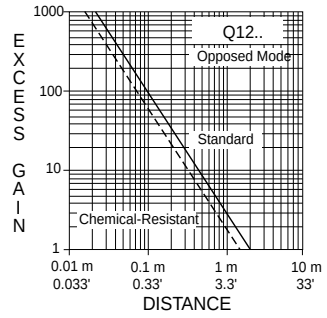
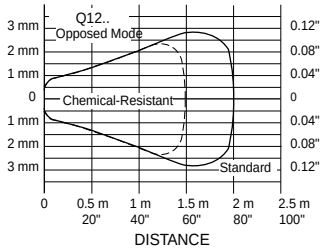
Standard Models



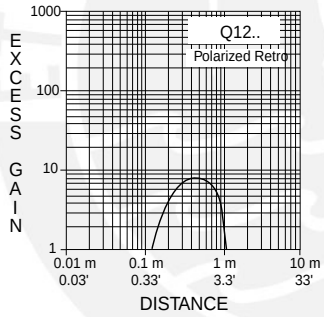
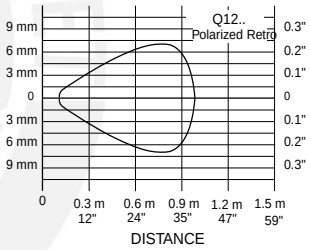
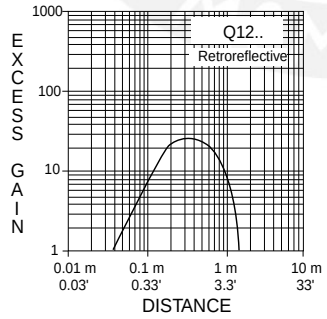
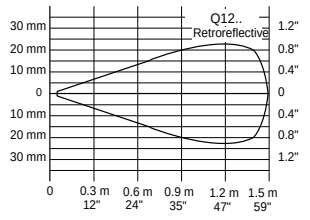
Chemical-Resistant Models



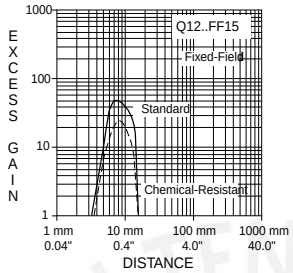
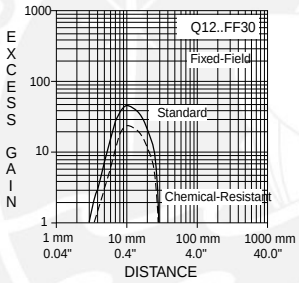
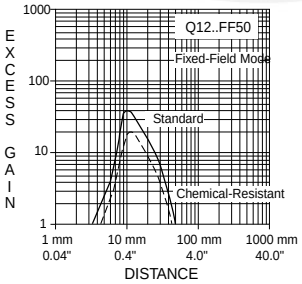
Performance Curves - Opposed Mode

	Excess Gain	Beam Pattern
Opposed		

Performance Curves - Retro Mode

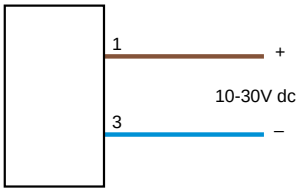
	Excess Gain	Beam Pattern
Polarized Retro	Performance based on use of a model **BRT-60X40C** retroreflector.	
		
Retro		

Performance Curves - Fixed-Field

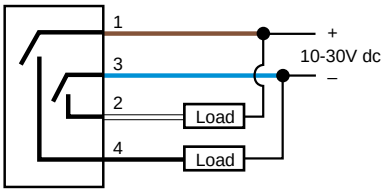
	Excess Gain
	Performance based on use of 90% reflectance white test card.*
Fixed-Field – 15 mm	 **Standard Models:** • Ø 0.4 mm spot size @ 10 mm focus • Ø 1.5 mm spot size @ 15 mm cutoff **Chemical-Resistant Models:** • Ø 0.4 mm spot size @ 8 mm focus • Ø 1.5 mm spot size @ 13 mm cutoff *** Using 18% gray test card:** cutoff distance will be 95% of value shown. *** Using 6% black test card:** cutoff distance will be 90% of value shown.
Fixed-Field – 30 mm	 **Standard Models:** • Ø 0.5 mm spot size @ 16 mm focus • Ø 3.0 mm spot size @ 30 mm cutoff **Chemical-Resistant Models:** • Ø 0.5 mm spot size @ 14 mm focus • Ø 3.0 mm spot size @ 28 mm cutoff *** Using 18% gray test card:** cutoff distance will be 90% of value shown. *** Using 6% black test card:** cutoff distance will be 80% of value shown.
Fixed-Field – 50 mm	 **Standard Models:** • Ø 0.5 mm spot size @ 16 mm focus • Ø 6.5 mm spot size @ 50 mm cutoff *** Using 18% gray test card:** cutoff distance will be 80% of value shown. *** Using 6% black test card:** cutoff distance will be 60% of value shown. **Chemical-Resistant Models:** • Ø 0.5 mm spot size @ 14 mm focus • Ø 6.5 mm spot size @ 48 mm cutoff *** Using 18% gray test card:** cutoff distance will be 70% of value shown. *** Using 6% black test card:** cutoff distance will be 50% of value shown.
Focus and spot sizes are typical.	
Legend: ———— Standard models – – – – – Chemical-Resistant models	

Hookups

Emitters



Bipolar Models



Wiring Key:

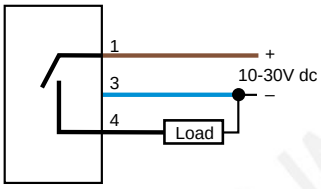
1 = Brown

2 = White

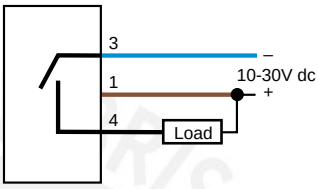
3 = Blue

4 = Black

PNP Models



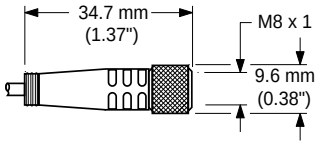
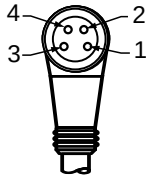
NPN Models



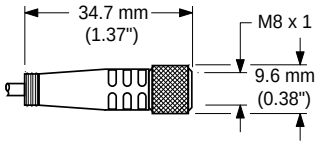
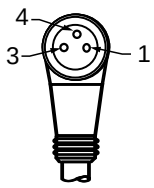
Cabled hookups only are shown. Hookups for QD models are functionally identical. (Emitters have no connection to black and white.)

NOTE: Please observe proper ESD precautions (grounding) when connecting QD models.

Quick-Disconnect (QD) Cordsets

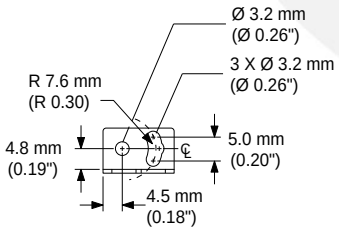
Style	Model	Length	Dimensions	Pinout
4-pin Pico-style straight with M8 threads	**PKG4M-2** **PKG4M-9**	2 m (6.5') 9 m (30')		Female  Wiring Key: 1 = Brown 2 = White 3 = Blue 4 = Black

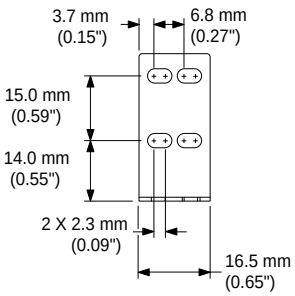
Quick-Disconnect (QD) Cordsets

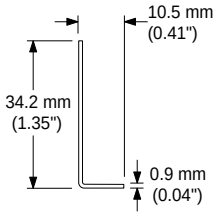
Style	Model	Length	Dimensions	Pinout
3-pin Pico-style straight with M8 threads	**PKG3M-2** **PKG3M-9**	2 m (6.5') 9 m (30')		Female  Wiring Key: 1 = Brown 3 = Blue 4 = Black

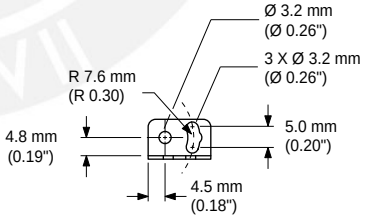
Mounting Brackets

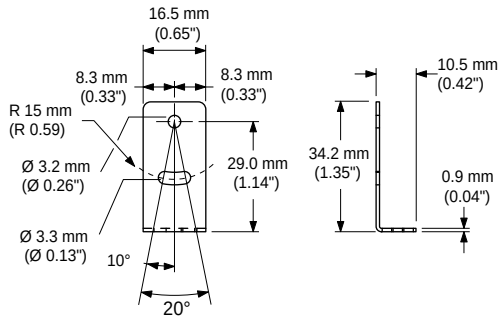
SMBQ12T		SMBQ12A	
• Right-angle bracket for use with standard Q12 models • 300 series stainless steel, 20 gauge		• Adjustable right-angle bracket for use with standard Q12 models • 300 series stainless steel, 20 gauge	











Apertures

Opposed-mode Q12 sensors (standard models only) may be fitted with apertures to narrow or shape the sensor's effective beam to more closely match the size or profile of the objects being sensed. A common example is the use of "line" (or "slot") type apertures to sense thread.

NOTE: The use of apertures will reduce the sensing range (see table below).

Model	Description		Reduced Sensor Range (Two Apertures Used)
APQ12-.5	Circular hole 	0.5 mm (0.02") diameter – 10 each	60 mm (2.4")
APQ12-1		1 mm (0.04") diameter – 10 each	190 mm (7.5")
APQ12-1.5		1.5 mm (0.06") diameter – 10 each	400 mm (15.7")
APQ12-2		2 mm (0.08") diameter – 10 each	725 mm (28.5")
APQ12-.5H	Horizontal slot	0.5 mm (0.02") – 10 each	350 mm (13.8")
APQ12-1H		1 mm (0.04") – 10 each	725 mm (28.5")
APQ12-.5V	Vertical slot	0.5 mm (0.02") – 10 each	450 mm (17.7")
APQ12-1V		1 mm (0.04") – 10 each	900 mm (35.4")
APQ12-4S	Protective jacket 	4 mm (0.16") square – 10 each	2000 mm (78.7")
APKQ12	Kit containing two of each aperture above – 18 total		—

WARNING . . . Not To Be Used for Personnel Protection

Never use this product as a sensing device for personnel protection. Doing so could lead to serious injury or death

This product does NOT include the self-checking redundant circuitry necessary to allow its use in personnel safety applications. A sensor failure or malfunction can cause either an energized or denergized sensor output condition. Consult your Banner Safety Products catalog for safety products that meet OSHA, ANSI and IEC standards for personnel protection.

BANNER
more sensors, more solutions

U-GAGE™ S18U Series Sensors with Analog Output

18 mm Ultrasonic Sensors with TEACH-mode programming

CE cULus

Features

- Fast, easy-to-use TEACH-Mode programming; no potentiometer adjustments
- Short dead zone
- Scalable output automatically distributes the output signal over the width of the programmed sensing window
- Two bi-colored status LEDs
- Rugged encapsulated design for harsh environments
- Choose 2 meter or 9 meter unterminated cable, or 5-pin Euro-style QD connector
- Wide operating range of -20° to +60°C (-4° to +140°F)
- Choose either straight or right-angle housing
- Temperature compensation
- Selectable response times of 2.5 or 30 ms
- Select analog models with either 0-10V dc or 4-20 mA output

Ultrasonic

Models

Model Number	Sensing Range	Cable*	Supply Voltage	Output	Housing Configuration
S18UUA	30 to 300 mm (1.2" to 11.8")	5-wire, 2 m (6.5') cable	10 to 30V dc	0 to 10V dc	Straight
S18UUAQ		5-pin Euro style QD			
S18UIA		5-wire, 2 m (6.5') cable		4 to 20 mA	
S18UIAQ		5-pin Euro style QD			
S18UUAR		5-wire, 2 m (6.5') cable		0 to 10V dc	Right-Angle
S18UUARQ		5-pin Euro style QD			
S18UIAR		5-wire, 2 m (6.5') cable		4 to 20 mA	
S18UIARQ		5-pin Euro style QD			

* 9 m cables are available by adding suffix "**W/30**" to the model number of any cabled sensor (e.g., **S18UUA W/30**). A model with a QD connector requires a mating cable; see page 10.

Information about discrete models is available on Banner's website: www.bannerengineering.com

WARNING . . . Not To Be Used for Personnel Protection

Never use these products as sensing devices for personnel protection. Doing so could lead to serious injury or death. These sensors do NOT include the self-checking redundant circuitry necessary to allow their use in personnel safety applications. A sensor failure or malfunction can cause either an energized or de-energized sensor output condition. Consult your current Banner Safety Products catalog for safety products which meet OSHA, ANSI and IEC standards for personnel protection.

Printed in USA 02/05 P/N 110738 rev. A

U-GAGE™ S18U Series Sensor — Analog Output

Principles of Operation

Ultrasonic sensors emit one or multiple pulses of ultrasonic energy, which travel through the air at the speed of sound. A portion of this energy reflects off the target and travels back to the sensor. The sensor measures the total time required for the energy to reach the target and return to the sensor. The distance to the object is then calculated using the following formula:

$$D = \frac{ct}{2}$$

D = distance from the sensor to the target
c = speed of sound in air
t = transit time for the ultrasonic pulse

To improve accuracy, an ultrasonic sensor may average the results of several pulses before outputting a new value.

Temperature Effects

The speed of sound is dependent upon the composition, pressure and temperature of the gas in which it is traveling. For most ultrasonic applications, the composition and pressure of the gas are relatively fixed, while the temperature may fluctuate.

In air, the speed of sound varies with temperature according to the following approximation:

$$C_{m/s} = 20\sqrt{273 + T_C}$$

$C_{m/s}$ = speed of sound in meters per second
T_C = temperature in °C

Or, in English units:

$$C_{ft/s} = 49\sqrt{460 + T_F}$$

$C_{ft/s}$ = speed of sound in feet per second
T_F = temperature in °F

Temperature Compensation
Changes in air temperature affect the speed of sound, which in turn affects the distance reading measured by the sensor. An increase in air temperature shifts both sensing window limits closer to the sensor. Conversely, a decrease in air temperature shifts both limits farther away from the sensor. This shift is approximately 3.5% of the limit distance for a 20° C change in temperature.

The S18U series ultrasonic sensors are temperature compensated. This reduces the error due to temperature by about 90%. The sensor will maintain its window limits to within 1.8% over the -20° to +60° C range.

NOTES:

- Exposure to direct sunlight can affect the sensor's ability to accurately compensate for changes in temperature.
- If the sensor is measuring across a temperature gradient, the compensation will be less effective.
- The temperature warmup drift upon power-up is less than 1.7% of the sensing distance. After 10 minutes, the apparent distance will be within 0.3% of the actual position. After 25 minutes, the sensing distance will be stable.

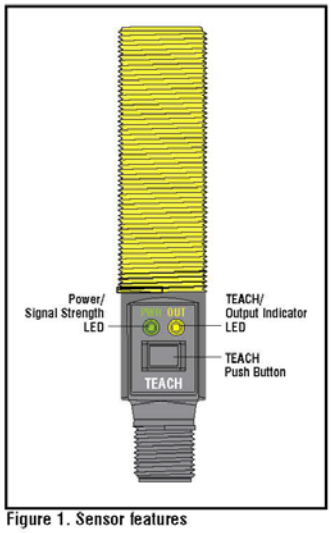


Figure 1. Sensor features

Sensor Programming

Two TEACH methods may be used to program the sensor:

- Teach individual minimum and maximum limits, or
- Use Auto-Window feature to center a sensing window around the taught position.

The sensor may be programmed either via its push button, or via a remote switch. Remote programming also may be used to disable the push button, preventing unauthorized personnel from adjusting the programming settings. To access this feature, connect the gray wire of the sensor to 0 - 2V dc, with a remote programming switch between the sensor and the voltage.

NOTE: The impedance of the Remote Teach input is 12 kΩ.

Programming is accomplished by following the sequence of input pulses (see programming procedures starting on page 4). The duration of each pulse (corresponding to a push button "click"), and the period between multiple pulses, are defined as "T":

0.04 seconds < T < 0.8 seconds

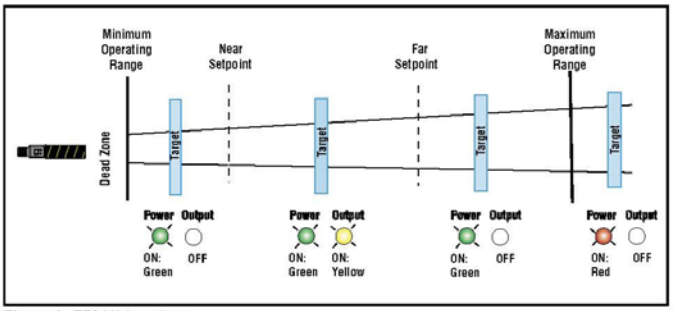


Figure 2. TEACH Interface

Status Indicators

Power ON/OFF LED	Indicates
OFF	Power is OFF.
ON Red	Target is weak or outside sensing range.
ON Green	Sensor is operating normally, good target.

Output/Teach LED	Indicates
OFF	Target is outside window limits.
Yellow	Target is within window limits.
ON Red (solid)	In Teach Mode, waiting for first limit.
ON Red (flashing)	In Teach Mode, waiting for second limit.

U-GAGE™ S18U Series Sensor — Analog Output

Teaching Minimum and Maximum Limits

General Notes on Programming

- The sensor will return to Run mode if the first Teach condition is not registered within 120 seconds.
- After the first limit is taught, the sensor will remain in Program mode until the Teach sequence is finished.
- To exit Program mode without saving any changes, press and hold the programming push button > 2 seconds (before teaching the second limit). The sensor will revert to the last saved limits.

Analog Output Slope:

The U-GAGE S18U sensor may be programmed for either a positive or a negative output slope, based on which limit is taught first (see Figure 3). If the Near limit is taught first, the slope will be positive. If the Far limit is taught first, the slope will be negative. Banner's scalable output automatically distributes the output signal over the width of the programmed sensing window.

In the event of signal loss, the analog output goes to 3.6 mA or 0V dc, which may be used to trigger an alarm.

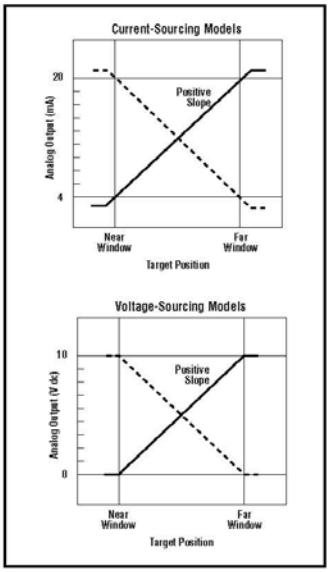


Figure 3. Analog output slope

	Procedure		Result
	Push Button 0.04 < "click" < 0.8 sec.	**Remote Wire** 0.04 sec. < T < 0.8 sec.	
Programming Mode	• Push and hold the push button	• No action required; sensor is ready for 1st limit teach	**Output LED:** ON Red **Power LED:** ON Green (good signal) or ON Red (no signal)
Teach First Limit	• Position the target for the first limit	• Position the target for the first limit	**Power LED:** Must be ON Green
	• "Click" the push button	• Single-pulse the remote line	**Teach Accepted** **(Sensor learns the 0V dc or 4 mA limit)** **Output LED:** Flashing Red **Teach Unacceptable** **Output LED:** ON Red
Teach Second Limit	• Position the target for the second limit	• Position the target for the second limit	**Power LED:** Must be ON Green
	• "Click" the push button	• Single-pulse the remote line	**Teach Accepted** **(Sensor learns the 10V dc or 20 mA limit)** **Output LED:** Yellow or OFF **Teach Unacceptable** **Output LED:** Flashing Red

Teaching Limits Using the Auto-Window Feature

Teaching the same limit twice for the same output automatically centers a 100 mm window on the taught position.

General Notes on Programming

- The sensor will return to Run mode if the first Teach condition is not registered within 120 seconds.
- After the first limit is taught, the sensor will remain in Program mode until the Teach sequence is finished.
- To exit Program mode without saving any changes, press and hold the programming push button > 2 seconds (before teaching the second limit). The sensor will revert to the last saved limits.
- Using this procedure the analog output will be centered on the taught position at approximately 5V dc or 12 mA.

	Procedure		Result
	Push Button **0.04 < "click" < 0.8 sec.**	**Remote Wire** **0.04 sec. < T < 0.8 sec.**	
Programming Mode	• Push and hold the push button 	• No action required; sensor is ready for first limit teach	**Output LED:** ON Red **Power LED:** ON Green (good signal) or ON Red (no signal)
Teach Limit	• Position the target for the center of the window	• Position the target for the center of the window	**Power LED:** Must be ON Green
	• "Click" the push button 	• Single-pulse the remote line 	**Teach Accepted** **Output LED:** Flashing Red **Teach Unacceptable** **Output LED:** ON Red
Re-Teach Limit	• Without moving the target, "click" the push button again 	• Without moving the target, single-pulse the remote line again 	**Teach Accepted** **Output LED:** Yellow or OFF **Teach Unacceptable** **Output LED:** Flashing Red

Push Button Lockout

Enables or disables the push button to prevent unauthorized adjustment of the program settings.

	Procedure		Result
	Push Button	Remote Wire 0.04 sec. < T < 0.8 sec.	
Enable/Disable Push Button	• Not available via push button	• Four-pulse the remote line 	• Push buttons are either enabled or disabled, depending on previous condition.

Specifications

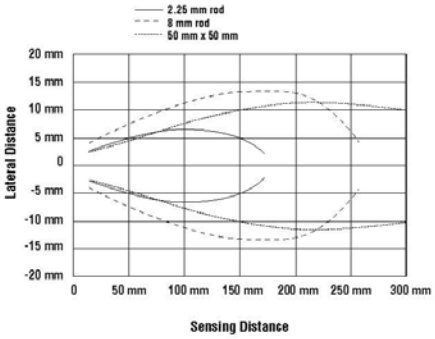
Sensing Range	30 to 300 mm (1.2" to 11.8")	
Supply Voltage	10 to 30V dc (10% maximum ripple); 65 mA max. (exclusive of load), 40 mA typical @ 25V input	
Ultrasonic Frequency	300 kHz, rep. rate 2.5 ms	
Supply Protection Circuitry	Protected against reverse polarity and transient voltages	
Output Configuration	**Analog Output:** 0 to 10V dc or 4 to 20 mA, depending on model	
Output Protection	Protected against short circuit conditions	
Output Ratings	**Analog Voltage Output:** 2.5 kΩ minimum load resistance Minimum supply for a full 10V output is 12V dc (for supply voltages between 10 and 12, V out max is at least V supply -2) **Analog Current Output:** 1 kΩ max @ 24V input Max load resistance = (Vcc-4)/0.02 ohms **For current output (4-20 mA) models,** ideal results are achieved when the total load resistance R = [(Vin – 3)/0.020]Ω. Example, at Vin = 24V dc, R ≈ 1 kΩ (1 watt). A worst-case shift of 1% of sensing distance is caused by operating the sensor at Vin = 30V dc and R = 0 Ω.	
Output Response Time (for a 95% step change)	**2.5 milliseconds:** Black wire at 5-30V dc **30 milliseconds:** Black wire at 0-2V dc (or open) Consult factory for other response speed options	
Delay at Power-Up	300 milliseconds	
Temperature Effect	0.02% of distance/ °C	
Linearity*	**2.5 ms response:** ±1 mm	**30 ms response:** ± 0.5 mm
Resolution*	**2.5 ms response:** 1 mm	**30 ms response:** 0.5 mm
Minimum Window Size	5 mm	
Adjustments	**Sensing window limits:** TEACH-Mode programming of near and far window limits may be set using the push button or remotely via TEACH input (see page 3).	
Indicators	**Range Indicator (Red/Green)**	**Green** — Target is within sensing range **Red** — Target is outside sensing range **OFF** — Sensing power is OFF
	Teach/Output Indicator (Yellow/Red)	**Yellow** — Target is within taught limits **OFF** — Target is outside taught window limits **Red** — Sensor is in TEACH mode
Remote TEACH Input	**Impedance:** 12 kΩ	
Construction	**Threaded Barrel:** Thermoplastic polyester **Push Button:** Santoprene	**Push Button Housing:** ABS/PC **Lightpipes:** Acrylic
Operating Conditions	**Temperature:** -20° to +60° C (-4° to +140° F)	**Maximum relative humidity:** 100%
Connections	2 m (6.5') or 9 m (30') shielded 5-conductor (with drain) PVC jacketed attached cable or 5-pin Euro-style quick-disconnect (see page 10 for quick-disconnect cable options)	
Environmental Rating	Leakproof design is rated IEC IP67; NEMA 6P	
Vibration and Mechanical Shock	All models meet Mil. Std. 202F requirements method 201A (vibration: 10 to 60 Hz max., double amplitude 0.06", maximum acceleration 10G). Also meets IEC 947-5-2 requirements: 30G 11 ms duration, half sine wave.	
Temperature Warmup Drift	Less than 1.7% of sensing distance upon power-up (see Temperature Compensation, page 2)	
Application Notes	Objects passing inside the specified near limit may produce a false response.	
Certifications	CE c UL us	

* Linearity and resolution are specified using a 50 mm x 50 mm (2" x 2") aluminum plate at 22°C under fixed sensing conditions.

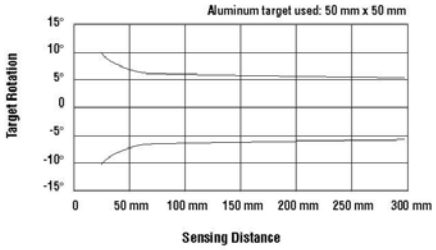
U-GAGE™ S18U Series Sensor — Analog Output

Sensor Response Curves

Effective Beam Pattern (Typical)

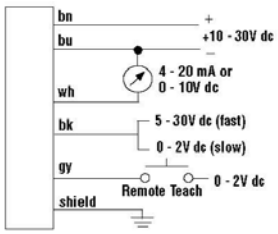


Maximum Target Rotation Angle

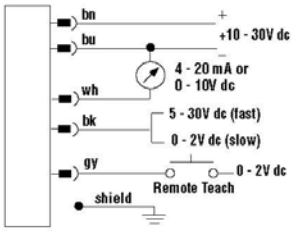


Hookups

Cabled Models



QD Models

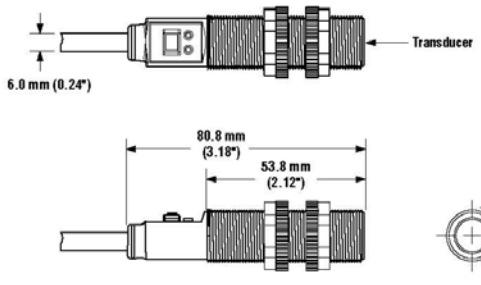


NOTE: It is recommended that the shield wire be connected to earth ground or DC common.

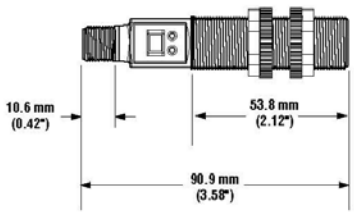
Dimensions

Straight Housing

Cabled Models

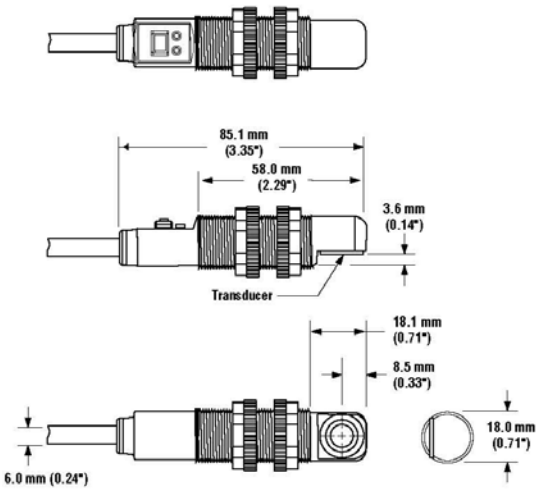


QD Models

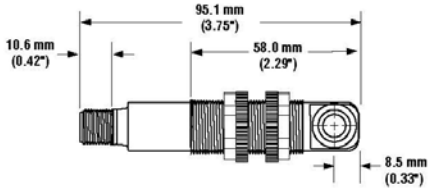


Right-Angle Housing

Cabled Models



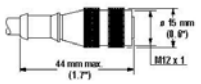
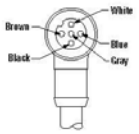
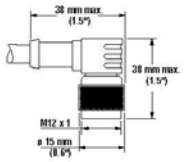
QD Models



U-GAGE™ S18U Series Sensor — Analog Output

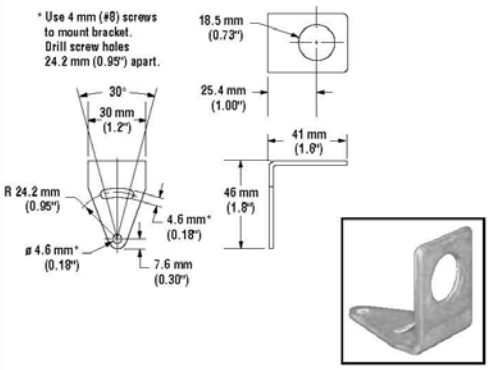
Accessories

Quick-Disconnect Cables

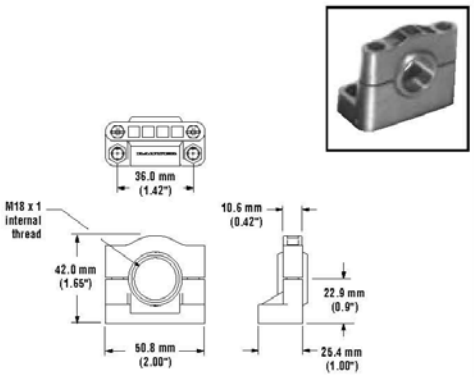
Style	Model	Length	Dimensions	Pinout
5-pin Euro-style straight, with shield	MQDEC2-506 MQDEC2-515 MQDEC2-530	2 m (6.5') 5 m (15') 9 m (30')		
5-pin Euro-style right-angle, with shield	MQDEC2-506RA MQDEC2-515RA MQDEC2-530RA	2 m (6.5') 5 m (15') 9 m (30')		

Mounting Brackets

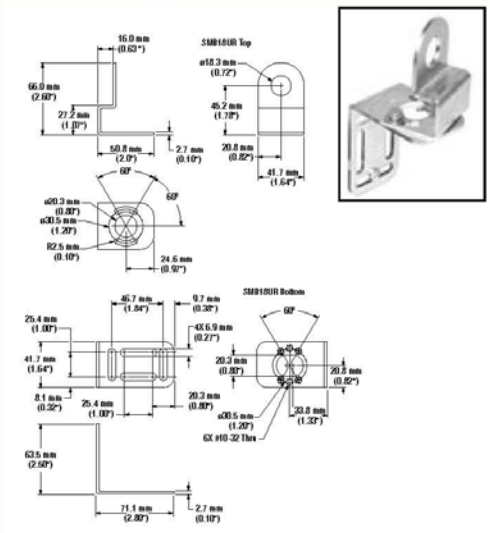
SMB18A	• 12-gauge, stainless steel, right-angle mounting bracket with a curved mounting slot for versatility and orientation • Clearance for M4 (#8) hardware



SMB18SF	• 18 mm swivel bracket • Black thermoplastic polyester • Includes stainless steel hardware



SMB18UR	• 2-piece universal 18 mm swivel bracket • 300 series stainless steel • Includes stainless steel swivel locking hardware



U-GAGE™ S18U Series Sensor — Analog Output

BANNER®
more sensors, more solutions

WARRANTY: Banner Engineering Corp. warrants its products to be free from defects for one year. Banner Engineering Corp. will repair or replace, free of charge, any product of its manufacture found to be defective at the time it is returned to the factory during the warranty period. This warranty does not cover damage or liability for the improper application of Banner products. This warranty is in lieu of any other warranty either expressed or implied.

P/N 110738 rev A.

Banner Engineering Corp., 9714 Tenth Ave. No., Minneapolis, MN USA 55441 • Phone: 763.544.3164 • www.bannerengineering.com • Email: sensors@bannerengineering.com

DIRECTIVE 2004/108/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 15 December 2004

on the approximation of the laws of the Member States relating to electromagnetic compatibility and repealing Directive 89/336/EEC

(Text with EEA relevance)

THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION,

Having regard to the Treaty establishing the European Community, and in particular Article 95 thereof,

Having regard to the proposal from the Commission,

Having regard to the opinion of the European Economic and Social Committee ([1]),

Acting in accordance with the procedure referred to in Article 251 of the Treaty ([2]),

Whereas:

(1) Council Directive 89/336/EEC of 3 May 1989 on the approximation of laws of the Member States relating to electromagnetic compatibility ([3]) has been the subject of a review under the initiative known as Simpler Legislation for the Internal Market (SLIM). Both the SLIM process and a subsequent in-depth consultation have revealed the need to complete, reinforce and clarify the framework established by Directive 89/336/EEC.

(2) Member States are responsible for ensuring that radiocommunications, including radio broadcast reception and the amateur radio service operating in accordance with International Telecommunication Union (ITU) radio regulations, electrical supply networks and telecommunications networks, as well as equipment connected thereto, are protected against electromagnetic disturbance.

(3) Provisions of national law ensuring protection against electromagnetic disturbance should be harmonised in order to guarantee the free movement of electrical and electronic apparatus without lowering justified levels of protection in the Member States.

(4) Protection against electromagnetic disturbance requires obligations to be imposed on the various economic operators. Those obligations should be applied in a fair and effective way in order to achieve such protection.

(5) The electromagnetic compatibility of equipment should be regulated with a view to ensuring the functioning of the internal market, that is to say, of an area without internal frontiers in which the free movement of goods, persons, services and capital is assured.

(6) The equipment covered by this Directive should include both apparatus and fixed installations. However, separate provision should be made for each. This is so because, whereas apparatus as such may move freely within the Community, fixed installations on the other hand are installed for permanent use at a predefined location, as assemblies of various types of apparatus and, where appropriate, other devices. The composition and function of such installations correspond in most cases to the particular needs of their operators.

(7) Radio equipment and telecommunications terminal equipment should not be covered by this Directive since they are already regulated by Directive 1999/5/EC of the European Parliament and of the Council of 9 March 1999 on radio equipment and telecommunications terminal equipment and the mutual recognition of their conformity ([4]). The electromagnetic compatibility requirements in both Directives achieve the same level of protection.

(8) Aircraft or equipment intended to be fitted into aircraft should not be covered by this Directive, since they are already subject to special Community or international rules governing electromagnetic compatibility.

(9) This Directive need not regulate equipment which is inherently benign in terms of electromagnetic compatibility.

(10) This Directive should not deal with the safety of equipment, since that is dealt with by separate Community or national legislation.

(11) Where this Directive regulates apparatus, it should refer to finished apparatus commercially available for the first time on the Community market. Certain components or sub-assemblies should, under certain conditions, be considered to be apparatus if they are made available to the end-user.

([1]) OJ C 220, 16.9.2003, p. 13.
([2]) Opinion of the European Parliament of 9 March 2004 (not yet published in the Official Journal) and Council Decision of 29 November 2004.
([3]) OJ L 139, 23.5.1989, p. 19. Directive as last amended by Directive 93/68/EEC (OJ L 220, 30.8.1993, p. 1).
([4]) OJ L 91, 7.4.1999, p. 10. Directive as amended by Regulation (EC) No 1882/2003 (OJ L 284, 31.10.2003, p. 1).

(12) The principles on which this Directive is based are those set out in the Council Resolution of 7 May 1985 on a new approach to technical harmonization and standards ([1]). In accordance with that approach, the design and manufacture of equipment is subject to essential requirements in relation to electromagnetic compatibility. Those requirements are given technical expression by harmonised European standards, to be adopted by the various European standardisation bodies, European Committee for Standardisation (CEN), European Committee for Electro-technical Standardisation (CENELEC) and European Telecommunications Standards Institute (ETSI). CEN, CENELEC and ETSI are recognised as the competent institutions in the field of this Directive for the adoption of harmonised standards, which they draw up in accordance with the general guidelines for cooperation between themselves and the Commission, and with the procedure laid down in Directive 98/34/EC of the European Parliament and of the Council of 22 June 1998 laying down a procedure for the provision of information in the field of technical standards and regulations and of rules on Information Society services ([2]).

(13) Harmonised standards reflect the generally acknowledged state of the art as regards electromagnetic compatibility matters in the European Union. It is thus in the interest of the functioning of the internal market to have standards for the electromagnetic compatibility of equipment which have been harmonised at Community level. Once the reference to such a standard has been published in the *Official Journal of the European Union*, compliance with it should raise a presumption of conformity with the relevant essential requirements, although other means of demonstrating such conformity should be permitted. Compliance with a harmonised standard means conformity with its provisions and demonstration thereof by the methods the harmonised standard describes or refers to.

(14) Manufacturers of equipment intended to be connected to networks should construct such equipment in a way that prevents networks from suffering unacceptable degradation of service when used under normal operating conditions. Network operators should construct their networks in such a way that manufacturers of equipment liable to be connected to networks do not suffer a disproportionate burden in order to prevent networks from suffering an unacceptable degradation of service. The European standardisation organisations should take due account of that objective (including the cumulative effects of the relevant types of electromagnetic phenomena) when developing harmonised standards.

(15) It should be possible to place apparatus on the market or put it into service only if the manufacturers concerned have established that such apparatus has been designed and manufactured in conformity with the requirements of this Directive. Apparatus placed on the market should bear the 'CE' marking attesting to compliance with this Directive. Although conformity assessment should be the responsibility of the manufacturer, without any need to involve an independent conformity assessment body, manufacturers should be free to use the services of such a body.

(16) The conformity assessment obligation should require the manufacturer to perform an electromagnetic compatibility assessment of apparatus, based on relevant phenomena, in order to determine whether or not it meets the protection requirements under this Directive.

(17) Where apparatus is capable of taking different configurations, the electromagnetic compatibility assessment should confirm whether the apparatus meets the protection requirements in the configurations foreseeable by the manufacturer as representative of normal use in the intended applications; in such cases it should be sufficient to perform an assessment on the basis of the configuration most likely to cause maximum disturbance and the configuration most susceptible to disturbance.

(18) Fixed installations, including large machines and networks, may generate electromagnetic disturbance, or be affected by it. There may be an interface between fixed installations and apparatus, and the electromagnetic disturbances produced by fixed installations may affect apparatus, and vice versa. In terms of electromagnetic compatibility, it is irrelevant whether the electromagnetic disturbance is produced by apparatus or by a fixed installation. Accordingly, fixed installations and apparatus should be subject to a coherent and comprehensive regime of essential requirements. It should be possible to use harmonised standards for fixed installations in order to demonstrate conformity with the essential requirements covered by such standards.

(19) Due to their specific characteristics, fixed installations need not be subject to the affixation of the 'CE' marking or to the declaration of conformity.

(20) It is not pertinent to carry out the conformity assessment of apparatus placed on the market for incorporation into a given fixed installation, and otherwise not commercially available, in isolation from the fixed installation into which it is to be incorporated. Such apparatus should therefore be exempted from the conformity assessment procedures normally applicable to apparatus. However, such apparatus should not be permitted to compromise the conformity of the fixed installation into which it is incorporated. Should apparatus be incorporated into more than one identical fixed installation, identifying the electromagnetic compatibility characteristics of these installations should be sufficient to ensure exemption from the conformity assessment procedure.

([1]) OJ C 136, 4.6.1985, p. 1.
([2]) OJ L 204, 21.7.1998, p. 37. Directive as last amended by the 2003 Act of Accession.

(21) A transitional period is necessary in order to ensure that manufacturers and other concerned parties are able to adapt to the new regulatory regime.

(22) Since the objective of this Directive, namely to ensure the functioning of the internal market by requiring equipment to comply with an adequate level of electromagnetic compatibility, cannot be sufficiently achieved by Member States and can therefore, by reason of its scale and effects, be better achieved at Community level, the Community may adopt measures, in accordance with the principle of subsidiarity as set out in Article 5 of the Treaty. In accordance with the principle of proportionality, as set out in that Article, this Directive does not go beyond what is necessary in order to achieve that objective.

(23) Directive 89/336/EEC should therefore be repealed,

HAVE ADOPTED THIS DIRECTIVE:

CHAPTER I

GENERAL PROVISIONS

Article 1

Subject matter and scope

1. This Directive regulates the electromagnetic compatibility of equipment. It aims to ensure the functioning of the internal market by requiring equipment to comply with an adequate level of electromagnetic compatibility. This Directive applies to equipment as defined in Article 2.

2. This Directive shall not apply to:

(a) equipment covered by Directive 1999/5/EC;

(b) aeronautical products, parts and appliances as referred to in Regulation (EC) No 1592/2002 of the European Parliament and of the Council of 15 July 2002 on common rules in the field of civil aviation and establishing a European Aviation Safety Agency ([1]);

(c) radio equipment used by radio amateurs within the meaning of the Radio Regulations adopted in the framework of the Constitution and Convention of the ITU ([2]), unless the equipment is available commercially. Kits of components to be assembled by radio amateurs and commercial equipment modified by and for the use of radio amateurs are not regarded as commercially available equipment.

3. This Directive shall not apply to equipment the inherent nature of the physical characteristics of which is such that:

(a) it is incapable of generating or contributing to electromagnetic emissions which exceed a level allowing radio and telecommunication equipment and other equipment to operate as intended; and

(b) it will operate without unacceptable degradation in the presence of the electromagnetic disturbance normally consequent upon its intended use.

4. Where, for the equipment referred to in paragraph 1, the essential requirements referred to in Annex I are wholly or partly laid down more specifically by other Community directives, this Directive shall not apply, or shall cease to apply, to that equipment in respect of such requirements from the date of implementation of those directives.

5. This Directive shall not affect the application of Community or national legislation regulating the safety of equipment.

Article 2

Definitions

1. For the purposes of this Directive, the following definitions shall apply:

(a) 'equipment' means any apparatus or fixed installation;

(b) 'apparatus' means any finished appliance or combination thereof made commercially available as a single functional unit, intended for the end user and liable to generate electromagnetic disturbance, or the performance of which is liable to be affected by such disturbance;

(c) 'fixed installation' means a particular combination of several types of apparatus and, where applicable, other devices, which are assembled, installed and intended to be used permanently at a predefined location;

(d) 'electromagnetic compatibility' means the ability of equipment to function satisfactorily in its electromagnetic environment without introducing intolerable electromagnetic disturbances to other equipment in that environment;

(e) 'electromagnetic disturbance' means any electromagnetic phenomenon which may degrade the performance of equipment. An electromagnetic disturbance may be electromagnetic noise, an unwanted signal or a change in the propagation medium itself;

([1]) OJ L 240, 7.9.2002, p. 1. Regulation as amended by Commission Regulation (EC) No 1701/2003 (OJ L 243, 27.9.2003, p. 5).

([2]) Constitution and Convention of the International Telecommunication Union adopted by the Additional Plenipotentiary Conference (Geneva, 1992) as amended by the Plenipotentiary Conference (Kyoto, 1994).

(f) 'immunity' means the ability of equipment to perform as intended without degradation in the presence of an electromagnetic disturbance;

(g) 'safety purposes' means the purposes of safeguarding human life or property;

(h) 'electromagnetic environment' means all electromagnetic phenomena observable in a given location.

2. For the purposes of this Directive the following shall be deemed to be an apparatus within the meaning of paragraph 1(b):

(a) 'components' or 'sub-assemblies' intended for incorporation into an apparatus by the end user, which are liable to generate electromagnetic disturbance, or the performance of which is liable to be affected by such disturbance;

(b) 'mobile installations' defined as a combination of apparatus and, where applicable, other devices, intended to be moved and operated in a range of locations.

Article 3

Placing on the market and/or putting into service

Member States shall take all appropriate measures to ensure that equipment is placed on the market and/or put into service only if it complies with the requirements of this Directive when properly installed, maintained and used for its intended purpose.

Article 4

Free movement of equipment

1. Member States shall not impede, for reasons relating to electromagnetic compatibility, the placing on the market and/or the putting into service in their territory of equipment which complies with this Directive.

2. The requirements of this Directive shall not prevent the application in any Member State of the following special measures concerning the putting into service or use of equipment:

(a) measures to overcome an existing or predicted electromagnetic compatibility problem at a specific site;

(b) measures taken for safety reasons to protect public telecommunications networks or receiving or transmitting stations when used for safety purposes in well-defined spectrum situations.

Without prejudice to Directive 98/34/EC, Member States shall notify those special measures to the Commission and to the other Member States.

The special measures which have been accepted shall be published by the Commission in the *Official Journal of the European Union*.

3. Member States shall not create any obstacles to the display and/or demonstration at trade fairs, exhibitions or similar events of equipment which does not comply with this Directive, provided that a visible sign clearly indicates that such equipment may not be placed on the market and/or put into service until it has been brought into conformity with this Directive. Demonstration may only take place provided that adequate measures are taken to avoid electromagnetic disturbances.

Article 5

Essential requirements

The equipment referred to in Article 1 shall meet the essential requirements set out in Annex I.

Article 6

Harmonised standards

1. 'Harmonised standard' means a technical specification adopted by a recognised European standardisation body under a mandate from the Commission in conformity with the procedures laid down in Directive 98/34/EC for the purpose of establishing a European requirement. Compliance with a 'harmonised standard' is not compulsory.

2. The compliance of equipment with the relevant harmonised standards whose references have been published in the *Official Journal of the European Union* shall raise a presumption, on the part of the Member States, of conformity with the essential requirements referred to in Annex I to which such standards relate. This presumption of conformity is limited to the scope of the harmonised standard(s) applied and the relevant essential requirements covered by such harmonised standard(s).

3. Where a Member State or the Commission considers that a harmonised standard does not entirely satisfy the essential requirements referred to in Annex I, it shall bring the matter before the Standing Committee set up by Directive 98/34/EC (hereinafter 'the Committee'), stating its reasons. The Committee shall deliver an opinion without delay.

4. Upon receipt of the Committee's opinion, the Commission shall take one of the following decisions with regard to the references to the harmonised standard concerned:

(a) not to publish;

(b) to publish with restrictions;

(c) to maintain the reference in the *Official Journal of the European Union*;

(d) to withdraw the reference from the *Official Journal of the European Union*.

The Commission shall inform the Member States of its decision without delay.

CHAPTER II

APPARATUS

Article 7

Conformity assessment procedure for apparatus

Compliance of apparatus with the essential requirements referred to in Annex I shall be demonstrated by means of the procedure described in Annex II (internal production control). However, at the discretion of the manufacturer or of his authorised representative in the Community, the procedure described in Annex III may also be followed.

Article 8

'CE' marking

1. Apparatus whose compliance with this Directive has been established by means of the procedure laid down in Article 7 shall bear the 'CE' marking which attests to that fact. The affixing of the 'CE' marking shall be the responsibility of the manufacturer or his authorised representative in the Community. The 'CE' marking shall be affixed in accordance with Annex V.

2. Member States shall take the necessary measures to prohibit the affixing to the apparatus, or to its packaging, or to the instructions for its use, of marks which are likely to mislead third parties in relation to the meaning and/or graphic form of the 'CE' marking.

3. Any other mark may be affixed to the apparatus, its packaging, or the instructions for its use, provided that neither the visibility nor the legibility of the 'CE' marking is thereby impaired.

4. Without prejudice to Article 10, if a competent authority establishes that the 'CE' marking has been unduly affixed, the manufacturer or his authorised representative in the Community shall bring the apparatus into conformity with the provisions concerning the 'CE' marking under conditions imposed by the Member State concerned.

Article 9

Other marks and information

1. Each apparatus shall be identified in terms of type, batch, serial number or any other information allowing for the identification of the apparatus.

2. Each apparatus shall be accompanied by the name and address of the manufacturer and, if he is not established within the Community, the name and address of his authorised representative or of the person in the Community responsible for placing the apparatus on the Community market.

3. The manufacturer shall provide information on any specific precautions that must be taken when the apparatus is assembled, installed, maintained or used, in order to ensure that, when put into service, the apparatus is in conformity with the protection requirements set out in Annex I, point 1.

4. Apparatus for which compliance with the protection requirements is not ensured in residential areas shall be accompanied by a clear indication of this restriction of use, where appropriate also on the packaging.

5. The information required to enable apparatus to be used in accordance with the intended purpose of the apparatus shall be contained in the instructions accompanying the apparatus.

Article 10

Safeguards

1. Where a Member State ascertains that apparatus bearing the 'CE' marking does not comply with the requirements of this Directive, it shall take all appropriate measures to withdraw the apparatus from the market, to prohibit its placing on the market or its putting into service, or to restrict the free movement thereof.

2. The Member State concerned shall immediately inform the Commission and the other Member States of any such measure, indicating the reasons and specifying, in particular, whether non-compliance is due to:

(a) failure to satisfy the essential requirements referred to in Annex I, where the apparatus does not comply with the harmonised standards referred to in Article 6;

(b) incorrect application of the harmonised standards referred to in Article 6;

(c) shortcomings in the harmonised standards referred to in Article 6.

3. The Commission shall consult the parties concerned as soon as possible, following which it shall inform the Member States whether or not it finds the measure to be justified.

4. Where the measure referred to in paragraph 1 is attributed to a shortcoming in harmonised standards, the Commission, after consulting the parties, shall, if the Member State concerned intends to uphold the measure, bring the matter before the Committee and initiate the procedure laid down in Article 6(3) and (4).

5. Where the non-compliant apparatus has been subject to the conformity assessment procedure referred to in Annex III, the Member State concerned shall take appropriate action in respect of the author of the statement referred to in Annex III, point 3, and shall inform the Commission and the other Member States accordingly.

Article 11

Decisions to withdraw, prohibit or restrict the free movement of apparatus

1. Any decision taken pursuant to this Directive to withdraw apparatus from the market, prohibit or restrict its placing on the market or its putting into service, or restrict the free movement thereof, shall state the exact grounds on which it is based. Such decisions shall be notified without delay to the party concerned, who shall at the same time be informed of the remedies available to him under the national law in force in the Member State in question and of the time limits to which such remedies are subject.

2. In the event of a decision as referred to in paragraph 1, the manufacturer, his authorised representative, or any other interested party shall have the opportunity to put forward his point of view in advance, unless such consultation is not possible because of the urgency of the measure to be taken as justified in particular with respect to public interest requirements.

Article 12

Notified bodies

1. Member States shall notify the Commission of the bodies which they have designated to carry out the tasks referred to in Annex III. When determining the bodies to be designated, Member States shall apply the criteria laid down in Annex VI.

Such notification shall state whether the bodies are designated to carry out the tasks referred to in Annex III for all apparatus covered by this Directive, and/or the essential requirements referred to in Annex I or whether the scope of designation is limited to certain specific aspects and/or categories of apparatus.

2. Bodies which comply with the assessment criteria established by the relevant harmonised standards shall be presumed to comply with the criteria set out in Annex VI covered by such harmonised standards. The Commission shall publish in the *Official Journal of the European Union* the references of those standards.

3. The Commission shall publish in the *Official Journal of the European Union* a list of notified bodies. The Commission shall ensure that the list is kept up to date.

4. If a Member State finds that a notified body no longer meets the criteria listed in Annex VI, it shall inform the Commission and the other Member States accordingly. The Commission shall withdraw the reference to that body from the list referred to in paragraph 3.

CHAPTER III

FIXED INSTALLATIONS

Article 13

Fixed installations

1. Apparatus which has been placed on the market and which may be incorporated into a fixed installation is subject to all relevant provisions for apparatus set out in this Directive.

However, the provisions of Articles 5, 7, 8 and 9 shall not be compulsory in the case of apparatus which is intended for incorporation into a given fixed installation and is otherwise not commercially available. In such cases, the accompanying documentation shall identify the fixed installation and its electromagnetic compatibility characteristics and shall indicate the precautions to be taken for the incorporation of the apparatus into the fixed installation in order not to compromise the conformity of that installation. It shall furthermore include the information referred to in Article 9(1) and (2).

2. Where there are indications of non-compliance of the fixed installation, in particular, where there are complaints about disturbances being generated by the installation, the competent authorities of the Member State concerned may request evidence of compliance of the fixed installation, and, when appropriate, initiate an assessment.

Where non-compliance is established, the competent authorities may impose appropriate measures to bring the fixed installation into compliance with the protection requirements set out in Annex I, point 1.

3. Member States shall set out the necessary provisions for identifying the person or persons responsible for the establishment of compliance of a fixed installation with the relevant essential requirements.

CHAPTER IV

FINAL PROVISIONS

Article 14

Repeal

Directive 89/336/EEC is hereby repealed as from 20 July 2007.

References to Directive 89/336/EEC shall be construed as references to this Directive and should be read in accordance with the correlation table set out in Annex VII.

Article 15

Transitional provisions

Member States shall not impede the placing on the market and/or the putting into service of equipment which is in compliance with the provisions of Directive 89/336/EEC and which was placed on the market before 20 July 2009.

Article 16

Transposition

1. Member States shall adopt and publish the laws, regulations and administrative provisions necessary to comply with this Directive by 20 January 2007. They shall forthwith inform the Commission thereof. They shall apply those provisions as from 20 July 2007. When Member States adopt those provisions, they shall contain a reference to this Directive or shall be accompanied by such reference on the occasion of their official publication. The methods of making such reference shall be laid down by Member States.

2. Member States shall communicate to the Commission the texts of the provisions of national law which they adopt in the field covered by this Directive.

Article 17

Entry into force

This Directive shall enter into force on the twentieth day after its publication in the *Official Journal of the European Union*.

Article 18

Addressees

This Directive is addressed to the Member States.

Done at Strasbourg, 15 December 2004.

For the European Parliament	*For the Council*
The President	*The President*
J. BORRELL FONTELLES	A. NICOLAÏ

ANNEX I

ESSENTIAL REQUIREMENTS REFERRED TO IN ARTICLE 5

1. **Protection requirements**

Equipment shall be so designed and manufactured, having regard to the state of the art, as to ensure that:

(a) the electromagnetic disturbance generated does not exceed the level above which radio and telecommunications equipment or other equipment cannot operate as intended;

(b) it has a level of immunity to the electromagnetic disturbance to be expected in its intended use which allows it to operate without unacceptable degradation of its intended use.

2. **Specific requirements for fixed installations**

Installation and intended use of components

A fixed installation shall be installed applying good engineering practices and respecting the information on the intended use of its components, with a view to meeting the protection requirements set out in Point 1. Those good engineering practices shall be documented and the documentation shall be held by the person(s) responsible at the disposal of the relevant national authorities for inspection purposes for as long as the fixed installation is in operation.

ANNEX II

CONFORMITY ASSESSMENT PROCEDURE REFERRED TO IN ARTICLE 7

(internal production control)

1. The manufacturer shall perform an electromagnetic compatibility assessment of the apparatus, on the basis of the relevant phenomena, with a view to meeting the protection requirements set out in Annex I, point 1. The correct application of all the relevant harmonised standards whose references have been published in the *Official Journal of the European Union* shall be equivalent to the carrying out of the electromagnetic compatibility assessment.
2. The electromagnetic compatibility assessment shall take into account all normal intended operating conditions. Where the apparatus is capable of taking different configurations, the electromagnetic compatibility assessment shall confirm whether the apparatus meets the protection requirements set out in Annex I, point 1, in all the possible configurations identified by the manufacturer as representative of its intended use.
3. In accordance with the provisions set out in Annex IV, the manufacturer shall draw up technical documentation providing evidence of the conformity of the apparatus with the essential requirements of this Directive.
4. The manufacturer or his authorised representative in the Community shall hold the technical documentation at the disposal of the competent authorities for at least ten years after the date on which such apparatus was last manufactured.
5. The compliance of apparatus with all relevant essential requirements shall be attested by an EC declaration of conformity issued by the manufacturer or his authorised representative in the Community.
6. The manufacturer or his authorised representative in the Community shall hold the EC declaration of conformity at the disposal of the competent authorities for a period of at least ten years after the date on which such apparatus was last manufactured.
7. If neither the manufacturer nor his authorised representative is established within the Community, the obligation to hold the EC declaration of conformity and the technical documentation at the disposal of the competent authorities shall lie with the person who places the apparatus on the Community market.
8. The manufacturer must take all measures necessary to ensure that the products are manufactured in accordance with the technical documentation referred to in point 3 and with the provisions of this Directive that apply to them.
9. The technical documentation and the EC declaration of conformity shall be drawn up in accordance with the provisions set out in Annex IV.

ANNEX III

CONFORMITY ASSESSMENT PROCEDURE REFERRED TO IN ARTICLE 7

1. This procedure consists of applying Annex II, completed as follows:
2. The manufacturer or his authorised representative in the Community shall present the technical documentation to the notified body referred to in Article 12 and request the notified body for an assessment thereof. The manufacturer or his authorised representative in the Community shall specify to the notified body which aspects of the essential requirements must be assessed by the notified body.
3. The notified body shall review the technical documentation and assess whether the technical documentation properly demonstrates that the requirements of the Directive that it is to assess have been met. If the compliance of the apparatus is confirmed, the notified body shall issue a statement to the manufacturer or his authorised representative in the Community confirming the compliance of the apparatus. That statement shall be limited to those aspects of the essential requirements which have been assessed by the notified body.
4. The manufacturer shall add the statement of the notified body to the technical documentation.

ANNEX IV

TECHNICAL DOCUMENTATION AND EC DECLARATION OF CONFORMITY

1. **Technical documentation**

The technical documentation must enable the conformity of the apparatus with the essential requirements to be assessed. It must cover the design and manufacture of the apparatus, in particular:

— a general description of the apparatus;

— evidence of compliance with the harmonised standards, if any, applied in full or in part;

— where the manufacturer has not applied harmonised standards, or has applied them only in part, a description and explanation of the steps taken to meet the essential requirements of the Directive, including a description of the electromagnetic compatibility assessment set out in Annex II, point 1, results of design calculations made, examinations carried out, test reports, etc.;

— a statement from the notified body, when the procedure referred to in Annex III has been followed.

2. **EC declaration of conformity**

The EC declaration of conformity must contain, at least, the following:

— a reference to this Directive,

— an identification of the apparatus to which it refers, as set out in Article 9(1),

— the name and address of the manufacturer and, where applicable, the name and address of his authorised representative in the Community,

— a dated reference to the specifications under which conformity is declared to ensure the conformity of the apparatus with the provisions of this Directive,

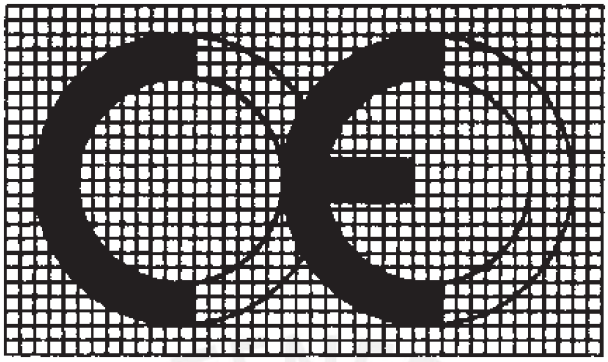
— the date of that declaration,

— the identity and signature of the person empowered to bind the manufacturer or his authorised representative.

ANNEX V

'CE' MARKING REFERRED TO IN ARTICLE 8

The 'CE' marking shall consist in the initials 'CE' taking the following form:



The 'CE' marking must have a height of at least 5 mm. If the 'CE' marking is reduced or enlarged the proportions given in the above graduated drawing must be respected.

The 'CE' marking must be affixed to the apparatus or to its data plate. Where this is not possible or not warranted on account of the nature of the apparatus, it must be affixed to the packaging, if any, and to the accompanying documents.

Where the apparatus is the subject of other Directives covering other aspects and which also provide for the 'CE' marking, the latter shall indicate that the apparatus also conforms with those other Directives.

However, where one or more of those Directives allow the manufacturer, during a transitional period, to choose which arrangements to apply, the 'CE' marking shall indicate conformity only with the Directives applied by the manufacturer. In that case, particulars of the Directives applied, as published in the *Official Journal of the European Union*, must be given in the documents, notices or instructions required by the Directives and accompanying such apparatus.

ANNEX VI

CRITERIA FOR THE ASSESSMENT OF THE BODIES TO BE NOTIFIED

1. The bodies notified by the Member States shall fulfil the following minimum conditions:
   (a) availability of personnel and of the necessary means and equipment;
   (b) technical competence and professional integrity of personnel;
   (c) independence in preparing the reports and performing the verification function provided for in this Directive;
   (d) independence of staff and technical personnel in relation to all interested parties, groups or persons directly or indirectly concerned with the equipment in question;
   (e) maintenance of professional secrecy by personnel;
   (f) possession of civil liability insurance unless such liability is covered by the Member State under national law.
2. Fulfilment of the conditions laid down in point 1 shall be verified at intervals by the competent authorities of the Member State.

ANNEX VII

CORRELATION TABLE

Directive 89/336/EEC	This Directive
Article 1, point 1	Article 2(1)(a), (b) and (c)
Article 1, point 2	Article 2(1)(e)
Article 1, point 3	Article 2(1)(f)
Article 1, point 4	Article 2(1)(d)
Article 1, points 5 and 6	-
Article 2(1)	Article 1(1)
Article 2(2)	Article 1(4)
Article 2(3)	Article 1(2)
Article 3	Article 3
Article 4	Article 5 and Annex I
Article 5	Article 4(1)
Article 6	Article 4(2)
Article 7(1)(a)	Article 6(1) and (2)
Article 7(1)(b)	-
Article 7(2).	-
Article 7(3)	-
Article 8(1)	Article 6(3) and (4)
Article 8(2)	-
Article 9(1)	Article 10(1) and (2)
Article 9(2)	Article 10(3) and (4)
Article 9(3)	Article 10(5)
Article 9(4)	Article 10(3)
Article 10(1), first sub-paragraph	Article 7, Annexes II and III
Article 10(1), second sub-paragraph	Article 8
Article 10(2)	Article 7, Annexes II and III
Article 10(3)	-
Article 10(4)	-
Article 10(5)	Article 7, Annexes II and III
Article 10(6)	Article 12
Article 11	Article 14
Article 12	Article 16
Article 13	Article 18
Annex I, point 1	Annex IV, point 2
Annex I, point 2	Annex V
Annex II	Annex VI
Annex III, last paragraph	Article 9(5)

SLOS094B – NOVEMBER 1970 – REVISED SEPTEMBER 2000

- **Short-Circuit Protection**
- **Offset-Voltage Null Capability**
- **Large Common-Mode and Differential Voltage Ranges**
- **No Frequency Compensation Required**
- **Low Power Consumption**
- **No Latch-Up**
- **Designed to Be Interchangeable With Fairchild μA741**

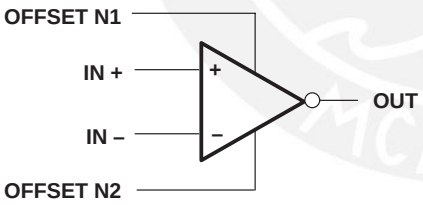
description

The μA741 is a general-purpose operational amplifier featuring offset-voltage null capability.

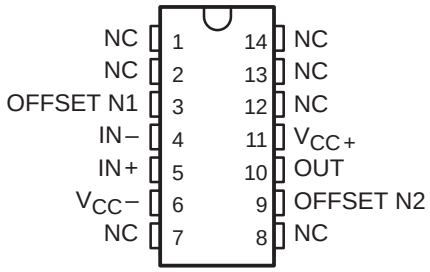
The high common-mode input voltage range and the absence of latch-up make the amplifier ideal for voltage-follower applications. The device is short-circuit protected and the internal frequency compensation ensures stability without external components. A low value potentiometer may be connected between the offset null inputs to null out the offset voltage as shown in Figure 2.

The μA741C is characterized for operation from 0°C to 70°C. The μA741I is characterized for operation from –40°C to 85°C.The μA741M is characterized for operation over the full military temperature range of –55°C to 125°C.

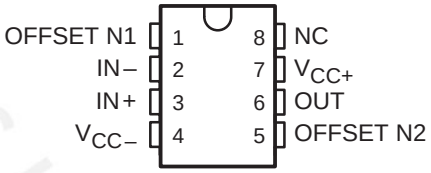
symbol



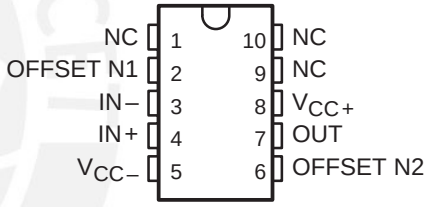
μA741M . . . J PACKAGE
(TOP VIEW)



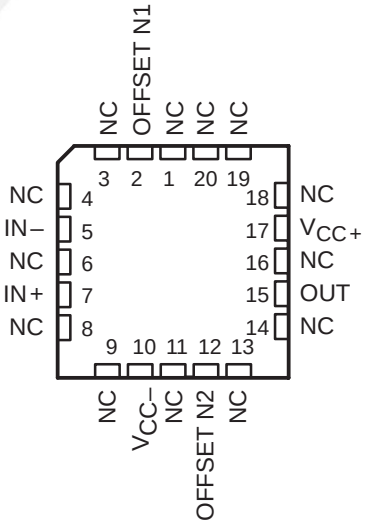
μA741M . . . JG PACKAGE
μA741C, μA741I . . . D, P, OR PW PACKAGE
(TOP VIEW)



μA741M . . . U PACKAGE
(TOP VIEW)



μA741M . . . FK PACKAGE
(TOP VIEW)



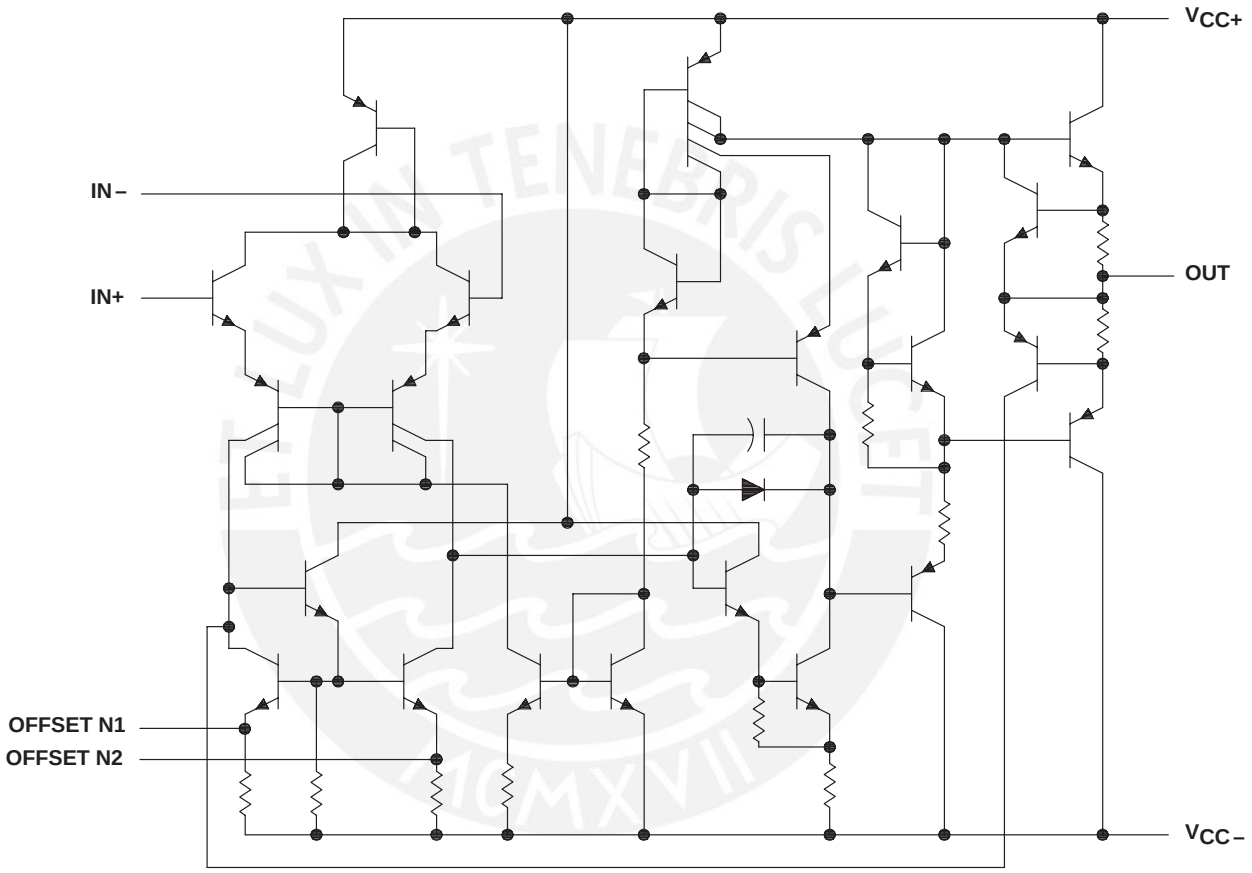
NC – No internal connection

AVAILABLE OPTIONS

T_A	PACKAGED DEVICES							CHIP FORM (Y)
	SMALL OUTLINE (D)	CHIP CARRIER (FK)	CERAMIC DIP (J)	CERAMIC DIP (JG)	PLASTIC DIP (P)	TSSOP (PW)	FLAT PACK (U)	
0°C to 70°C	µA741CD				µA741CP	µA741CPW		µA741Y
–40°C to 85°C	µA741ID				µA741IP			
–55°C to 125°C		µA741MFK	µA741MJ	µA741MJG			µA741MU	

The D package is available taped and reeled. Add the suffix R (e.g., µA741CDR).

schematic

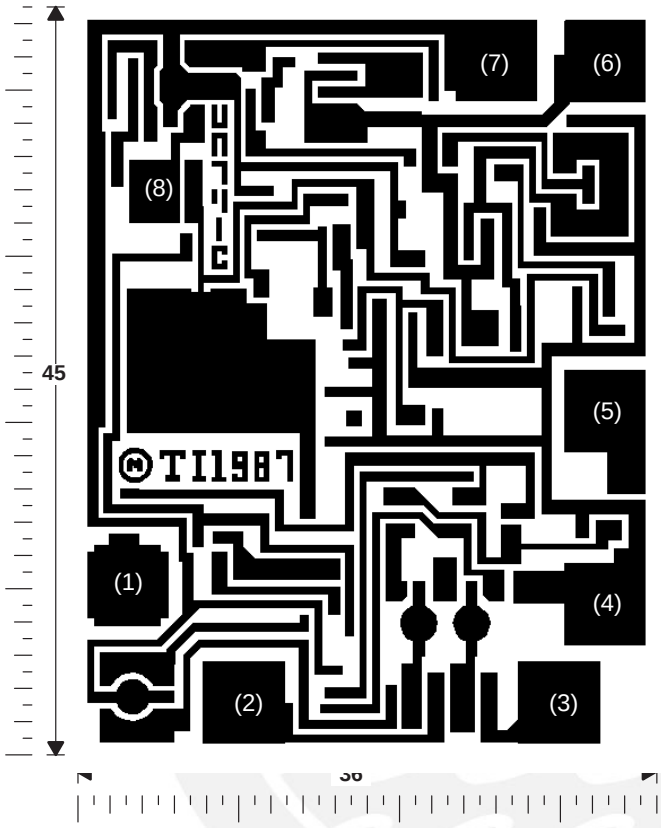


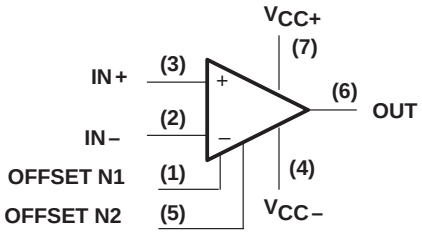
Component Count	
Transistors	22
Resistors	11
Diode	1
Capacitor	1

μA741Y chip information

This chip, when properly assembled, displays characteristics similar to the μA741C. Thermal compression or ultrasonic bonding may be used on the doped-aluminum bonding pads. Chips may be mounted with conductive epoxy or a gold-silicon preform.

BONDING PAD ASSIGNMENTS





CHIP THICKNESS: 15 TYPICAL

BONDING PADS: 4 × 4 MINIMUM

$T_J$max = 150°C.

TOLERANCES ARE ±10%.

ALL DIMENSIONS ARE IN MILS.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

		μA741C	μA741I	μA741M	UNIT
Supply voltage, V_{CC+} (see Note 1)		18	22	22	V
Supply voltage, V_{CC-} (see Note 1)		–18	–22	–22	V
Differential input voltage, V_{ID} (see Note 2)		±15	±30	±30	V
Input voltage, V_I any input (see Notes 1 and 3)		±15	±15	±15	V
Voltage between offset null (either OFFSET N1 or OFFSET N2) and V_{CC-}		±15	±0.5	±0.5	V
Duration of output short circuit (see Note 4)		unlimited	unlimited	unlimited	
Continuous total power dissipation		See Dissipation Rating Table			
Operating free-air temperature range, T_A		0 to 70	–40 to 85	–55 to 125	°C
Storage temperature range		–65 to 150	–65 to 150	–65 to 150	°C
Case temperature for 60 seconds	FK package			260	°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds	J, JG, or U package			300	°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	D, P, or PW package	260	260		°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values, unless otherwise noted, are with respect to the midpoint between V_{CC+} and V_{CC-}.
2. Differential voltages are at IN+ with respect to IN–.
3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 V, whichever is less.
4. The output may be shorted to ground or either power supply. For the μA741M only, the unlimited duration of the short circuit applies at (or below) 125°C case temperature or 75°C free-air temperature.

DISSIPATION RATING TABLE

PACKAGE	$T_A \le 25°C$ POWER RATING	DERATING FACTOR	DERATE ABOVE T_A	$T_A = 70°C$ POWER RATING	$T_A = 85°C$ POWER RATING	$T_A = 125°C$ POWER RATING
D	500 mW	5.8 mW/°C	64°C	464 mW	377 mW	N/A
FK	500 mW	11.0 mW/°C	105°C	500 mW	500 mW	275 mW
J	500 mW	11.0 mW/°C	105°C	500 mW	500 mW	275 mW
JG	500 mW	8.4 mW/°C	90°C	500 mW	500 mW	210 mW
P	500 mW	N/A	N/A	500 mW	500 mW	N/A
PW	525 mW	4.2 mW/°C	25°C	336 mW	N/A	N/A
U	500 mW	5.4 mW/°C	57°C	432 mW	351 mW	135 mW

electrical characteristics at specified free-air temperature, $V_{CC\pm}$ = ±15 V (unless otherwise noted)

PARAMETER		TEST CONDITIONS	T_A†	μA741C			μA741I, μA741M			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
V_{IO}	Input offset voltage	V_O = 0	25°C		1	6		1	5	mV
			Full range			7.5			6	
$\Delta V_{IO(adj)}$	Offset voltage adjust range	V_O = 0	25°C		±15			±15		mV
I_{IO}	Input offset current	V_O = 0	25°C		20	200		20	200	nA
			Full range			300			500	
I_{IB}	Input bias current	V_O = 0	25°C		80	500		80	500	nA
			Full range			800			1500	
V_{ICR}	Common-mode input voltage range		25°C	±12	±13		±12	±13		V
			Full range	±12			±12			
V_{OM}	Maximum peak output voltage swing	R_L = 10 kΩ	25°C	±12	±14		±12	±14		V
		R_L ≥ 10 kΩ	Full range	±12			±12			
		R_L = 2 kΩ	25°C	±10	±13		±10	±13		
		R_L ≥ 2 kΩ	Full range	±10			±10			
A_{VD}	Large-signal differential voltage amplification	R_L ≥ 2 kΩ	25°C	20	200		50	200		V/mV
		V_O = ±10 V	Full range	15			25			
r_i	Input resistance		25°C	0.3	2		0.3	2		MΩ
r_o	Output resistance	V_O = 0, See Note 5	25°C		75			75		Ω
C_i	Input capacitance		25°C		1.4			1.4		pF
CMRR	Common-mode rejection ratio	V_{IC} = V_{ICR}min	25°C	70	90		70	90		dB
			Full range	70			70			
k_{SVS}	Supply voltage sensitivity ($\Delta V_{IO}/\Delta V_{CC}$)	V_{CC} = ±9 V to ±15 V	25°C		30	150		30	150	μV/V
			Full range			150			150	
I_{OS}	Short-circuit output current		25°C		±25	±40		±25	±40	mA
I_{CC}	Supply current	V_O = 0, No load	25°C		1.7	2.8		1.7	2.8	mA
			Full range			3.3			3.3	
P_D	Total power dissipation	V_O = 0, No load	25°C		50	85		50	85	mW
			Full range			100			100	

† All characteristics are measured under open-loop conditions with zero common-mode input voltage unless otherwise specified. Full range for the μA741C is 0°C to 70°C, the μA741I is –40°C to 85°C, and the μA741M is –55°C to 125°C.

NOTE 5: This typical value applies only at frequencies above a few hundred hertz because of the effects of drift and thermal feedback.

operating characteristics, $V_{CC\pm}$ = ±15 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	μA741C			μA741I, μA741M			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_r	Rise time	V_I = 20 mV, R_L = 2 kΩ, C_L = 100 pF, See Figure 1		0.3			0.3		μs
	Overshoot factor			5%			5%		
SR	Slew rate at unity gain	V_I = 10 V, R_L = 2 kΩ, C_L = 100 pF, See Figure 1		0.5			0.5		V/μs

electrical characteristics at specified free-air temperature, $V_{CC\pm}$ = ±15 V, T_A = 25°C (unless otherwise noted)

PARAMETER		TEST CONDITIONS	μA741Y			UNIT
			MIN	TYP	MAX	
V_{IO}	Input offset voltage	V_O = 0		1	6	mV
$\Delta V_{IO(adj)}$	Offset voltage adjust range	V_O = 0		±15		mV
I_{IO}	Input offset current	V_O = 0		20	200	nA
I_{IB}	Input bias current	V_O = 0		80	500	nA
V_{ICR}	Common-mode input voltage range		±12	±13		V
V_{OM}	Maximum peak output voltage swing	R_L = 10 kΩ	±12	±14		V
		R_L = 2 kΩ	±10	±13		
A_{VD}	Large-signal differential voltage amplification	$R_L \geq$ 2 kΩ	20	200		V/mV
r_i	Input resistance		0.3	2		MΩ
r_o	Output resistance	V_O = 0, See Note 5		75		Ω
C_i	Input capacitance			1.4		pF
CMRR	Common-mode rejection ratio	$V_{IC} = V_{ICR}$min	70	90		dB
k_{SVS}	Supply voltage sensitivity ($\Delta V_{IO}/\Delta V_{CC}$)	V_{CC} = ±9 V to ±15 V		30	150	μV/V
I_{OS}	Short-circuit output current			±25	±40	mA
I_{CC}	Supply current	V_O = 0, No load		1.7	2.8	mA
P_D	Total power dissipation	V_O = 0, No load		50	85	mW

† All characteristics are measured under open-loop conditions with zero common-mode voltage unless otherwise specified.
NOTE 5: This typical value applies only at frequencies above a few hundred hertz because of the effects of drift and thermal feedback.

operating characteristics, $V_{CC\pm}$ = ±15 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	μA741Y			UNIT
			MIN	TYP	MAX	
t_r	Rise time	V_I = 20 mV, R_L = 2 kΩ, C_L = 100 pF, See Figure 1		0.3		μs
	Overshoot factor			5%		
SR	Slew rate at unity gain	V_I = 10 V, R_L = 2 kΩ, C_L = 100 pF, See Figure 1		0.5		V/μs

PARAMETER MEASUREMENT INFORMATION

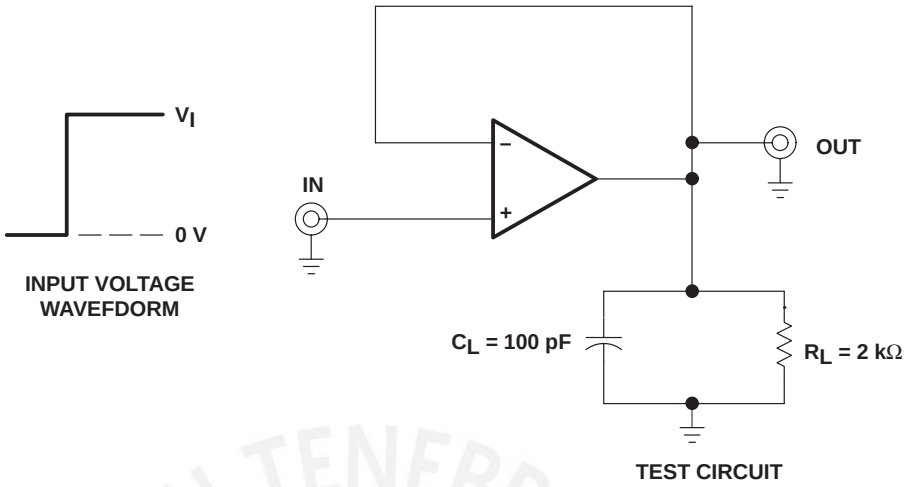


Figure 1. Rise Time, Overshoot, and Slew Rate

APPLICATION INFORMATION

Figure 2 shows a diagram for an input offset voltage null circuit.

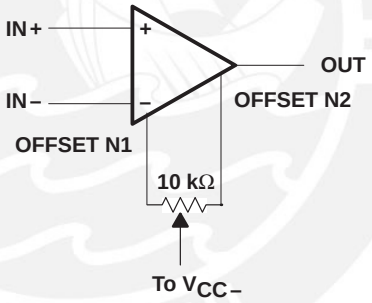


Figure 2. Input Offset Voltage Null Circuit

TYPICAL CHARACTERISTICS†

INPUT OFFSET CURRENT vs FREE-AIR TEMPERATURE

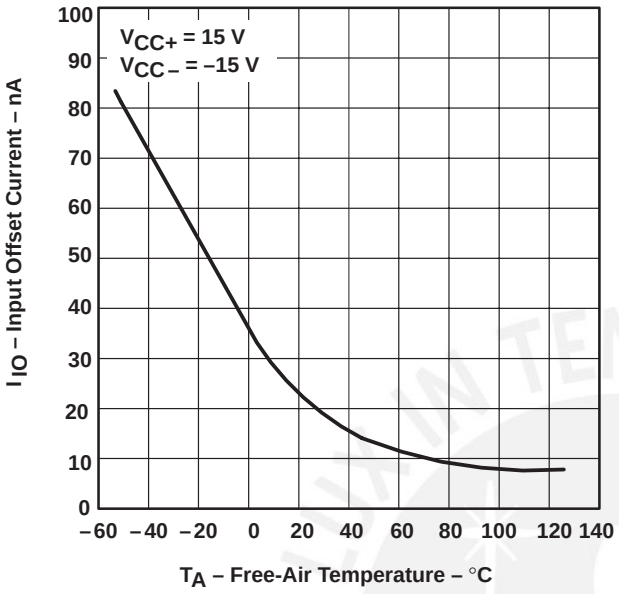


Figure 3

INPUT BIAS CURRENT vs FREE-AIR TEMPERATURE

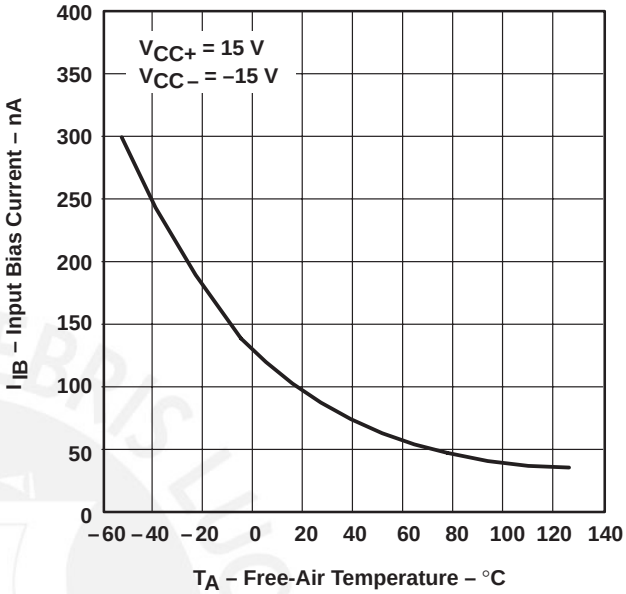


Figure 4

MAXIMUM PEAK OUTPUT VOLTAGE vs LOAD RESISTANCE

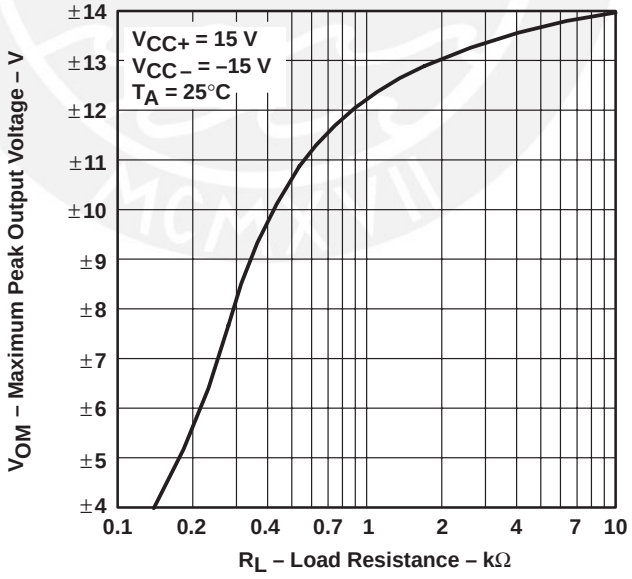


Figure 5

† Data at high and low temperatures are applicable only within the rated operating free-air temperature ranges of the various devices.

TYPICAL CHARACTERISTICS

MAXIMUM PEAK OUTPUT VOLTAGE vs FREQUENCY

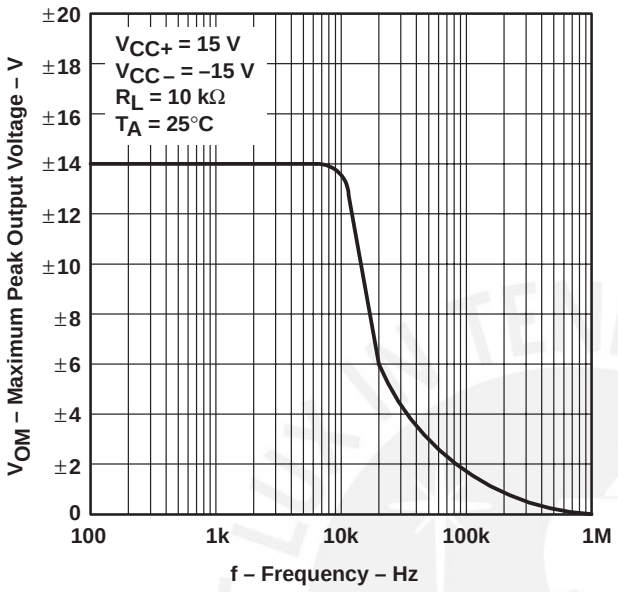


Figure 6

OPEN-LOOP SIGNAL DIFFERENTIAL VOLTAGE AMPLIFICATION vs SUPPLY VOLTAGE

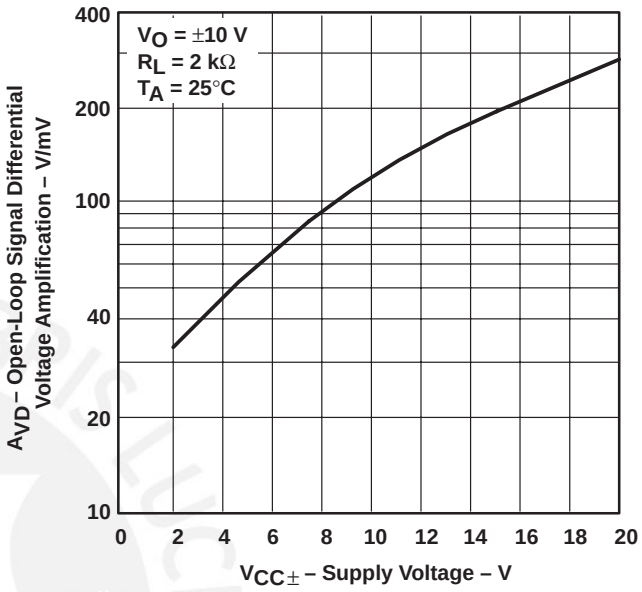
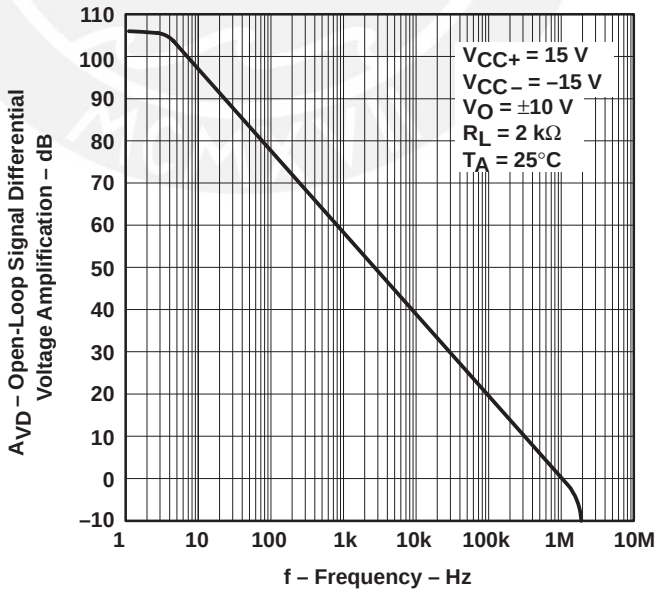


Figure 7

OPEN-LOOP LARGE-SIGNAL DIFFERENTIAL VOLTAGE AMPLIFICATION vs FREQUENCY



TYPICAL CHARACTERISTICS

COMMON-MODE REJECTION RATIO vs FREQUENCY

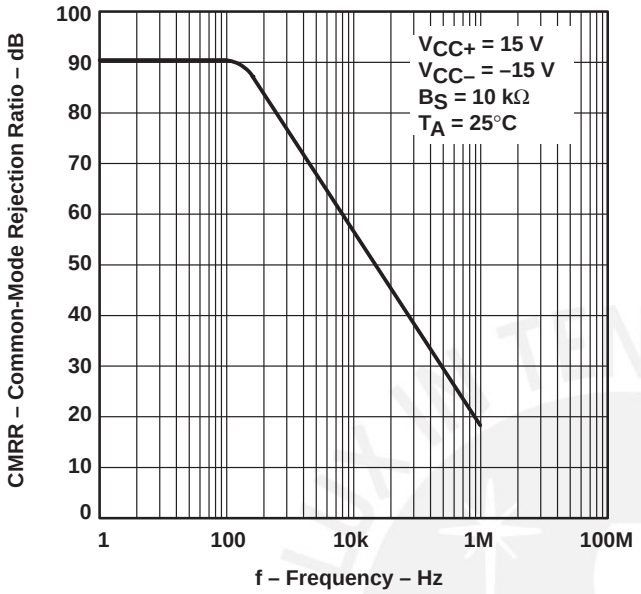


Figure 8

OUTPUT VOLTAGE vs ELAPSED TIME

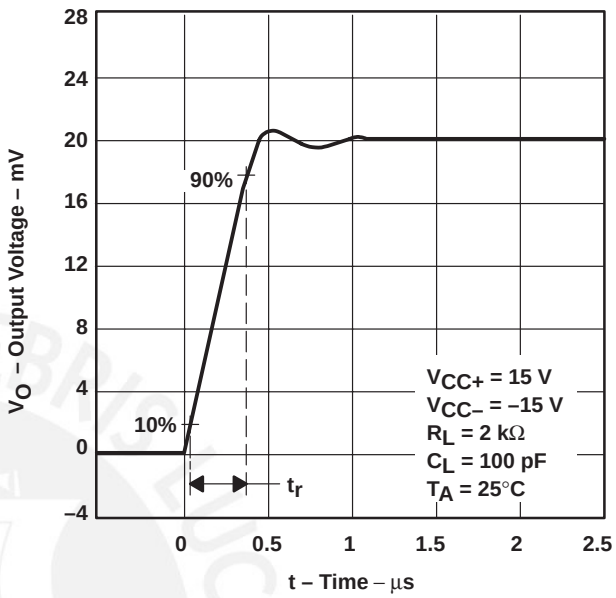


Figure 9

VOLTAGE-FOLLOWER LARGE-SIGNAL PULSE RESPONSE

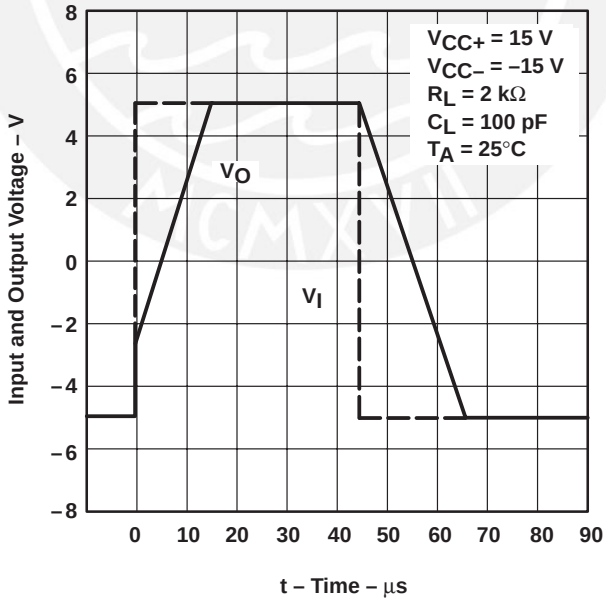


Figure 10

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Detalles de Producto

USB al adaptador RS232 (ZT-RS232B)

USB al adaptador RS232 (ZT-RS232B)

Descripción de Producto
Características:
1. Convertir el puerto estándar del USB a la conformidad de la interfaz en serie 2. de RS-232 DB9 completamente con la especificación 2.0, 1.1 del USB
3. Compatible a los equipos con RS-232 la ayuda estándar de la interfaz en serie 4. sobre tarifa de transferencia de datos 1Mbps
5. Modo automático del apretón de manos de Suport
6. Gerencia para despertar de la ayuda y de la energía alejada
7. Windows 98 /98SE/ME/2000/XP/Vista y 8. máximos X de la ayuda o sobre la comunicación de la velocidad completa del USB 8. y energías bajas accionadas autobús consume

Contenido del paquete:
USB del A. al cable RS232
Cable de extensión del USB del B.
C. CD del conductor de software

Otros artículos opcionales:
USB al cable RS232 (DB9 solamente)
USB al cable RS232 (DB9 y DB25)
Tarjeta de PCMCIA RS232 (DB9)

El Cable Adaptador USB a Puerto Serie RS232 DB9 de 1 pie de Startech.com permite conectar dispositivos seriales RS232 DB9 a su Mac o PC portátil o de escritorio a través de un puerto USB, como si el ordenador tuviera un conector DB9M incorporado.

Una solución eficaz y económica que permite acortar la distancia en términos de compatibilidad entre ordenadores modernos y periféricos preexistentes con conectividad serial.

Viene con 2-años de garantía y soporte técnico gratuito de por vida con el respaldo de StarTech.com

The StarTech.com Advantage

- Instalación y operación simple que brinda conectividad serial a PCs más antiguas.
- Al ser alimentado por el bus USB se evitan las molestias de estar acarreando un adaptador de corriente externo para este puerto serie adicional

- La larga lista de sistemas operativos compatibles hace de este dispositivo un producto perfecto para casi todas las aplicaciones RS232

Especificaciones Técnicas

Garantía	
Warranty	2 Years
Hardware	
Cantidad de Puertos	1
Estilo de Puerto	Cable Adapter
Interfaz	Serial
Tipo de Bus	USB 2.0
Estándares Industriales	USB 1.1/2.0 RS232
ID del Conjunto de Chips	Prolific - PL-2303
Conector(es)	
Tipo(s) de Conector(es)	1 - DB-9 (9 pin; D-Sub) Macho 1 - USB A (4 pin) Macho
Rendimiento	
Tasa Máxima de Baudios	921.6 Kbps
Protocolo Serie	RS-232
FIFO	192 Bytes
Software	
Compatibilidad OS	Windows® 7 (32/64), Vista(32/64), XP(32/64), 2000, ME, 98SE Windows® Server 2008 R2 Mac OS 10.x (Tested up to 10.8)

	Linux
Características Físicas	
Longitud del Cable	304.8 mm [12 in]
Tipo de Gabinete	Plástico
Longitud del Producto	370 mm [14.6 in]
Ancho del Producto	34 mm [1.3 in]
Altura del Producto	18 mm [0.7 in]
Peso del Producto	81.6 g [2.9 oz]
Requisitos Ambientales	
Humedad	HR 0~65%
Temperatura de Almacenamiento	-20°C to 80°C (-4°F to 176°F)
Temperatura Operativa	0°C to 60°C (32°F to 140°F)
Información de la Caja	
Peso (de la Caja) del Envío	0.1 kg [0.2 lb]

7.1.- Comunicación serie

Comentaremos la comunicación SCI (Serial Comunication Interface), poniendo un sencillo ejemplo del funcionamiento y utilidades con el puerto RS-232.

Después haremos hincapié en nuestro dispositivo SCI por USB con el integrado de la empresa FTDI, FT232BM, el cual es un transciver capaz de emular un VCP en nuestro ordenador actuando como hardware el puerto USB.

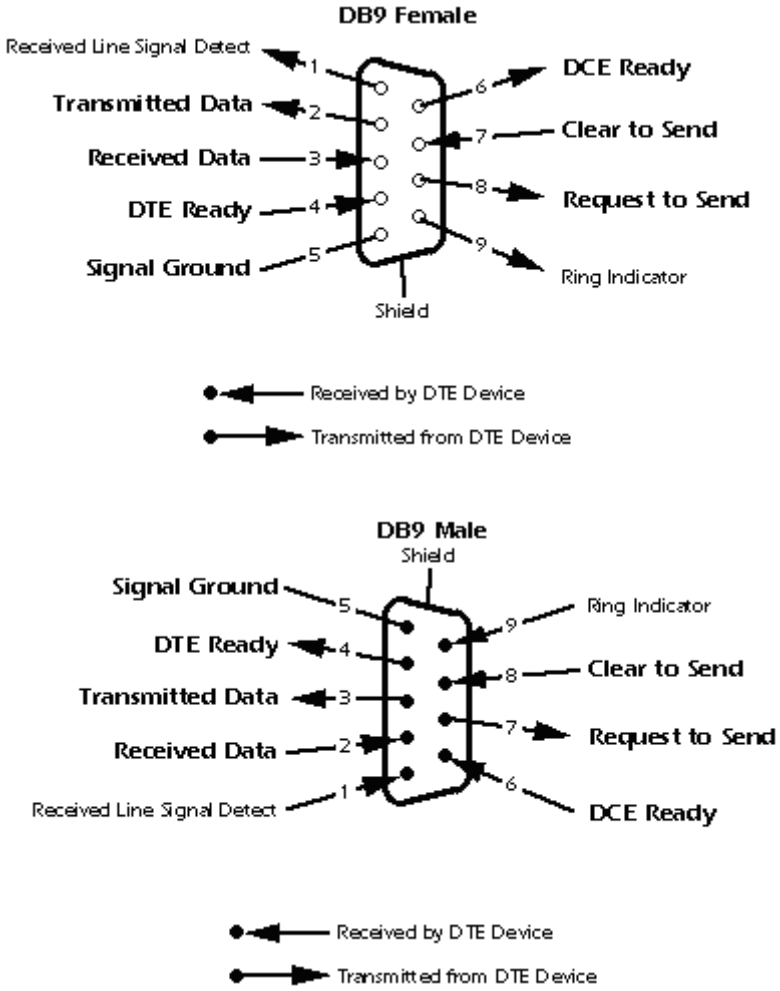
7.1.1 RS-232

SCI es un enlace serie **asíncrono** que data de 1962. También es conocido como **UART** (Universal Asynchronous receiver transmitter). Es **full-duplex** y sólo admite la transmisión-recepción entre dos

elementos (**punto a punto**). La velocidad máxima de transmisión suele ser **256kbps** con cables de**15m** (50 pies). El protocolo es el de la norma **RS-232** pero los niveles eléctricos difieren, por lo que es necesario un integrado adaptador de niveles tipo **MAX232** para poder utilizar el SCI como puerto RS-232. Dicho puerto es también conocido como puerto **COM** o puerto serie.

7.1.1.1 Conectores y cables

El conector más usual es el DB9, mostrado en la siguiente figura.



Conector DB9 y función de cada línea

El conector macho (plug) y el hembra (socket) son algo diferentes. En el primero, la patilla 2 corresponde a la línea de recepción, y la 3 a la línea de transmisión. En el segundo ocurre al revés. De este modo, al interconectar dos dispositivos, si uno tiene conector macho y otro hembra, el cable a utilizar será un cable no cruzado. Si los dos dispositivos tienen el mismo tipo de conector entonces será necesario un cable cruzado, en el que el pin 3 de un extremo está conectado al 2 del otro y viceversa.

El conector RS-232 de nuestro PC es macho. Dado que los cables más comunes y fáciles de encontrar son los macho-hembra, lo más cómodo es que en nuestra tarjeta utilicemos al menos un conector hembra. A la hora de diseñar la placa tendremos que tener presente:

- La patilla 2 y la 3 tienen distintas funciones dependiendo de si el conector es macho o hembra.

- Los footprints de los conectores macho y hembra parecen iguales, pero el orden de sus patillas difiere.

7.1.1.2 El integrado MAX232

El integrado MAX232 está basado en dos **bombas de carga,** una dobladora y otra inversora. Gracias ellas las salidas del integrado pueden tomar valores de +-10V a partir de una tensión de alimentación de 5V. De ahí la necesidad de los condensadores externos, cuyo valor será **1uF** para el MAX232 y**100nF** para el MAX232A y el ST232.

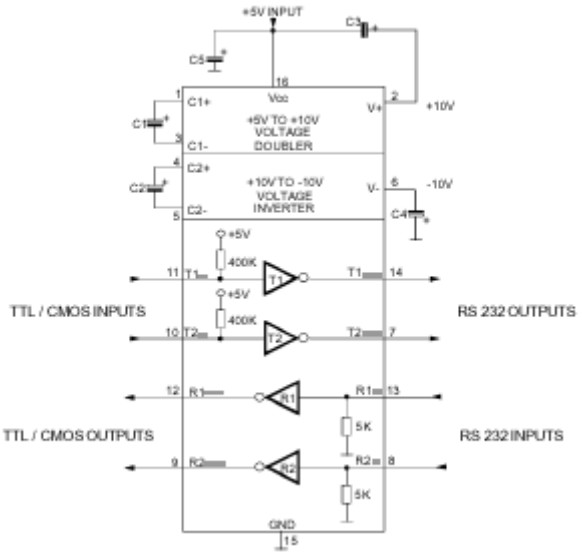
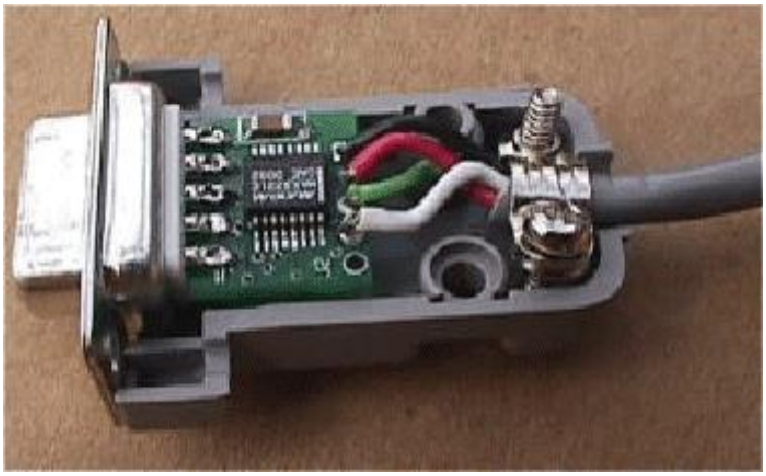


Diagrama de conexiones del ST232/MAX232



Cable adaptador TTL-RS232 basado en MAX232

7.1.1.3 Protocolo

El **estado dominante** de la línea es '1', por lo que se dice que es un protocolo **NRZ** (No Return to Zero). Eléctricamente un '1' se corresponde con un valor entre -3 y -15V, y un '0' con un valor entre +3 y +15V.

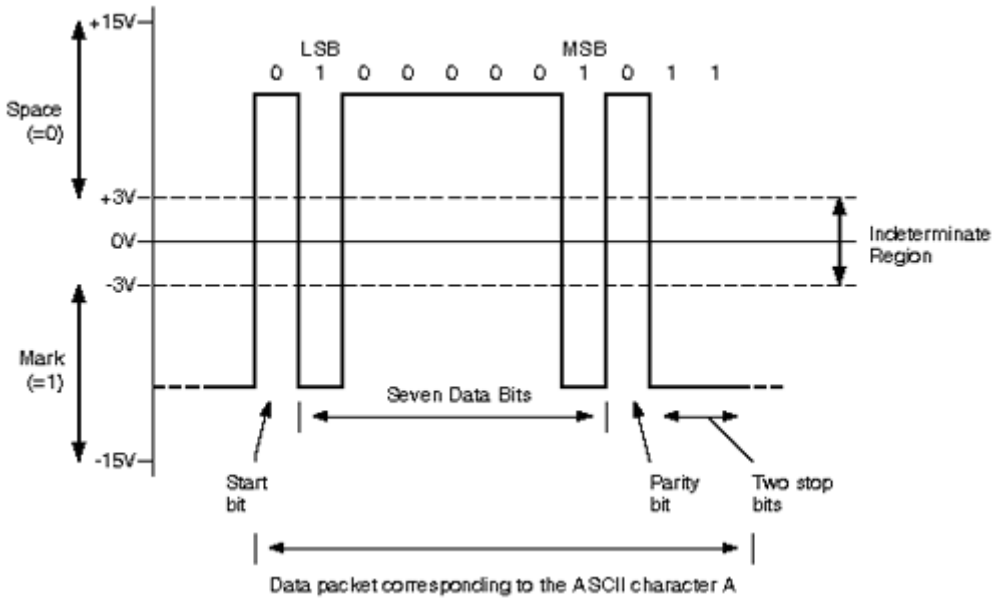
En primer lugar se manda un bit de **start** (0) que sirve al receptor para la **sincronización**. Posteriormente se envía el mensaje (1 byte) y un bit de stop (1). Para enviar el siguiente byte se repetiría el proceso. A este tipo de transmisión se le denomina **8N1**.

A veces se añade un bit de paridad para comprobar si el dato se ha recibido correctamente. Su valor depende del byte enviado y del tipo de paridad.

- Paridad par: Su valor es 0 si el número de ceros enviados es par

- Paridad impar: Su valor es 0 si el número de ceros enviados es impar

Existen otras variaciones en el protocolo tales como la utilización de dos bits de stop, transmisión de datos de siete bits, etc. por lo que tendremos que verificar que las dos unidades utilizan el mismo protocolo. Nosotros usaremos siempre el 8N1, que es con mucho el más utilizado.



Transmisión vía RS232 en modo 7 bits

A la hora de visualizar la transmisión con el osciloscopio hay que tener en cuenta dos cosas:

- Dado que primero se envía el bit de menos peso, el byte debe ser leído de derecha a izquierda.

- Como hemos visto los valores negativos de tensión corresponden a un '1', y los positivos a un '0'. Para no liarnos lo mejor es invertir el canal para ver los unos como tensiones positivas y los ceros como tensiones negativas.

7.1.2 USB

7.1.2.1 Introducción al USB

El **Bus de Serie Universal** (**USB**, de sus siglas en inglés *Universal Serial Bus*) es una interfaz que provee un estándar de bus serie para conectar dispositivos a un ordenador personal (generalmente a un PC). Un sistema USB tiene un diseño asimétrico, que consiste en un solo servidor y múltiples dispositivos conectados en serie para ampliar la gama de conexión, en una estructura de árbol utilizando concentradores especiales.

El estándar incluye la transmisión de energía eléctrica al dispositivo conectado. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios sin necesitar fuentes de alimentación extra. La mayoría de los concentradores incluyen fuentes de alimentación que brindan energía a los dispositivos conectados a ellos, pero algunos dispositivos consumen tanta energía que necesitan su propia fuente de alimentación. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites).

El diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades plug-and-play permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar. Cuando se conecta un nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar.

El USB puede conectar periféricos como ratones, teclados, escáneres, cámaras digitales, impresoras, discos duros, tarjetas de sonido y componentes de red. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha empezado a desplazar a los puertos paralelos porque el USB hace sencillo el poder agregar más de una impresora a un ordenador personal.

En el caso de los discos duros, el USB es poco probable que reemplace completamente a los buses como el ATA (IDE) y el SCSI porque el USB tiene un rendimiento un poco más lento que esos otros estándares. El nuevo estándar Serial ATA permite tasas de transferencia de hasta aproximadamente 150 MB por segundo. Sin embargo, el USB tiene una importante ventaja en su habilidad de poder instalar y desinstalar dispositivos sin tener que abrir el sistema, lo cual es útil para dispositivos de almacenamiento desinstalables. Hoy en día, una gran parte de los fabricantes ofrece dispositivos USB portátiles que ofrecen un rendimiento casi indistinguible en comparación con los ATA (IDE).

El estándar USB 1.1 tenía dos velocidades de transferencia: 1.5 Mbits/s para teclados, ratón, joysticks, etc., y velocidad completa a 12 Mbit/s. La mayor ventaja del estándar USB 2.0 es añadir un modo de alta velocidad de 480 Mbit/s. En su velocidad más alta, el USB compite directamente con FireWare.

Las especificaciones USB 1.0, 1.1 y 2.0 definen dos tipos de conectores para conectar dispositivos al servidor: A y B. Sin embargo, la capa mecánica ha cambiado en algunos conectores. Por ejemplo, el IBM UltraPort es un conector USB privado localizado en la parte superior del LCD de los ordenadores portátiles de IBM. Utiliza un conector mecánico diferente mientras mantiene las señales y protocolos característicos del USB. Otros fabricantes de artículos pequeños han desarrollado también sus medios de conexión pequeños, y una gran variedad de ellos han aparecido. Algunos de baja calidad.

Una extensión del USB llamada "USB-On-The-Go" permite a un puerto actuar como servidor o como dispositivo - esto se determina por qué lado del cable está conectado al aparato. Incluso después de que el cable está conectado y las unidades se están comunicando, las 2 unidades pueden "cambiar de papel" bajo el control de un programa. Esta facilidad está específicamente diseñada para dispositivos como PDA, donde el enlace USB podría conectarse a un PC como un dispositivo, y conectarse como servidor a un teclado o ratón. El "USB-On-The-Go" también ha diseñado 2 conectores pequeños, el mini-A y el mini-B, así que esto debería detener la proliferación de conectores miniaturizados de entrada.

7.1.2.2 Conectores y cables

El USB dispone en su diversidad diferentes tipos de conectores o clavijas dependiendo del uso que le vayamos a dar.

Para ello se dispone en el mercado diferentes tipos de conectores USB, de los cuales sólo indicaremos a continuación aquellos que son los más usados comúnmente.

Conector usb Tipo A

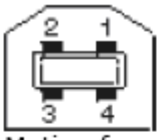
Conector usb Tipo B

Comparativa mini usb macho tipo B y usb macho tipo A

Dependiendo del conector, el número de pines difiere entre ellos y también la disposición de las señales, podemos observarlo en la siguiente tabla:

USB TYPE A and B

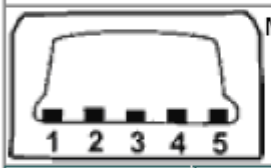
Mating face of USB type A female



Mating face of USB type B female

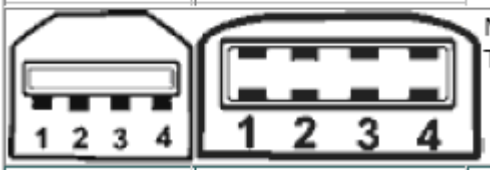
PIN#	SIGNAL	PIN#	SIGNAL
1	+5	3	+Data
2	-Data	4	GND

Mini USB TYPE B



Mating face of 5-pin Mini USB Type B female

PIN#	SIGNAL	PIN#	SIGNAL
1	+5	4	NC
2	-DATA	5	GND
3	+DATA		



Mating face of 4-pin Mini USB Type B female

PIN#	SIGNAL	PIN#	SIGNAL
1	+5	3	+DATA
2	-DATA	4	GND

Disposición de señales de los distintos conectores usb

7.1.2.3 El integrado FT232BM

Dicho integrado realiza las funciones de interface USB <-> Serial mediante hardware, con los mínimos componentes externos.

Las principales características del integrado son:

El integrado se realiza en smd y tiene 32 patillas, es capaz de soportar Full Handshaking. La UART interna del integrado soporta las siguientes características:

- 7 / 8 bits de datos.
- 1 / 2 bits de parada.
- Y diferentes tipos de fin de datos (paridad, espacio…).

La velocidad de la comunicación al ser por hardware es bastante rápida, ésta depende del la señal.

- 3M Baudio (TTL).
- 1M Baudio (RS232).
- 3M Baudios (RS442/RS485).

Tiene un buffer de transmisión de 384 bytes y un buffer de transmisión de 128 bytes.

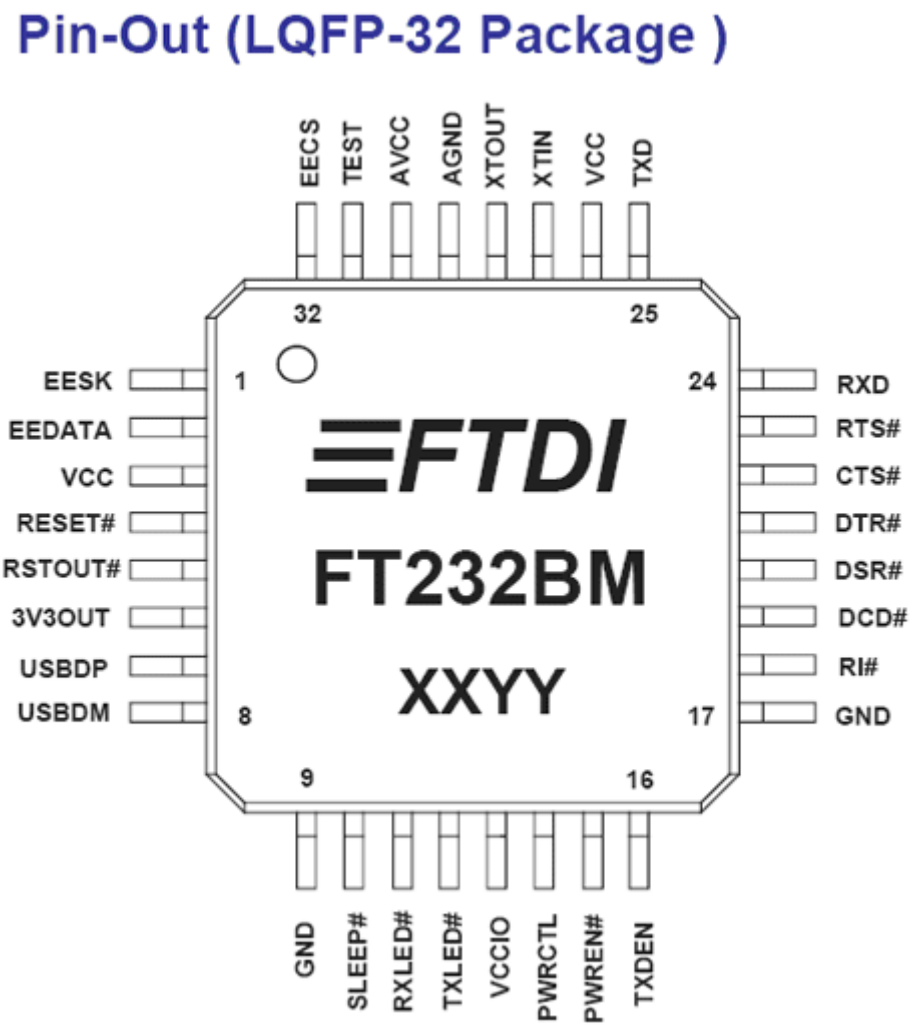
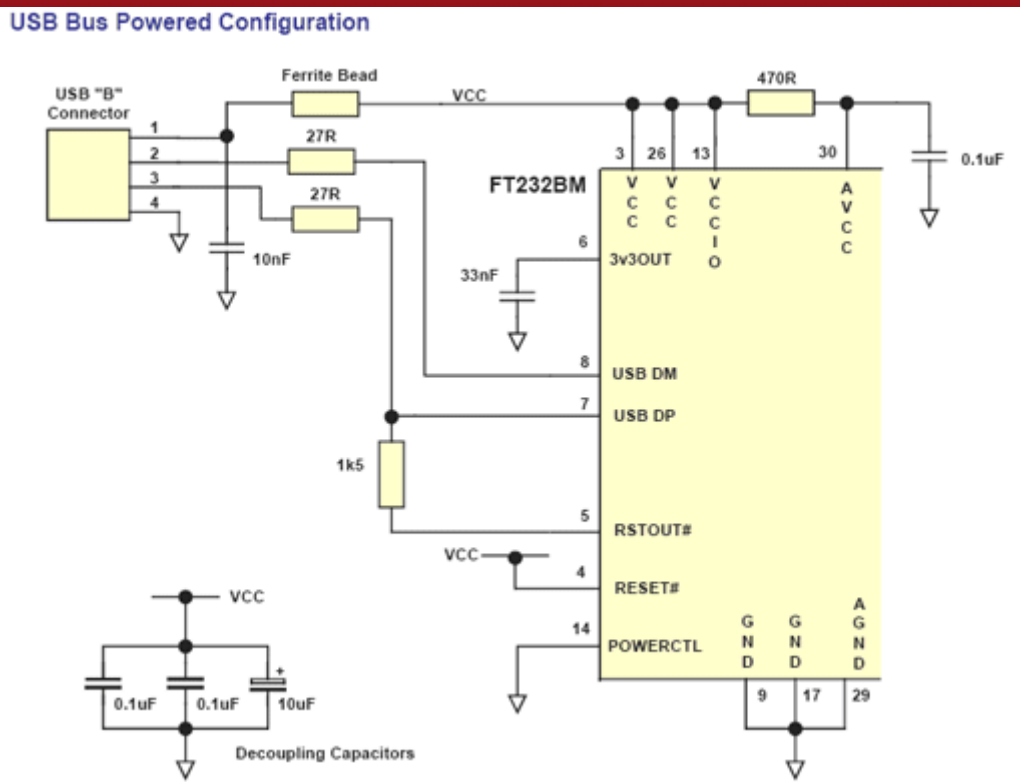


Diagrama de pines

A continuación pondremos el esquemático realizado en nuestra interface para la comunicación entre el PC y nuestro PIC.



Esquemático básico de conexiones usb.

Como podemos observar el número de componentes en mínimo, sólo hace falta 4 resistencias y 5 condensadores, de los cuales 4 de ellos son condensadores de desacoplo, usados para limpiar la señal de alimentación que le llega integrado.

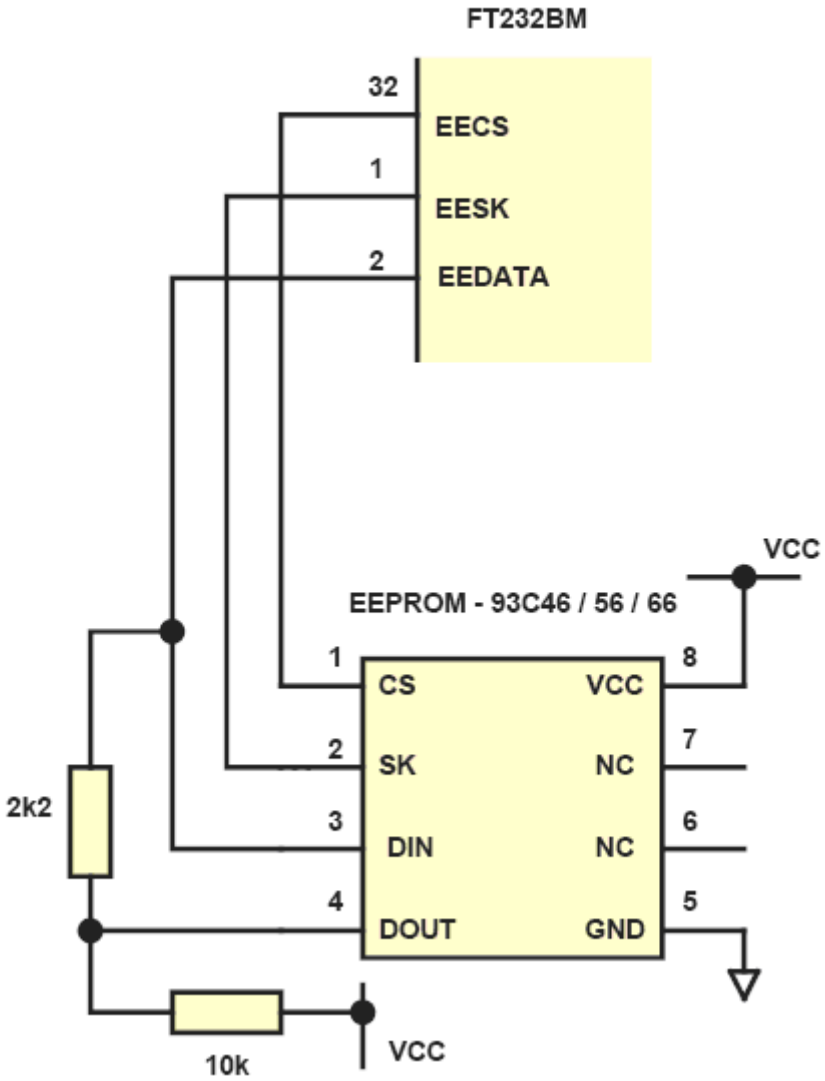
En esta configuración el integrado se alimenta directamente de la alimentación del USB dado por el PC, con lo que no necesita alimentación externa.

También podemos insertar fácilmente una EEPROM externa para poder configurar nuestro dispositivo, de modo que podemos darle un VID y un nombre en concreto a nuestro dispositivo, el cual para nosotros no es muy importante pero si da un gran juego al integrado.

Para ello se optó por una EEPROM de la casa Microchip 93C46, sus principales características son:

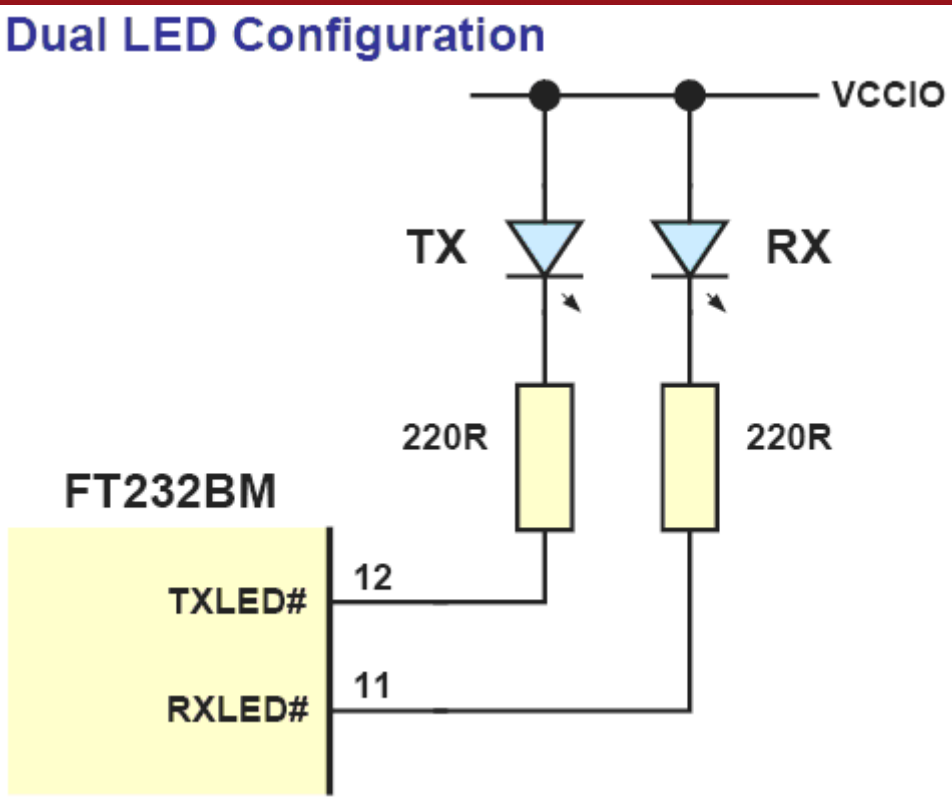
- Memoria EEPROM.
- 1K de capacidad de memoria.
- 128 x 8 bits de tabla de organización.
- Bajo consumo gracias a la tecnología CMOS.

EEProm Configuration



Esquemático conexión FT232BM y EEPROM 93C46

El FT232BM tiene dos salidas para mostrar mediante LEDs si éste está recibiendo o enviando información, lo cual nos es tremendamente útil a la hora de hacer pruebas de comunicación y poder ver lo que ocurre a nuestro integrado.



Conexión de led para la visualización de transmisión de datos

```
;*******************************************************************************
;***** Nombre del programa: MEDIDOR DE VOLUMEN
;***** Autor: Juan Diego Jimenez Carpio
;***** Fecha: junio 2013
;***** Descripción del programa: programa implmentado con algoritmo de multitarea.
;***** controla 4 tareas las cuales tienen las siguiente funciones:
;***** - controlar una pantalla LCD
;***** - escanear un teclado matricial
;***** - realizar caclulos para medir el volumen de cajas
;***** - controlar la transmision de datos por comunicacion USART
;*******************************************************************************

.include "C:\VMLAB\include\m16def.inc"

.equ distancia_fotos=48;46  ;   distancia entre sensores fotoelctricos 46mm

.equ rango_maximo_ancho=250  ;250mm
.equ rangominimo_ancho=40 ;40mm
.equ rango_maximo_alto=250  ;250mm
.equ rangominimo_alto=40 ;40mm

; Define here Reset and interrupt vectors, if any
;
.dseg
.org $240
tecla: .byte 1
exclusive_or: .byte 1
cont_error: .byte 1
```

```
flag_inicio_faja: .byte 1
flag_se_a_calibrado: .byte 1
rr22: .byte 1
rr23: .byte 1
flag_mostrar_datos: .byte 1
envia_dat_serial: .byte 1
cantid_cajas: .byte 1

flag_inicio_sensado: .byte 1
datos_nuevos: .byte 1
nueva_velocida: .byte 1
dimensiones_nuevas: .byte 1

        cuenta_vel: .byte 1
        cuenta_largo: .byte 1
        contador: .byte 1
        cantidad_datos: .byte 1
        flanco_foto1: .byte 1
        flanco_foto2: .byte 1
        contad_restante_timer2_largo: .byte 1
        contad_restante_timer2_vel: .byte 1
        vel_0: .byte 1
        largo: .byte 1
        ancho: .byte 1
        alto: .byte 1

ADCaaL:.byte 1
ADCaaH:.byte 1
```

```
yadress: .byte  2
adc_anch: .byte 2
adc_alt: .byte  2


.org $60
.cseg
reset:
   rjmp start
   reti      ; Addr $01
   rjmp            foto1_flanco_subida_bajada       ; Addr $02
   reti      ; Addr $03
   rjmp  foto2_flanco_subida_bajada       ; Addr $04
   reti      ; Addr $05
   reti      ; Addr $06        Use 'rjmp myVector'
   reti      ; Addr $07        to define a interrupt vector
   rjmp            fin_cuenta_timer2      ;   TIMER2 OVF Timer/Counter2 Overflow      ; Addr
$08
   reti      ; Addr $09
   reti      ; Addr $0A
   reti      ; Addr $0B        This is just an example
   reti      ; Addr $0C        Not all MCUs have the same
   reti      ; Addr $0D        number of interrupt vectors
   reti      ; Addr $0E
   reti      ; Addr $0F
   reti      ; Addr $10
.org $26
   rjmp  Muestreo_Timer_ADC                  ;Timer/Counter0 Compare Match
```

```
.org $30

; Program starts here after Reset
;
start:

        rjmp    IniTareas


;***************************************************************************
;*** DRIVER MULTITAREA
;*** OPTIMIZADA PARA CONTEXTO
;*** Ventaja:          - Guarda todos los registros de trabajo
;***                                  - No requiere cuidado especial al llamar a DoEvents
;***                                  - Se puede llamar a DoEvents en cualquier punto
;***
;*** Desventaja:       - Usa mas recusos (RAM): 33 bytes adicionales por cada tarea
;***                                  - Demora mas tiempo en conmutar tareas:
39+134=173
;***************************************************************************
.equ NumTareas = 4
.equ TamPila =63
.equ TamPilaContexto= 33            ; 33 adicionales para contexto

.cseg
Lista_Tareas:
 .DW Tarea1
 .DW Tarea2
 .DW Tarea3
```

```
 .DW Tarea4


.dseg
Tarea:          .byte 1                         ; Current Task: 0-(NumTareas-1)
Tabla_SP:       .byte 2*NumTareas               ; HI-LO
Buff_Pila:      .byte (TamPila+TamPilaContexto)*NumTareas                ; Todas las Pilas
de las tareas


.cseg
IniTareas:
        cli                                                 ; Deshabilito
interrupciones


        ldi     R16,high(RAMEND)        ; Inicializo Pila
        out     SPH,R16
        ldi     R16,low(RAMEND)
        out     SPL,R16


        rcall   IniPorts                                    ; Puertos a estado incial.
Reposo


        ldi     ZH,high(Lista_Tareas*2)
        ldi     ZL,low(Lista_Tareas*2)
        ldi     XH,high(Tabla_SP)
        ldi     XL,low(Tabla_SP)
        ldi     YH,high(Buff_Pila-1)
        ldi     YL,low(Buff_Pila-1)
        ldi     R20,NumTareas       ; Cantidad de tareas a inicializar
IniTareas_Lazo:
        adiw    YL,TamPila          ; Me voy al final de la Pila
```

```
        adiw    YL,TamPilaContexto

        out     SPH,YH              ; Guardo Tarea y contexto
        out     SPL,YL

        lpm     R16,Z+              ; Leo la direccion de inicio de Tarea
        lpm     R17,Z+
        push    R16                 ; Direccion de Tarea
        push    R17

        clr     R16                 ; Inicialmente contexto en $00
        ldi     R17,33              ; Contexto
IniTareas_Contexto:
        push    R16
        dec     R17
        brne    IniTareas_Contexto

        in          R0,SPL              ; Almaceno en tabla_SP direccion del
SP virtual
        in          R1,SPH
        st          X+,R1
        st          X+,R0

        dec     R20
        brne    IniTareas_Lazo

        ldi     R16,0               ; Comienzo en Tarea0
        sts     Tarea,R16
        lds     R16,Tabla_SP
```

```
out     SPH,R16
lds     R16,Tabla_SP+1
out     SPL,R16
sei

pop     R16                     ; Recupero contexto y Run tarea 0
out     SREG,R16
pop     R31
pop     R30
pop     R29
pop     R28
pop     R27
pop     R26
pop     R25
pop     R24
pop     R23
pop     R22
pop     R21
pop     R20
pop     R19
pop     R18
pop     R17
pop     R16
pop     R15
pop     R14
pop     R13
pop     R12
pop     R11
pop     R10
```

```
        pop     R9
        pop     R8
        pop     R7
        pop     R6
        pop     R5
        pop     R4
        pop     R3
        pop     R2
        pop     R1
        pop     R0
        ret


; En caso de no saber el estado inicial de un pin,
; se recomienda ponerlo en PULL-UP
IniPorts:
        push    R16

;       ldi     R16,$ff
;       out     PORTB,R16
        ldi     R16,$00
        out     DDRB,R16

;       ldi     R16,$ff
;       out     PORTC,R16
        ldi     R16,$00
        out     DDRC,R16

;       ldi     R16,$ff
```

```
;       out     PORTD,R16
        ldi     R16,$00
        out     DDRD,R16

        pop     R16
        ret



.macro DoEvents
        call    DoEventos
.endm

DoEventos:
        push    R0                              ; Guarda Contexto
        push    R1
        push    R2
        push    R3
        push    R4
        push    R5
        push    R6
        push    R7
        push    R8
        push    R9
        push    R10
        push    R11
        push    R12
        push    R13
        push    R14
        push    R15
```

```
push    R16
push    R17
push    R18
push    R19
push    R20
push    R21
push    R22
push    R23
push    R24
push    R25
push    R26
push    R27
push    R28
push    R29
push    R30
push    R31
in              R16,SREG                        ; Banderas
push    R16

ldi     XH,high(Tabla_SP)   ; Apunto en la tabla_sp al SP actual
ldi     XL,low(Tabla_SP)
lds     R16,Tarea
lsl     R16
add     XL,R16
clr     R16
adc     XH,R16
in              R16,SPL                                 ; Leo el SP actual
in              R17,SPH
st              X+,R17                                  ; Almaceno el SP actual
```

```
        st      X+,R16

        lds     R16,Tarea               ; Paso a la siguiente Tarea
        inc     R16
        cpi     R16,NumTareas
        brne    DoEventos_Cont
        clr     R16                     ; Regreso a Tarea0
        ldi     XH,high(Tabla_SP)
        ldi     XL,low(Tabla_SP)
DoEventos_Cont:
        sts     Tarea,R16

        ld      R17,X+                  ; Leo SP de la siguiente
tarea
        ld      R16,X+
        cli
        out     SPH,R17                 ; Modifico el SP
        out     SPL,R16
        sei

        nop
        nop
        nop
        nop
        nop

        pop     R16                     ; Recupero contexto
        out     SREG,R16
        pop     R31
```

```
pop     R30
pop     R29
pop     R28
pop     R27
pop     R26
pop     R25
pop     R24
pop     R23
pop     R22
pop     R21
pop     R20
pop     R19
pop     R18
pop     R17
pop     R16
pop     R15
pop     R14
pop     R13
pop     R12
pop     R11
pop     R10
pop     R9
pop     R8
pop     R7
pop     R6
pop     R5
pop     R4
pop     R3
pop     R2
```

```
        pop     R1
        pop     R0
        ret


;***************************************************************************
;
;***************************************************************************
;
;***************************************************************************
;
;***************************************************************************
;
;***************************************************************************
;
;***************************************************************************
;
;***************************************************************************
;
;*** TAREA 1 - MENU
;***************************************************************************
;
.macro printf
        ldi             ZH,high(@1*2)
        ldi             ZL,low(@1*2)
        ldi     r16,@0          ; ubicacion del dato en la pantalla
        rcall   lcd_mensaje
.endm


Tarea1:
        ldi     R16,0b00111111          ;Puerto B: B0-B3  Nibble de r17s del LCD
(lineas de control del LCD):   PB5=E        PB4=R/W  pd7=RS
        out     DDRB,R16
        ldi             R16,0b10010000              ; PD7 y pd4: TEACH
        out             PORTD,R16
        ldi     R16,0b11110010          ;Puerto
```

```
        out     DDRD,R16                ;PD5:Motor PD2-PD3:Interrupciones externas
PD0:RX      PD1:TX


        rcall inicia_variables
        ;rcall retardo 1 ms
        rcall Configura_LCD



        printf $c4,Mensaje_l1
        printf $9a,Mensaje_l2


Tarea1_Menu:
        DoEvents
        lds r16,tecla
        cpi r16,$1f             ; se presiono menu?
        brne Tarea1_Menu
        ldi r16,$0
        sts flag_mostrar_datos,r16
Muestra_Menu:
        rcall Limpiar
        printf $88,Mensaje_menu
        printf $c0,Mensaje_menu1
        printf $94,Mensaje_menu2
        printf $d4,Mensaje_menu3


Escanea_teclado:
        DoEvents
        lds r16,tecla
        cpi r16,$11             ; se presiono 1?
```

```
brne sensa_2
        lds r16,flag_se_a_calibrado
        cpi r16,10
        brsh sise_calibro
        rcall limpiar
        printf $c0,debe_calibrar
        printf $94,debe_calibrar2
        rjmp Tarea1_Menu
        sise_calibro:
        lds r18,flag_inicio_faja
        cpi r18,10
        brsh faja_prendida
            ldi     zh,high(tabla*2) ;inicia faja a minima velocidad
            ldi     zl,low(tabla*2)
            lpm     r18,z+
            lpm     r19,z+
            out     OCR1AH,r19
            out     OCR1AL,r18
            ldi r16,$ff                         ; la faja a iniciado
            sts flag_inicio_faja,r16
            sts flag_inicio_sensado,r16
        faja_prendida:
        ldi r16,0b11000000              ; int1 int0
        out GIFR,r16                    ; pone a cero los falgs de
interrupciones
        out GICR,r16                    ; abilita interrupciones
        sei                             ;abilita interrupciones globales
        rcall limpiar
```

```
        printf $c0,proceso_iniciado
        rjmp Tarea1_Menu
sensa_2:
lds r16,tecla
cpi r16,$12             ; se presiono 2?
brne sensa_3
        rcall limpiar
        ldi r16,100
        sts flag_inicio_faja,r16
        printf $c0,Mensaje_variar
        printf $94,Mensaje_sale
        lazo_velocida:
                DoEvents
                lds r16,tecla
                cpi r16,$1A             ; se presiono up?
                brne sensa_down
                ldi r16,0
                sts tecla,r16
                rcall sube_velocidad
                rjmp lazo_velocida
                sensa_down:
                lds r16,tecla
                cpi r16,$1B             ; se presiono down?
                brne sensa_menu
                ldi r16,0
                sts tecla,r16
                rcall baja_velocidad
                rjmp lazo_velocida
                sensa_menu:
```

```
                lds r16,tecla
                cpi r16,$1F             ; se presiono menu?
                brne lazo_velocida
                ldi r16,0
                sts tecla,r16
                rjmp Muestra_Menu
sensa_3:
lds r16,tecla
cpi r16,$13             ; se presiono 3?
brne sensa_4
        lds r16,flag_inicio_faja
        cpi r16,10
        brsh mensaje_sedebe_detener_faja
        rcall limpiar
        printf $80,menu_calibrado
        printf $c0,calibrado_nuevo
        printf $94,calibrado_prev
        printf $d4,calibrado_prev2
        lazo_calibrado:
                DoEvents
                lds r16,tecla
                cpi r16,$11             ;se presiono 1?
                brne sensa_old
                rcall nueva_calibracion
                rjmp lazo_calibrado
                sensa_old:
                lds r16,tecla
                cpi r16,$12             ;se presiono 2?
                brne sensa_meun
```

```
                ldi r16,0
                sts tecla,r16
                rcall vieja_calibracion
                rjmp lazo_calibrado
                sensa_meun:
                lds r16,tecla
                cpi r16,$1f             ;se preciono menu?
                brne lazo_calibrado
                rjmp Muestra_Menu
        mensaje_sedebe_detener_faja:
        rcall limpiar
        printf $c0,debe_detener_faja
        printf $94,debe_detener_faja2
        rjmp Tarea1_Menu
sensa_4:
lds r16,tecla
cpi r16,$14             ; se presiono 4?
brne sensa_5
        lds r16,flag_inicio_faja
        cpi r16,10
        brlo muestra_mensaje_debeinicarfaja
                ldi r16,$FF
                sts flag_mostrar_datos,r16
                rcall muestra_etiquetas_dimensiones
                rjmp Tarea1_Menu
        muestra_mensaje_debeinicarfaja:
        rcall limpiar
        printf $c0,debe_iniciar_faja
        printf $94,debe_iniciar_faja2
```

```
        rjmp Tarea1_Menu
sensa_5:
lds r16,tecla
cpi r16,$15             ; se presiono 5?
brne sensa_6
        rcall limpiar
        printf $80,msj_serial0
        printf $c0,msj_serial1
        printf $94,Msj_serial2
        lazo_envio_serial:
                DoEvents
                lds r16,tecla
                cpi r16,$11             ; se presiono 1?
                brne sensa_det_envio
                ldi r16,$ff
                sts envia_dat_serial,r16
                sts tecla,r16
                rcall limpiar
                printf $c0,envio_data_activo
                rjmp lazo_envio_serial
                sensa_det_envio:
                lds r16,tecla
                cpi r16,$12             ; se presiono 2?
                brne sensa_menus
                ldi r16,0
                sts envia_dat_serial,r16
                sts tecla,r16
                rcall limpiar
                printf $c0,envio_data_desactivo
```

```
                rjmp lazo_envio_serial
                sensa_menus:
                lds r16,tecla
                cpi r16,$1F             ; se presiono menu?
                brne lazo_envio_serial
                rjmp Muestra_Menu
sensa_6:
lds r16,tecla
cpi r16,$16             ; se presiono 6?
brne sensa_dnuevo
        ldi r16,0
        sts tecla,r16
        sts flag_inicio_faja,r16
        sts flag_mostrar_datos,r16
        sts envia_dat_serial,r16
        cli
        ldi     r19,0
        out OCR1AH,r19
        out OCR1AL,r19
        ldi r16,0b11000000      ; int1 int0
        out GIFR,r16                    ; pone a cero los falgs de interrupciones
        ldi r16,0b00000000      ;
        out GICR,r16              ; desabilita interrupciones

        rcall limpiar
        printf $c0,stop_proces
        rjmp Tarea1_Menu
sensa_dnuevo:
rjmp Escanea_teclado
```

```
Mensaje_l1:                .db "MEDIDOR   DE",0,0
Mensaje_l2:                .db "VOLUMEN",0

Mensaje_menu:              .db "MENU",0,0
Mensaje_menu1:             .db "1.Iniciar  2.Vel",0,0
Mensaje_menu2:             .db "3.Calibrar 4.Datos",0,0
Mensaje_menu3:             .db "5.Envia PC 6.Detener",0,0

Mensaje_variar:            .db "Utilice las flechas.",0,0
Mensaje_sale:              .db "Menu:Salir",0,0
Mensaje_limite_max: .db "Maximo limite",0
Mensaje_limite_min:  .db "Minimo limite",0
espacio:                   .db "           ",0

debe_calibrar:         .db "Primero debe",0,0
debe_calibrar2:            .db "calibrar sensores",0

proceso_iniciado:      .db "Proceso iniciado :)",0

debe_detener_faja:     .db "Primero debe parar",0,0
debe_detener_faja2:  .db "el proceso",0,0

menu_calibrado:            .db "Calibrado",0
calibrado_nuevo:       .db "1.Calibrado nuevo",0
calibrado_prev:            .db "2.Usar calibrado",0,0
calibrado_prev2:       .db "  anterior",0,0

colok_caja:                .db "Coloque una caja",0,0
```

```
colok_caja2:            .db "de 21x21x21cm",0

retire_objetos_delafaja:    .db "Retire los objetos",0,0
retire_objetos_delafaja2:   .db "sobre la faja",0
retire_objetos_delafaja3:   .db "luego presione enter",0,0

exito:                      .db "Sensores calibrados",0
para_salir:                 .db "Salir: Menu",0

debe_iniciar_faja:      .db "Primero debe iniciar",0,0
debe_iniciar_faja2:     .db "el proceso",0,0

cajas:                      .db "Caja",0,0
mensaje_volumen:        .db "Volumen:",0,0
cm3:                        .db "cm",3,0
coma0:                          .db "0,",0,0
ms:                             .db "m/s",0
altoo:                      .db "Alto:",0
cm:                             .db "cm",0,0
anchoo:                         .db "Ancho:",0,0
largoo:                     .db "Largo:",0,0

envio_data_activo:      .db "Envio activo",0,0
envio_data_desactivo:               .db "Envio desactivado",0

msj_serial0:            .db "Envio a PC",0,0
msj_serial1:            .db "1.Activar tiemporeal",0,0
msj_serial2:            .db "2.Desactivar envio",0,0
```

```
stop_proces:            .db "Proceso detenido",0,0


;TABLA VELOCIDADES
tabla:
                ;.dw 4999;11999        19999
                .dw 5999;12999
                .dw 6999;13999
                .dw 7999;14999
                .dw 8999;15999
                .dw 9998;16999
                ;.dw 17999
        ;       .dw 18999



;*****************************************************************************************************
*****
muestra_etiquetas_dimensiones:
push r16
rcall limpiar
printf $90,cajas
printf $80,mensaje_volumen
printf $8c,cm3
printf $e1,coma0
printf $e5,ms
printf $94,altoo
printf $9d,cm
printf $d4,anchoo
printf $de,cm
printf $c0,largoo
```

```
printf $ca,cm
ldi r16,$D0
rcall WriteIR
ldi r16,'#'
rcall WriteDR
rcall imprime_cajita
pop r16
ret
;*****************************************************************************************************
*****

;*************************************************************************
imprime_cajita:
push r16
push r18

ldi r16,$8f
rcall WriteIR
ldi r16,4
rcall WriteDR
ldi r16,$cf
rcall WriteIR
ldi r16,4
rcall WriteDR

ldi r16,$a3
rcall WriteIR
ldi r16,5
rcall WriteDR
```

```
ldi r16,$a4
rcall WriteIR
ldi r16,6
rcall WriteDR
ldi r16,$a5
rcall WriteIR
ldi r16,6
rcall WriteDR
ldi r16,$a6
rcall WriteIR
ldi r16,6
rcall WriteDR
ldi r16,$a7
rcall WriteIR
ldi r16,6
rcall WriteDR

ldi r16,$a1
rcall WriteIR
ldi r16,6
rcall WriteDR
ldi r16,$a2
rcall WriteIR
ldi r16,6
rcall WriteDR

ldi r16,$a0
rcall WriteIR
ldi r16,7
```

```
rcall WriteDR


ldi r16,$e0
rcall WriteIR
ldi r16,4
rcall WriteDR


pop r18
pop r16
ret
;***************************************************************************
;*************************************************************************************************
*****
nueva_calibracion:
push    r16
rcall    limpiar
printf $80,colok_caja
printf $c0,colok_caja2
printf $94,retire_objetos_delafaja3
lazo_nueva_calibracion:
        DoEvents
        lds     r16,tecla
        cpi     r16,$1d                                 ;se presiono enter?
        breq    continua_calibrado
rjmp    lazo_nueva_calibracion
continua_calibrado:
ldi     r16,0
sts     tecla,r16
rcall   teach_baja_sensor_Pa6_Pa7       ; puerto pa6 y pa7  pasa de alta a baja por
0.05seg
```

```
rcall     limpiar
printf $80,retire_objetos_delafaja
printf $c0,retire_objetos_delafaja2
printf $94,retire_objetos_delafaja3
lazo_nueva_calibracio:
          DoEvents
          lds       r16,tecla
          cpi       r16,$1d                                   ; se presiono enter?
          breq      continua_calibrad
rjmp      lazo_nueva_calibracio
continua_calibrad:
rcall     teach_baja_sensor_Pa6_Pa7          ; puerto pa6 y pa7 pasa de alta a baja por
0.05seg
ldi       r16,$ff                                   ; flag : ya se calibro sensores
sts       flag_se_a_calibrado,r16
rcall     limpiar
printf $c0,exito
printf $94,para_salir
pop       r16
ret
;*************************************************************************************************************
*****
teach_baja_sensor_Pa6_Pa7:
push r16
cbi porta,6
cbi porta,7
ldi r16,50
rcall RetardoXms
sbi porta,6
sbi porta,7
```

```
pop r16
ret
;*****************************************************************************************************
*****
vieja_calibracion:
push    r16
ldi     r16,$ff         ; flag : ya se calibro sensores
sts     flag_se_a_calibrado,r16
rcall   limpiar
printf $c0,exito
printf $94,para_salir
pop     r16
ret
;*****************************************************************************************************
*****
;*****************************************************************************************************
*****
sube_velocidad:
push r21
push r22
push r23
        in              r21,OCR1AL
        in              r21,OCR1AH
        cpi     r21,$27
        brsh    mostrar_mensaje_limite_max

        clc
        ldi             zh,high(tabla*2)
        ldi             zl,low(tabla*2)
        lds     r22,rr22
```

```
        add     zl,r22
        lds     r23,rr23
        add     zh,r23

        lpm     r18,z+
        lpm     r19,z+

        out     OCR1AH,r19
        out     OCR1AL,r18
        inc     r22
        inc     r22
        sts     rr22,r22

        printf $80,espacio
        rjmp fin_sube_velocidad

        mostrar_mensaje_limite_max:
        printf $80,Mensaje_limite_max

fin_sube_velocidad:
pop r23
pop r22
pop r21
ret
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****

baja_velocidad:
push r21
```

```
push r22
push r23
        in      r21,OCR1AL
        in      r21,OCR1AH
        cpi     r21,$17
        breq    mostrar_mensaje_limite_min

        clc
        ldi     zh,high(tabla*2)
        ldi     zl,low(tabla*2)
        lds     r22,rr22
        add     zl,r22
        lds     r23,rr23
        add     zh,r23

        ldi     r18,2
        ldi     r19,0
        sub     zl,r18
        sbc     zh,r19
        lpm     r18,z+
        lpm     r19,z+

        out     OCR1AH,r19
        out     OCR1AL,r18
        ldi     r18,2
        ldi     r19,0
        sub     zl,r18
        sbc     zh,r19
        dec     r22
```

```
        dec             r22
        sts     rr22,r22
        printf $80,espacio
        rjmp    fin_baja_velocidad
        mostrar_mensaje_limite_min:
        printf $80,Mensaje_limite_min

fin_baja_velocidad:
pop r23
pop r22
pop r21
ret
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****


inicia_variables:
        push    R16

        ldi r16,0
        sts flag_se_a_calibrado,r16
        sts flag_inicio_faja,r16
        sts flag_mostrar_datos,r16
        sts envia_dat_serial,r16

        sts cantid_cajas,r16

        pop R16
        ret
```

```
;*************************************************************************************************
*****

;*************************************************************************
lcd_mensaje:
        push    r16
        push    zh
        push    zl

        rcall   WriteIR
        lazo_muestra:
        lpm     r16,Z+                  ;Lee y muestra caracter en LCD
        cpi     r16,0
        breq    fin_envia_lcd
        rcall   WriteDR
        rjmp    lazo_muestra
        fin_envia_lcd:
        pop     zl
        pop     zh
        pop     r16
ret
;*************************************************************************************************
******************

;*************************************************************************
;**** Subrutina: Envia una instruccion al LCD
;**** Entrada:  r16
WriteIR:

        push    R16
        push    r31
```

```
cbi         portb,4     ;RW=0
cbi         portb,5     ;E=0
cbi         portd,7     ;RS=0

swap   r16      ;XX54XXXX
mov    r31,r16
andi   r31,$0f
out    PORTB,r31
cbi    portb,4      ;RW=0
sbi         portb,5     ;E=1
cbi         portd,7     ;RS=0
rcall  Retardo50us
cbi    portb,4      ;RW=0
cbi         portb,5     ;E=0
cbi         portd,7     ;RS=0
rcall  Retardo50us

swap   r16
andi   r16,$0F
out    PORTB,r16
cbi    portb,4      ;RW=0
sbi         portb,5     ;E=1
cbi         portd,7     ;RS=0
rcall  Retardo50us
cbi    portb,4      ;RW=0
cbi         portb,5     ;E=0
cbi         portd,7     ;RS=0

rcall  Retardo50us
```

```
        rcall   Retardo50us
        rcall   Retardo50us
        rcall   Retardo50us
        rcall   Retardo50us
        rcall   Retardo50us
        rcall   Retardo50us
        rcall   Retardo50us
        rcall   Retardo50us
        rcall   Retardo50us
;-DoEvents

        sbi     portb,4         ;RW=1
        cbi             portb,5     ;E=0
        cbi     portd,7     ;RS=0

        pop     r31
        pop     R16
        ret
;*********************************************************************************
;**** Subrutina: Envia un dato al LCD
;**** Entrada:  r16
WriteDR:
        push    R16
        push    R17
        push    r31

        mov     r17,r16
        cbi     portb,4                 ;RW=0
        cbi             portb,5         ;E=0
```

```
sbi         portd,7     ;RS=1

swap  r17
mov   r31,r17
andi  r31,$0F
out   PORTB,r31           ;se envia el nibble MSB
cbi   portb,4             ;RW=0
sbi         portb,5     ;E=1
sbi         portd,7     ;RS=1
rcall Retardo50us
cbi   portb,4             ;RW=0
cbi         portb,5     ;E=0
sbi         portd,7     ;RS=1
rcall Retardo50us

swap  r17
andi  r17,$0F
out   PORTB,r17           ;se envia el nibble LSB
cbi   portb,4             ;RW=0
sbi         portb,5     ;E=1
sbi         portd,7     ;RS=1
rcall Retardo50us
cbi   portb,4             ;RW=0
cbi         portb,5     ;E=0
sbi         portd,7     ;RS=1

sbi   portb,4             ;RW=1
cbi         portb,5     ;E=0
cbi         portd,7     ;RS=0
```

```
        pop     r31
        pop     R17
        pop     R16

        ret
;***************************************************************************
;
;*************************************************************************************************************
;
******************

;*** Subrutina: Configura LCD  (En sus hojas técnicas se muestra
; la secuencia que se debe seguir para inicializar la pantalla LCD)
; Dar un tiempo para encender la pantalla LCD
; Configurar (Funcion Set) la pantalla para manejar r17s de 8 bits
; Configurar el control ON/OFF (desactivar el display, el cursor y el parpadeo)
; Limpiar la pantalla
; Configurar en modo Set para actualizar el contador de direcciones
; Mover el cursor a su posición inicial
Configura_LCD:

        push    R16
        push    R17
        ldi     r16,1
        rcall   RetardoXms      ;espera 1 ms

        cbi             portb,4                 ;RW=0    ;XX54XXXX
        cbi             portb,5         ;E=0
        cbi             portd,7         ;RS=0

        ldi     r16,$20                         ;Función Set: Configuracion a 4 bits :   DB7
DB6 DB5 DB4 0010xxxx
```

```
        swap    r16
        out     PORTB,r16
        cbi             portb,4                 ;RW=0
        sbi             portb,5         ;E=1
        cbi             portd,7         ;RS=0
        rcall   Retardo50us
        cbi             portb,4                 ;RW=0
        cbi             portb,5         ;E=0
        cbi             portd,7         ;RS=0
        rcall   Retardo50us
        rcall   Retardo50us

        ldi     r16,$28                 ;Funcion Set
        rcall   WriteIR
        ldi     r16,$08                 ;desactivar la pantalla, el cursor y el parpadeo
        rcall   WriteIR
        ldi     r16,$01                         ;limpiar la pantalla (funcion clear)
        rcall   WriteIR
        ldi     r16,$06                 ; El cursor avanza a la derecha cada vez que se
imprime
        rcall   WriteIR         ; un caracter

        ldi     r16,0b00001100          ;Enciende pantalla y desactivar cursor
        rcall   WriteIR

        rcall char_flechas

        pop     R17
        pop     R16
```

```
        ret
;*************************************************************************
;*********************************************************************
Retardo50us:
        push    R16
        ldi     r16,16
        lazo_retardo50us:
        dec     r16
        brne    lazo_retardo50us
        pop             R16
        ret
;***********************************************
;*********************************************************************
;*** Espera milisegundos. Se asume fosc=1MHz
;*** Entrada: R16 = Numero de milisegundos
;*********************************************************************
RetardoXms:
        push    R16
        push    XL
        push    XH

Delayms_Lazo1:
        ldi             XH,high(250)
        ldi             XL,low(250)
Delayms_Lazo2:
        sbiw    XL,1
        brne    Delayms_Lazo2
        dec             R16
```

```
        brne    Delayms_Lazo1
        pop             XH
        pop             XL
        pop             R16
        ret
;***********************************************
;*******************************************************************************
;**** Limpia la pantalla LCD Y regresa el cursor
Limpiar:
        push    R16
        ldi     r16,$01
        rcall   WriteIR
        ldi r16,1
        rcall RetardoXms
        pop     R16
        ret
;*******************************************************************************
;*******************************************************************************

char_flechas:
ldi r16,0b01001000
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01001001
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR
```

```
ldi r16,0b01001010
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01001011
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01001100
rcall WriteIR
ldi r16,0b01011111
rcall WriteDR

ldi r16,0b01001101
rcall WriteIR
ldi r16,0b00001110
rcall WriteDR

ldi r16,0b01001110
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR
```

```
;----------------------------

ldi r16,0b01010110
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR


ldi r16,0b01010101
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR


ldi r16,0b01010100
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01010011
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR


ldi r16,0b01010010
rcall WriteIR
ldi r16,0b01011111
rcall WriteDR
```

```
ldi r16,0b01010001
rcall WriteIR
ldi r16,0b00001110
rcall WriteDR

ldi r16,0b01010000
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR
;---------------

ldi r16,0b01011110
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01011101
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01011100
rcall WriteIR
ldi r16,0b00011100
```

```
rcall WriteDR

ldi r16,0b01011011
rcall WriteIR
ldi r16,0b00000010
rcall WriteDR


ldi r16,0b01011010
rcall WriteIR
ldi r16,0b01000100
rcall WriteDR


ldi r16,0b01011001
rcall WriteIR
ldi r16,0b00000010
rcall WriteDR

ldi r16,0b01011000
rcall WriteIR
ldi r16,0b00011100
rcall WriteDR


;_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:

ldi r16,0b01100000
rcall WriteIR
```

```
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01100001
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01100010
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01100011
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01100100
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01100101
rcall WriteIR
```

```
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01100110
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01100111
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

;_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:

ldi r16,0b01101000
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR


ldi r16,0b01101001
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR


ldi r16,0b01101010
rcall WriteIR
```

```
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01101011
rcall WriteIR
ldi r16,0b00011111
rcall WriteDR


ldi r16,0b01101100
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR


ldi r16,0b01101101
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01101110
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01101111
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR
```

```
;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;_;

ldi r16,0b01110000
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR


ldi r16,0b01110001
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR


ldi r16,0b01110010
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01110011
rcall WriteIR
ldi r16,0b00011111
rcall WriteDR


ldi r16,0b01110100
rcall WriteIR
ldi r16,0b00000000
```

```
rcall WriteDR

ldi r16,0b01110101
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01110110
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01110111
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

;_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:_:

ldi r16,0b01111000
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01111001
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR
```

```
ldi r16,0b01111010
rcall WriteIR
ldi r16,0b00000000
rcall WriteDR

ldi r16,0b01111011
rcall WriteIR
ldi r16,0b00000111
rcall WriteDR

ldi r16,0b01111100
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01111101
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01111110
rcall WriteIR
ldi r16,0b00000100
rcall WriteDR

ldi r16,0b01111111
rcall WriteIR
ldi r16,0b00000100
```

```
rcall WriteDR

ret

;***************************************************************************
;
*************************************************************************

;***************************************************************************
;

;***************************************************************************
;

;***************************************************************************
;

;***************************************************************************
;

;***************************************************************************
;

;***************************************************************************
;

;*** TAREA 2 - Teclado

;***************************************************************************
;

Tarea2:


        ldi     R16,0b00001111      ;Puerto C: C0-C3      Filas del teclado matricial:
        PC0=F1      PC1=F2      PC2=F3      PC3=F4

        out     DDRC,R16            ; C4-C7         Colum. del teclado matricial: PC4=C1
        PC5=C2      PC6=C3   PC7=C4


        ldi r16,0

        sts tecla,r16


teclado_lazo:

DoEvents

sbi portc,0             ;F1

cbi portc,1             ;F2

cbi portc,2              ;F3

cbi portc,3             ;F4

in r16,pinc
```

```
andi r16,0b11110000

cpi r16,0b00010000          ;C1
breq tecla_1
rjmp sigue_2
tecla_1:
rcall tecla_1_suelta
rjmp teclado_lazo

sigue_2:
cpi r16,0b00100000          ;C2
breq tecla_2
rjmp sigue_3
tecla_2:
rcall tecla_2_suelta
rjmp teclado_lazo

sigue_3:
cpi r16,0b01000000          ;C3
breq tecla_3
rjmp sigue_up
tecla_3:
rcall tecla_3_suelta
rjmp teclado_lazo

sigue_up:
cpi r16,0b10000000          ;C4
breq tecla_up
rjmp sigue_4
```

```
tecla_up:
rcall tecla_up_suelta
rjmp teclado_lazo

sigue_4:
cbi portc,0                    ;F1
sbi portc,1          ;F2
cbi portc,2          ;F3
cbi portc,3          ;F4
in r16,pinc
andi r16,0b11110000

cpi r16,0b00010000          ;C1
breq tecla_4
rjmp sigue_5
tecla_4:
rcall tecla_4_suelta
rjmp teclado_lazo

sigue_5:
cpi r16,0b00100000          ;C2
breq tecla_5
rjmp sigue_6
tecla_5:
rcall tecla_5_suelta
rjmp teclado_lazo

sigue_6:
cpi r16,0b01000000          ;C3
```

```
breq tecla_6
rjmp sigue_down
tecla_6:
rcall tecla_6_suelta
rjmp teclado_lazo

sigue_down:
cpi r16,0b10000000          ;C4
breq tecla_down
rjmp sigue_7
tecla_down:
rcall tecla_down_suelta
rjmp teclado_lazo

sigue_7:
cbi portc,0                 ;F1
cbi portc,1         ;F2
sbi portc,2         ;F3
cbi portc,3         ;F4
in r16,pinc
andi r16,0b11110000

cpi r16,0b00010000          ;C1
breq tecla_7
rjmp sigue_8
tecla_7:
rcall tecla_7_suelta
rjmp teclado_lazo
```

```
sigue_8:
cpi r16,0b00100000          ;C2
breq tecla_8
rjmp sigue_9
tecla_8:
rcall tecla_8_suelta
rjmp teclado_lazo

sigue_9:
cpi r16,0b01000000          ;C3
breq tecla_9
rjmp sigue_2nd
tecla_9:
rcall tecla_9_suelta
rjmp teclado_lazo

sigue_2nd:
cpi r16,0b10000000          ;C4
breq tecla_2nd
rjmp sigue_clear
tecla_2nd:
rcall tecla_2nd_suelta
rjmp teclado_lazo

sigue_clear:
cbi portc,0                         ;F1
cbi portc,1         ;F2
cbi portc,2         ;F3
sbi portc,3         ;F4
```

```
in r16,pinc
andi r16,0b11110000

cpi r16,0b00010000          ;C1
breq tecla_clear
rjmp sigue_0
tecla_clear:
rcall tecla_clear_suelta
rjmp teclado_lazo

sigue_0:
cpi r16,0b00100000          ;C2
breq tecla_0
rjmp sigue_help
tecla_0:
rcall tecla_0_suelta
rjmp teclado_lazo

sigue_help:
cpi r16,0b01000000          ;C3
breq tecla_help
rjmp sigue_enter
tecla_help:
rcall tecla_help_suelta
rjmp teclado_lazo

sigue_enter:
cpi r16,0b10000000          ;C4
breq tecla_enter
```

```
rjmp teclado_lazo
tecla_enter:
rcall tecla_enter_suelta
rjmp teclado_lazo

tecla_1_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b00010000  ;C1
breq    tecla_1_suelta

ldi     r16,$11
sts     tecla,r16
ret

tecla_2_suelta:
DoEvents
in              r16,pinc
andi    r16,0b11110000
cpi     r16,0b00100000  ;C2
breq    tecla_2_suelta

ldi     r16,$12
sts     tecla,r16
ret

tecla_3_suelta:
DoEvents
```

```
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b01000000  ;C3
breq    tecla_3_suelta

ldi     r16,$13
sts     tecla,r16
ret

tecla_up_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b10000000  ;C4
breq    tecla_up_suelta
ldi     r16,$1A
sts     tecla,r16
ret

tecla_4_suelta:
DoEvents
in              r16,pinc
andi    r16,0b11110000
cpi     r16,0b00010000  ;C1
breq    tecla_4_suelta

ldi     r16,$14
sts     tecla,r16
ret
```

```
tecla_5_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b00100000  ;C2
breq    tecla_5_suelta

ldi     r16,$15
sts     tecla,r16
ret

tecla_6_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b01000000  ;C3
breq    tecla_6_suelta

ldi     r16,$16
sts     tecla,r16
ret

tecla_down_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b10000000  ;C4
breq    tecla_down_suelta
```

```
ldi     r16,$1B
sts     tecla,r16
ret

tecla_7_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b00010000  ;C1
breq    tecla_7_suelta

ldi     r16,$17
sts     tecla,r16
ret

tecla_8_suelta:
DoEvents
in              r16,pinc
andi    r16,0b11110000
cpi     r16,0b00100000  ;C2
breq    tecla_8_suelta

ldi     r16,$18
sts     tecla,r16
ret

tecla_9_suelta:
DoEvents
```

```
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b01000000  ;C3
breq    tecla_9_suelta

ldi             r16,$1f
sts     tecla,r16
ret

tecla_2nd_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b10000000  ;C4
breq    tecla_2nd_suelta

ldi     r16,$1d
sts     tecla,r16
ret

tecla_clear_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b00010000  ;C1
breq    tecla_clear_suelta

ldi     r16,$1E
sts     tecla,r16
```

```
ret

tecla_0_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b00100000  ;C2
breq    tecla_0_suelta

ldi     r16,$10
sts     tecla,r16
ret

tecla_help_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b01000000  ;C3
breq    tecla_help_suelta

ldi     r16,$1F
sts     tecla,r16
ret

tecla_enter_suelta:
DoEvents
in      r16,pinc
andi    r16,0b11110000
cpi     r16,0b10000000  ;C4
```

```
breq    tecla_enter_suelta

ldi     r16,$1D
sts     tecla,r16
ret



;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*************************************************************************
;*** TAREA 3 - Medicion
;*************************************************************************
Tarea3:

        rcall Configura_TMR1
        rcall Configura_puertos
        rcall Configura_ADC
        rcall config_timer_ADC      ; frecuencia de meustreo del ADC
        rcall config_timer_2          ;  timer para calular el largo ela caja
        rcall config_interr_exter   ;config inter externas en flacno de subida y bajada
respectibamente

        ldi r16,0
        sts datos_nuevos,r16
```

```
        sts nueva_velocida,r16
        sts dimensiones_nuevas,r16

Lazo_tarea3:
DoEvents
lds r16,flag_inicio_sensado
cpi r16,10
brsh calcula_nueva_velocida
rjmp Lazo_tarea3

calcula_nueva_velocida:
lds r16,nueva_velocida
cpi r16,10
brsh halla_veloci

lds r16,datos_nuevos
cpi r16,1
brsh realiza_calculos
rjmp Lazo_tarea3


halla_veloci:
rcall calcula_vel_faja_eprom
clr r16
sts nueva_velocida,r16
rjmp Lazo_tarea3

realiza_calculos:
rcall calcula_largo_guard_eprom
```

```
rcall calcula_ancho_guard_eprom


rcall calcula_alto_guard_eprom
clr r16
sts datos_nuevos,r16
ldi r16,$ff
sts dimensiones_nuevas,r16

lds r16,flag_mostrar_datos
cpi r16,10
brsh muestro_datos_lcd
rjmp Lazo_tarea3
muestro_datos_lcd:
lds r16,vel_0
rcall convertir_a_ascii
        ldi r16,$e3
        call WriteIR
        mov     r16,R18
        call WriteDR
        mov r16,R17
        call WriteDR
lds r16,largo
rcall convertir_a_ascii
        ldi r16,$c6
        call WriteIR
        mov r16,R19
        call WriteDR
        mov     r16,R18
```

```
        call WriteDR
        ldi     r16,','
        call WriteDR
        mov r16,R17
        call WriteDR
lds r16,alto
rcall convertir_a_ascii
        ldi r16,$99
        call WriteIR
        mov r16,R19
        call WriteDR
        mov     r16,R18
        call WriteDR
        ldi     r16,','
        call WriteDR
        mov r16,R17
        call WriteDR
lds r16,ancho
rcall convertir_a_ascii
        ldi r16,$da
        call WriteIR
        mov r16,R19
        call WriteDR
        mov     r16,R18
        call WriteDR
        ldi     r16,','
        call WriteDR
        mov r16,R17
        call WriteDR
```

```
        ldi r16,$88
        call WriteIR
lds r16,alto
lds r17,ancho
lds r18,largo
rcall mul_1byt_1byt_1byt
rcall bin24_ascii
        mov r16,R22
        call WriteDR
        mov     r16,R21
        call WriteDR
        mov     r16,R20
        call WriteDR
        mov r16,R19
        call WriteDR


ldi r16,$D1
rcall WriteIR
lds r16,cantid_cajas
inc r16
sts cantid_cajas,r16
rcall convertir_a_ascii
mov r16,r19
rcall WriteDR
mov r16,r18
rcall WriteDR
mov r16,r17
```

```
rcall WriteDR

nop
nop
nop
nop

rjmp Lazo_tarea3

;*************************************************************************************************
*****
foto1_flanco_subida_bajada:
push r16
        lds     r16,flanco_foto1  ; 1= subida     0 = bajada
        cpi     r16,1
        brlo    flanco_bajada_1

        ldi     r16,0
        sts     contador,r16
        out     TCNT2,r16                       ; contador de timer2 =0

        in      r16,TIMSK
        ori     r16,0b01000000      ;  abilita inter por overflow timer 2
        out     TIMSK,r16           ; aca inicia la cuenta del timer2 para calcular el largo

        in      r16,TIFR                        ; limpia banderas de interrupcion por overflow2
        ori     r16,0b11000000
        out     TIFR,r16
```

```
        ldi     r16,0b00001110                  ; ....1110  : INT1 falco subida foto 2
INT0 falnco bajada    foto1

        out     MCUCR,r16


        ldi     r16,0                   ; configura flanco de bajada

        sts     flanco_foto1,r16


        ldi     R16,0b11010011      ;1:adc abilitado 1:inicio manual 0:AutoTriggering OFF
10:limpia bandera y desactiva interrupcion de fin de conversion 011:pre=8 frec conv=125K

        out     ADCSRA, R16


        rjmp    fin_flanco_subida_1

flanco_bajada_1:

        lds     r16,contador

        sts     cuenta_largo,r16


        in      r16,TCNT2                   ;leez contador restante de tiemr2

        sts     contad_restante_timer2_largo,r16


        ldi     r16,0

        out     TCNT2,r16           ;contadore timer2=0


        in      r16,TIMSK

        andi    r16,0b10111111

        out     TIMSK,r16           ;abilita inter por overflow timer 2


        in      r16,MCUCR

        ori     r16,$03     ; ......11  :      INT0 falnco subida    foto1

        out     MCUCR,r16
```

```
        ldi     r16,1
        sts     flanco_foto1,r16
        sts     datos_nuevos,r16
fin_flanco_subida_1:
pop r16
reti
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****

foto2_flanco_subida_bajada:     ; en flanco de subida saca velocidad, en flanco de bajada
opera dimensiones y volummen
push r16

        lds     r16,flanco_foto2        ; 1= subida     0 = bajada
        cpi     r16,1
        brlo    flanco_bajada_2

        lds     r16,contador
        sts     cuenta_vel,r16

        in      r16,MCUCR
        andi    r16,0b11110011
        ori     r16,0b00001000                  ; ....1011  : INT1 falco bajada foto 2
        out     MCUCR,r16

        in      r16,TCNT2                       ; lee contador restante de tiemr2
        sts     contad_restante_timer2_vel,r16

        in      r16,TIMSK
```

```
        ori     r16,0b00000010          ;, 01....10 desab comparematch timer2 y abilita
overflow timer2..... abilita int comparacion timer0 y desabilita int overflow timer0
        out     TIMSK,r16               ;abilita ADC

        ldi     yh,high(adc_anch)       ;ubicaion en ram para almacenar datos
        sts     yadress,yh
        ldi             yl,low(adc_anch)        ;ubicaion en ram para almacenar datos
        sts yadress+1,yl

        ldi     r16,$ff
        sts     nueva_velocida,r16
        ldi     r16,0
        sts     flanco_foto2,r16

        rjmp    fin_flanco_subida_2
flanco_bajada_2:
        ldi     r16,1
        sts     flanco_foto2,r16

        in      r16,MCUCR
        ori     r16,$c0      ; ....11..  :      INT1 falnco subida    foto2
        out     MCUCR,r16

        ldi     R16,0b01010011    ;0:adc desabilitado 1:inicio manual 0:AutoTriggering
OFF 10:limpia bandera y desactiva interrupcion de fin de conversion 011:pre=8 frec
conv=125K
        out     ADCSRA, R16

ldi r16,0
sts cuenta_vel,r16
```

```
sts cuenta_largo,r16
sts contador,r16
sts cantidad_datos,r16
sts contad_restante_timer2_largo,r16
sts contad_restante_timer2_vel,r16


fin_flanco_subida_2:
pop r16
reti
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****
fin_cuenta_timer2:
push r16
        lds r16,contador
        inc r16
        sts contador,r16    ; reinicio de cuenta timer2 automatico TCNT2=0
pop r16
reti
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****
Muestreo_Timer_ADC:      ; lee el adc de canal 0 y canal 1 y los guarda en RAM
push r16
push xh
push xl            ; mejorar puntero Y q se almacene n RAM sin push pop!!!!!!
        ldi      R16,0b01000000      ;01:avcc=5v 1:ajusizq 00000:ADC0 para el canal 0
ANCHO
        out      ADMUX,R16
```

```
        rcall   leeADC              ; tiempo para que la conversion ese lista en ADCH y
ADCL
        ldi     xh,0        ;ubicaion en ram de ADCL
        ldi     xl,$24      ;ubicaion en ram de ADCL
        ld      r16,x+      ; lee y depsues pasa a ADCH
        lds yh,yadress
        lds yl,yadress+1
        st      y+,r16      ;ppuntero(Y): direccion donde kiero q se guarde la
conversion
        ld      r16,x
        st      y+,r16

        ldi     R16,0b01000001      ;01:avcc=5v 1:ajusizq 00001:ADC1 para el canal 1
ALTO
        out     ADMUX,R16
        rcall   leeADC
        ldi     xh,0        ;ubicaion en ram de ADCL
        ldi     xl,$24      ;ubicaion en ram de ADCL
        ld      r16,x+
        st      y+,r16
        ld      r16,x
        st      y+,r16
        sts yadress,yh
        sts yadress+1,yl

        lds     r16,cantidad_datos
        cpi     r16,10
        brlo no_apaga_timer
                in      r16,TIMSK
```

```
                andi    r16,0b11111101          ;, 01....10 desab comparematch timer2
y abilita overflow timer2..... abilita int comparacion timer0 y desabilita int overflow timer0
                out     TIMSK,r16
                ldi     r16,0
                sts     cantidad_datos,r16
                rjmp fin_adc                    ; para q no incremente      cantidad_datos
        no_apaga_timer:
        inc r16
        sts cantidad_datos,r16
        fin_adc:                                ;cantidad_datos no incrementado
pop xl
pop xh
pop r16
reti
;***************************************************************************************************
*****

;***************************************************************************************************
*****
leeADC:
push r16
        in      r16,adcsra      ; ad star conversion =1
        ori     r16,$40
        out     adcsra,r16
        leeADClazo:
        in      r16,adcsra
        sbrc    r16,6;adsc
        rjmp    leeADClazo
pop r16
ret
```

```
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****

calcula_vel_faja_eprom:

push r16

push r17

push r18

push r19

push r20

push r21


lds r18,cuenta_vel

clc

rol r18

clr r16

clr r19

adc r19,r16

lsl r19

clc

rol r18

adc r19,r16

lsl r19

clc

rol r18

adc r19,r16

clr r17          ;multiplica cuenta_velx2048    ;rpt1 en r19:r18:r17


lds r20,contad_restante_timer2_vel

clc
```

```
rol r20
clr r16
clr r21
adc r21,r16
lsl r21
clc
rol r20
adc r21,r16
lsl r21
clc
rol r20
adc r21,r16
clr r22          ;multiplica contad_restante_timer2_velx8    ;rpt1 en r22:r21:r20


;sumo los tiempos
add r20,r17
adc r21,r18
adc r22,r19;     tiempo total en r22:r21:20   = denominador


ldi r18,byte3(distancia_fotos*100000)
ldi r17,high(distancia_fotos*100000)
ldi r16,low(distancia_fotos*100000)  ;numerador


rcall div_3byt_3byt ; divide 3 bytes ente 3 bytes entrada  r18:r17:r16  /  r22:r21:r20 salida :
r18:r17:r16


sts vel_0,r16


pop r21
```

```
pop r20
pop r19
pop r18
pop r17
pop r16
ret
;*************************************************************************************
;*************************************************************************************
*****
calcula_largo_guard_eprom:
push r16
lds r18,cuenta_largo
clc
rol r18
clr r16
clr r19
adc r19,r16
lsl r19
clc
rol r18
adc r19,r16
lsl r19
clc
rol r18
adc r19,r16
clr r17          ;multiplica cuenta_largox2048    ;rpt1 en r19:r18:r17

lds r20,contad_restante_timer2_largo
clc
```

```
rol r20
clr r16
clr r21
adc r21,r16
lsl r21
clc
rol r20
adc r21,r16
lsl r21
clc
rol r20
adc r21,r16
clr r22             ;multiplica contad_restante_timer2_largox8    ;rpt1 en r22:r21:r20

;sumo los tiempos
add r20,r17
adc r21,r18
adc r22,r19;      tiempo total en r22:r21:20   de largo

lds r16,vel_0

rcall mul_3byt_1byt     ; multiplia 3 bytes por 1 byte   entrada r19:r18:r17 x r16     salida:
r20:r19:r18:r17
mov r16,r17
mov r17,r18
mov r18,r19
mov r19,r20

ldi r20,low(100000)
```

```
ldi r21,byte2(100000)
ldi r22,byte3(100000)

rcall div_4byt_3byt
sts largo,r16

pop r16
ret
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****
calcula_ancho_guard_eprom:
push r16
push r17
push r18
push r19
push r20
push r21
push r23

lds r17,adc_anch
lds r16,adc_anch+1
lds r19,adc_anch+4
lds r18,adc_anch+5
add r17,r19
adc r16,r18
lds r19,adc_anch+8
lds r18,adc_anch+9
add r17,r19
```

```
adc r16,r18
lds r19,adc_anch+12
lds r18,adc_anch+13
add r17,r19
adc r16,r18
lds r19,adc_anch+16
lds r18,adc_anch+17
add r17,r19
adc r16,r18
lds r19,adc_anch+20
lds r18,adc_anch+21
add r17,r19
adc r16,r18
lds r19,adc_anch+24
lds r18,adc_anch+25
add r17,r19
adc r16,r18
lds r19,adc_anch+28
lds r18,adc_anch+29
add r17,r19
adc r16,r18
clc
ror r16
ror r17
clc
ror r16
ror r17
clc
ror r16
```

```
ror r17

;entrada r18:r17 x r16
mov r18,r16
ldi r16,80
rcall mul_2byt_1byt ; rpta r19:r18:r17

ldi r16,(56) ; 56 se btiene de la formula en el documento
sub r18,r16

ldi r17,(218-8)

sub r17,r18

sts ancho,r17

pop r23
pop r21
pop r20
pop r19
pop r18
pop r17
pop r16
ret
.*****
;
*****

.*****
;
*****
calcula_alto_guard_eprom:
push r19
```

```
push r20
push r21
push r22
push r23
push r24
push r25
push r26
push r27
push r28
push r29


lds r17,adc_alt
lds r16,adc_alt+1
lds r19,adc_alt+4
lds r18,adc_alt+5
add r17,r19
adc r16,r18
lds r19,adc_alt+8
lds r18,adc_alt+9
add r17,r19
adc r16,r18
lds r19,adc_alt+12
lds r18,adc_alt+13
add r17,r19
adc r16,r18
lds r19,adc_alt+16
lds r18,adc_alt+17
add r17,r19
```

```
adc r16,r18
lds r19,adc_alt+20
lds r18,adc_alt+21
add r17,r19
adc r16,r18
lds r19,adc_alt+24
lds r18,adc_alt+25
add r17,r19
adc r16,r18
lds r19,adc_alt+28
lds r18,adc_alt+29
add r17,r19
adc r16,r18
clc
ror r16
ror r17
clc
ror r16
ror r17
clc
ror r16
ror r17

;entrada r18:r17 x r16
mov r18,r16
ldi r16,77
rcall mul_2byt_1byt ; rpta r19:r18:r17

ldi r16,(54) ; 54 se btiene de la formula en el documento
```

```
sub r18,r16

ldi r17,(210)

sub r17,r18

sts alto,r17

pop r29
pop r28
pop r27
pop r26
pop r25
pop r24
pop r23
pop r22
pop r21
pop r20
pop r19
ret
.*****************************************************************************************************
;
*****

.*************************************************************************
;
;Entradas: R16
;Salidas: r19,r18,r17
convertir_a_ascii:

        push    R16
```

```
        ldi     R19,0
        ldi     R18,0
centena:
        cpi     R16,100
        brlo    decena
        subi    R16,100
        inc     R19
        rjmp    centena

decena:
        cpi     R16,10
        brlo    unidad
        subi    R16,10
        inc     R18
        rjmp    decena

unidad:
        subi    R16,-($30)
        mov             R17,R16
        subi    R18,-($30)
        subi    R19,-($30)

        pop     R16
        ret
;***************************************************************************
;*************************************************************************************************
*****

mul_3byt_1byt:  ; multiplia 3 bytes por 1 byte   entrada r19:r18:r17 x r16     salida:
r20:r19:r18:r17

push r16
```

```
mul r16,r17
mov r17,r0
mov r20,r1
mul r16,r18
mov r18,r0
add r18,r20
mov r20,r1
mul r16,r19
mov r19,r0
add r19,r20
mov r20,r1
ldi r16,0
adc r20,r16

pop r16
ret
;*****************************************************************************************************
*****

;*****************************************************************************************************
*****

mul_1byt_1byt_1byt:     ; nmultiplica 1 byte x 1 byte x 1byte  entrada r16,r17,r18 salida
r18:r17:r16
mul r17,r18
mov r18,r1
mov r17,r0
rcall mul_2byt_1byt ; rpta r19:r18:r17
mov r16,r17
mov r17,r18
mov r18,r19
```

```
ret

;*****************************************************************************************************************
*****

;*****************************************************************************************************************
*****

mul_2byt_1byt: ; multiplia 2 bytes por 1 byte   entrada r18:r17 x r16     salida: r19:r18:r17

push r16


mul r16,r17

mov r17,r0

mov r19,r1

mul r16,r18

mov r18,r0

add r18,r19

mov r19,r1

ldi r16,0

adc r19,r16


pop r16

ret

;*****************************************************************************************************************
*****

;*****************************************************************************************************************
*****

div_4byt_3byt: ; divide 4 bytes entre 3 bytes entrada  r19:r18:r17:r16  /  r22:r21:r20 salida :
r18:r17:r16

push r19

push r20

push r21

push r22

push r23
```

```
push r24
push r25
push r26
push r27
push r28

clr r28

mov r24,r20
mov r25,r21
mov r26,r22
mov r27,r23

div1:
cp r23,r19
brsh div2
add r20,r24
adc r21,r25
adc r22,r26
adc r23,r27
inc r28
rjmp div1

div2:
cp r22,r18
brsh div3
add r20,r24
adc r21,r25
adc r22,r26
```

```
adc r23,r27
inc r28
rjmp div2

div3:
mov r16,r28
inc r16

pop r28
pop r27
pop r26
pop r25
pop r24
pop r23
pop r22
pop r21
pop r20
pop r19
ret
;*************************************************************************************************
*****

;*************************************************************************************
div_3byt_3byt:
push r19
push r20
push r21
push r22
push r24
push r25
```

```
push r26
push r27
push r28
push r29

ldi r16,0
ldi r19,1
ldi r27,0
ldi r28,0
ldi r29,0


mov r24,r20
mov r25,r21
mov r26,r22
compara:
cp r18,r22
brlo resultado

add r20,r24
adc r21,r25
adc r22,r26

add r27,r19
adc r28,r16
adc r29,r16

rjmp compara
resultado:
```

```
;la respuesta esta en r29:r28:r27

mov r16,r27
mov r17,r28
mov r18,r29

pop r29
pop r28
pop r27
pop r26
pop r25
pop r24
pop r22
pop r21
pop r20
pop r19
ret
;*************************************************************************************************
;
;*************************************************************************************************
*****
Configura_TMR1:
        push    r18
        push    r22
        push    r23

        ldi     r18,(1<<COM1A1|0<<COM1A0|1<<WGM11|0<<WGM10)
        out     TCCR1A,r18
        ldi     r18,(1<<WGM13|1<<WGM12|0<<CS12|0<<CS11|1<<CS10)
        out     TCCR1B,r18
```

```
        ldi     r18,high(9999)
        out     ICR1H,r18
        ldi     r18,low(9999)
        out     ICR1L,r18
        ldi             r22,0
        sts     rr22,r22
        ldi             r23,0
        sts     rr23,r23

        pop     r23
        pop     r22
        pop             r18

        ret
;*************************************************************************************************
*****
Configura_puertos:

        push R16

        ldi             R16,$00
        out             PORTA,R16
        sbi             porta,6
        sbi             porta,7
        ldi             R16,0b11111100      ;Puerto A: A0-A1        ADC
        out             DDRA,R16        ;                       A6-A6   teach

        ldi             R16,0b10010000      ;PD7 y pd4: TEACH
        out             PORTD,R16
```

```
        ldi     R16,0b11110010      ;Puerto
        out     DDRD,R16            ;D5:Motor
                                                ;D2-D3  Interrupciones externas
                                                ;PD0:RX        PD1:TX


        pop     R16
        ret
;*****************************************************************************************************
******************

;*****************************************************************************************************
*****

Configura_ADC:
        push    R16
        ldi     R16,0b01000000              ;01:avcc=5v 1:ajusizq 00000:ADC0 para el
canal0
        out     ADMUX, R16
        ldi     R16,0b11010011    ;1:adc abilitado 1:inicio manual 0:AutoTriggering OFF
10:limpia bandera y desactiva interrupcion de fin de conversion 011:pre=8 frec conv=125K
        out     ADCSRA, R16
        pop     R16
        ret
;*****************************************************************************************************
*****

config_timer_ADC:
        push    r16
        ldi     r16,0b00001100 ;0:foc0 0..1:ctc .00.:sin salida 100:pre=256
        out     TCCR0,r16
        ldi     r16,30    ;OCR0=30 frec timer=125Hz   muestras cada 8ms
        out     OCR0,r16
        ldi     r16,0b00000000 ;, 01....10 desab comparematch timer2 y abilita overflow
timer2..... abilita int comparacion timer0 y desabilita int overflow timer0
```

```
        out     TIMSK,r16
        pop     r16
        ret
;*****************************************************************************************************
*****

config_timer_2:
        push    r16
        ldi     r16,0b00000010 ;0:foc2 0..0:modo normal top $FF  .00.:sin salida
010:pre=8
        out     TCCR2,r16
        pop     r16
        ret
;*****************************************************************************************************
*****

config_interr_exter:
        push    r16
        ldi     r16,0b00001111                  ; ....1111  : INT1 falco subida foto 2
INT0 falnco subida    foto1
        out     MCUCR,r16
        ldi     r16,1
        sts     flanco_foto1,r16
        sts     flanco_foto2,r16
        pop     r16
        ret
;*****************************************************************************************************
*****

;*****************************************************************************
;input: R18, R17, R16 = 24 bit value 0 ... 8000
;output: R22, R21, R20, R19 = 4 digits (ASCII)
;cycle:
;
```

```
bin24_ascii:

      ldi    r22, -1 + '0'
_bcd1:  inc    r22
      subi   r16, low(10000000)       ;-10000000
      sbci   r17, high(10000000)
             sbci   R18, byte3(10000000)
      brcc   _bcd1

      ldi    r22, 10 + '0'
_bcd2:  dec    r22
      subi   r16, low(-1000000)       ;+1000000
      sbci   r17, high(-1000000)
             sbci   r18, byte3(-1000000)
      brcs   _bcd2

      ldi    r21, -1 + '0'
_bcd3:  inc    r21
      subi   r16, low(100000)         ;-100000
      sbci   r17, high(100000)
             sbci   r18, byte3(100000)
      brcc   _bcd3

      ldi    r20, 10 + '0'
_bcd4:  dec    r20
      subi   r16, low(-10000)           ;+10000
             sbci   r17, high(-10000)
             sbci   r18, byte3(-10000)
      brcs   _bcd4
```

```
        ldi     r19, -1 + '0'
_bcd5:  inc     r19
        subi    r16, low(1000)          ;-1000
                sbci    r17, high(1000)
                sbci    r18, byte3(1000)
        brcc    _bcd5

        ret
;***************************************************************************
;***************************************************************************
;***************************************************************************
;***************************************************************************
;***************************************************************************
;***************************************************************************
;***************************************************************************
;***************************************************************************
;*** TAREA 4 - Trasmision y recepcion
;***************************************************************************
Tarea4:

rcall Configurar_USART
ldi r16,0
sts cont_error,r16
Lazo_tarea4:

DoEvents
lds r16,envia_dat_serial
cpi r16,10
```

```
brsh tx_activado
rjmp Lazo_tarea4

tx_activado:
lds r16,dimensiones_nuevas
cpi r16,10
brsh tx_medidas
rjmp Lazo_tarea4

tx_medidas:
lds r16,cont_error
inc r16
sts cont_error,r16
cpi r16,3
brlo tx_correcta
rcall limpiar
printf $c0,error
printf $94,trans

rjmp fin_transmision
tx_correcta:
DoEvents
rcall trans_data

call RxByte
cpi r16,'B'
brne tx_medidas

fin_transmision:
```

```
clr r16
sts dimensiones_nuevas,r16
rjmp Lazo_tarea4



;*****************************************************************************
trans_data:
ldi r20,0
lds r16,alto
rcall convertir_a_ascii
        mov r16,R19
        call TxByte
        eor r20,r16
        mov     r16,R18
        call TxByte
        eor r20,r16
        mov r16,R17
        call TxByte
        eor r20,r16
        ldi       r16,'@'
        call TxByte

lds r16,ancho
rcall convertir_a_ascii
        mov r16,R19
        call TxByte
        eor r20,r16
        mov     r16,R18
        call TxByte
```

```
        eor r20,r16
        mov r16,R17
        call TxByte
        eor r20,r16
        ldi     r16,'@'
        call TxByte

lds r16,largo
rcall convertir_a_ascii
        mov r16,R19
        call TxByte
        eor r20,r16
        mov     r16,R18
        call TxByte
        eor r20,r16
        mov r16,R17
        call TxByte
        eor r20,r16
        ldi     r16,'@'
        call TxByte

        mov r16,r20
        call TxByte
ret
;*****************************************************************************
        ; Espero hasta recibir un dato y lo devuelvo en R16
RxByte:
;DoEvents
        sbis            UCSRA,RXC
```

```
        rjmp    RxByte
        in      R16,UDR
        ret
;***************************************************************************
;***************************************************************************

; Envia R16 por el serial
TxByte:
;DoEvents
        sbis    UCSRA,UDRE
        rjmp    TxByte
        out     UDR,R16
        ret

;***************************************************************************
;**** Configuración del USART
Configurar_USART:
        push    R16

        ldi     R16,high($0C)  ; 9600 bps (fosc= 1MHZ)
        out     UBRRH,R16
        ldi     R16,low($0C)
        out     UBRRL,R16

        ldi     R16,(0<<RXC |0<<TXC | 1<<U2X | 0<<MPCM) ; Velocidad DOBLE (U2X =
1), no multiprocesador
        out     UCSRA,R16
```

```
        ldi     R16,(1<<URSEL | 0<<UMSEL | 0<<UPM1 | 0<<UPM0 | 0<<USBS |
1<<UCSZ1 | 1<<UCSZ0); Comunicación asíncrona, sin paridad, 1bit de parada, 7 bits de
datos

        out     UCSRC,R16


        ldi     R16,(0<<RXCIE | 0<<TXCIE | 0<<UDRIE | 1<<RXEN | 1<<TXEN |
0<<UCSZ2 | 0<<TXB8); Interrupciones y Rx deshabilitadas, Tx habilitada

        out     UCSRB,R16


        pop     R16


        ret

;***********************************************************************


error:  .db "Error de",0,0

trans:  .db "Transmision",0
```

```
'*****************************************************************************

'***** Nombre del programa: Medidor de Volumen

'***** Autor: Juan Diego Jimenez Carpio

'***** Fecha: Junio 2013

'***** Descripción del programa: Mediante una interfaz muestra las medidas en timepo real

'***** de las cajas medidas. Tiene la opcion de exportar los datos a Excel.

'*****************************************************************************

'*****************************************************************************************************
**********

Public Class Form1


'*****************************************************************************************************
**********

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        btnexportar.Visible = False

    End Sub


'*****************************************************************************************************
**********

    Private Sub btnexportar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnexportar.Click


        Dim i As Integer

        Dim nFila As Integer


        Dim AppExcel As Object

        AppExcel = CreateObject("Excel.application")

        With AppExcel

            .Visible = True

            .Workbooks.Add() " Agregamos un Libro Nuevo
```

```
        nFila = 2 " Agregos los titulos a nuestra columnas
        .Cells(nFila, 2) = "N° CAJA      "
        .Cells(nFila, 3) = "ALTO (cm)    "
        .Cells(nFila, 4) = "ANCHO (cm)   "
        .Cells(nFila, 5) = "LARGO (cm)   "
        .Cells(nFila, 6) = "VOLUMEN (cm3)"

        For i = 0 To ListView1.Items.Count - 1
            nFila = nFila + 1
            .Cells(nFila, 2) = ListView1.Items.Item(i).SubItems(0).Text
            .Cells(nFila, 3) = ListView1.Items.Item(i).SubItems(1).Text
            .Cells(nFila, 4) = ListView1.Items.Item(i).SubItems(2).Text
            .Cells(nFila, 5) = ListView1.Items.Item(i).SubItems(3).Text
            .Cells(nFila, 6) = ListView1.Items.Item(i).SubItems(4).Text
        Next

        .range("B2:F2").Font.Bold = True
        .range("B2:F2").Interior.Color = RGB(98, 255, 255)
        .Cells.EntireColumn.AutoFit()

    End With



    End Sub

'*************************************************************************************************
***********

    Private Sub mySerialPort_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles spserial_usb.DataReceived
```

```
CheckForIllegalCrossThreadCalls = False
Dim a As Integer
Dim cont As Integer
Dim num_errores As Integer
Dim datos As String
Dim confirm, xor_ascii As String
Dim term1, term2, term3, termtemp, rept_xor As Integer

datos = spserial_usb.ReadExisting
Dim q() As String = datos.Split("@")

term1 = q(0) Mod 10
termtemp = q(0) \ 10
term2 = termtemp Mod 10
term3 = termtemp \ 10
rept_xor = (term1) Xor (term2) Xor (term3)

term1 = q(1) Mod 10
termtemp = q(1) \ 10
term2 = termtemp Mod 10
term3 = termtemp \ 10
rept_xor = (term1) Xor (term2) Xor (term3) Xor rept_xor

term1 = q(2) Mod 10
termtemp = q(2) \ 10
term2 = termtemp Mod 10
term3 = termtemp \ 10
rept_xor = (term1) Xor (term2) Xor (term3) Xor rept_xor
```

```
xor_ascii = Chr(rept_xor + 48)

If xor_ascii = q(3) Then
    RichTextBox1.Text = q(0) / 10 & " cm"
    RichTextBox2.Text = q(1) / 10 & " cm"
    RichTextBox3.Text = q(2) / 10 & " cm"
    confirm = "B"
    spserial_usb.Write(confirm)

    a = 0
    Dim lvi As New ListViewItem
    cont = RichTextBox4.Text
    cont = cont + 1
    lvi.Text = cont
    RichTextBox4.Text = cont
    lvi.SubItems.Add(q(a) / 10)
    a = a + 1
    lvi.SubItems.Add(q(a) / 10)
    a = a + 1
    lvi.SubItems.Add(q(a) / 10)

    lvi.SubItems.Add((q(a - 1) * q(a - 2) * q(a)) / 1000)

    ListView1.Items.Add(lvi)
    ListView1.EnsureVisible(ListView1.Items.Count - 1)

    btnexportar.Visible = True
Else
    confirm = "M"
```

```
            spserial_usb.Write(confirm)
            num_errores = RichTextBox5.Text
            num_errores = num_errores + 1
            If num_errores >= 3 Then
                MsgBox("Error en la Transmisión")
            End If
        End If

    End Sub

'*************************************************************************************************
***********

    Private Sub Buttonsalir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Buttonsalir.Click
        Me.Close()
    End Sub

'*************************************************************************************************
***********

    Private Sub Buttoninicio_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Buttoninicio.Click
        spserial_usb = My.Computer.Ports.OpenSerialPort("COM4")
        spserial_usb.BaudRate = 9600
        spserial_usb.DataBits = 8
        spserial_usb.StopBits = 1
        spserial_usb.Parity = 0

        ListView1.Visible = True

        ListView1.View = View.Details
        ListView1.GridLines = True
```

```
        ListView1.FullRowSelect = True
        ListView1.HideSelection = False
        ListView1.MultiSelect = False

        'Headings
        ListView1.Columns.Add("N° CAJA", 70)
        ListView1.Columns.Add("ALTO (cm)", 70)
        ListView1.Columns.Add("ANCHO (cm)", 80)
        ListView1.Columns.Add("LARGO (cm)", 80)
        ListView1.Columns.Add("VOLUMEN (cm3)", 100)

        For k = 0 To ListView1.Columns.Count - 1
            ListView1.Columns(k).Width = -2
        Next

    End Sub

'*************************************************************************************************************
***********

End Class
'*************************************************************************************************************
***********
```