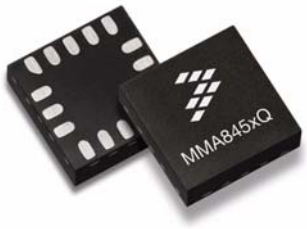
Xtrinsic MMA8451Q 3-Axis, 14-bit/8-bit Digital Accelerometer

MMA8451Q

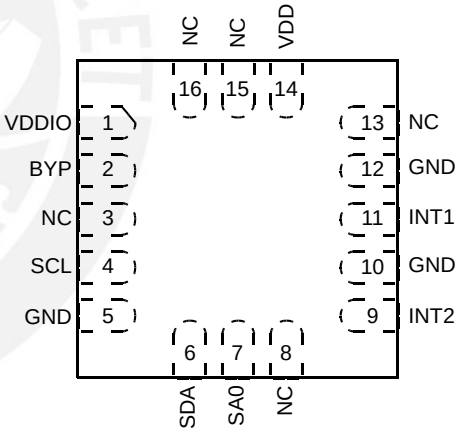
The MMA8451Q is a smart, low-power, three-axis, capacitive, micromachined accelerometer with 14 bits of resolution. This accelerometer is packed with embedded functions with flexible user programmable options, configurable to two interrupt pins. Embedded interrupt functions allow for overall power savings relieving the host processor from continuously polling data. There is access to both low-pass filtered data as well as high-pass filtered data, which minimizes the data analysis required for jolt detection and faster transitions. The device can be configured to generate inertial wakeup interrupt signals from any combination of the configurable embedded functions allowing the MMA8451Q to monitor events and remain in a low-power mode during periods of inactivity. The MMA8451Q is available in a 3 mm by 3 mm by 1 mm QFN package.

Top and Bottom View



16 PIN QFN
3 mm by 3 mm by 1 mm
CASE 2077-02

Top View



Pin Connections

Features

- 1.95V to 3.6V supply voltage
- 1.6V to 3.6V interface voltage
- ±2g/±4g/±8g dynamically selectable full-scale
- Output Data Rates (ODR) from 1.56 Hz to 800 Hz
- 99 µg/√Hz noise
- 14-bit and 8-bit digital output
- I^2C digital output interface (operates to 2.25 MHz with 4.7 kΩ pullup)
- Two programmable interrupt pins for seven interrupt sources
- Three embedded channels of motion detection
- Freefall or Motion Detection: 1 channel
- Pulse Detection: 1 channel
- Jolt Detection: 1 channel
- Orientation (Portrait/Landscape) detection with programmable hysteresis
- Automatic ODR change for Auto-WAKE and return to SLEEP
- 32-sample FIFO
- High-Pass Filter Data available per sample and through the FIFO
- Self-Test
- RoHS compliant
- Current Consumption: 6 µA to 165 µA

Typical Applications

- E-Compass applications
- Static orientation detection (Portrait/Landscape, Up/Down, Left/Right, Back/Front position identification)
- Notebook, e-reader, and Laptop Tumble and Freefall Detection
- Real-time orientation detection (virtual reality and gaming 3D user position feedback)
- Real-time activity analysis (pedometer step counting, freefall drop detection for HDD, dead-reckoning GPS backup)
- Motion detection for portable product power saving (Auto-SLEEP and Auto-WAKE for cell phone, PDA, GPS, gaming)
- Shock and vibration monitoring (mechatronic compensation, shipping and warranty usage logging)
- User interface (menu scrolling by orientation change, tap detection for button replacement)

ORDERING INFORMATION			
Part Number	**Temperature Range**	**Package Description**	**Shipping**
MMA8451QT	-40°C to +85°C	QFN-16	Tray
MMA8451QR1	-40°C to +85°C	QFN-16	Tape and Reel

freescale Xtrinsic

Contents

Related Documentation

The MMA8451Q device features and operations are described in a variety of reference manuals, user guides, and application notes. To find the most-current versions of these documents:

1. Go to the Freescale homepage at:

   http://www.freescale.com/

2. In the Keyword search box at the top of the page, enter the device number MMA8451Q.
3. In the Refine Your Result pane on the left, click on the Documentation link.

1 Block Diagram and Pin Description

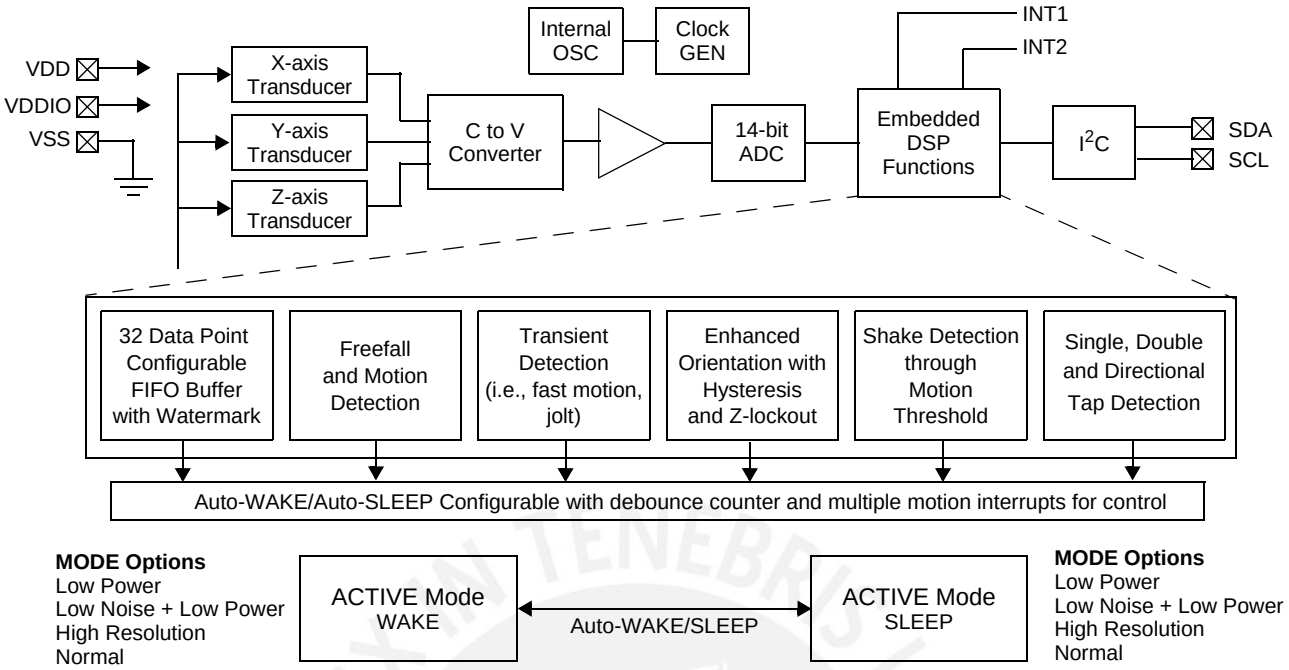


Figure 1. Block Diagram

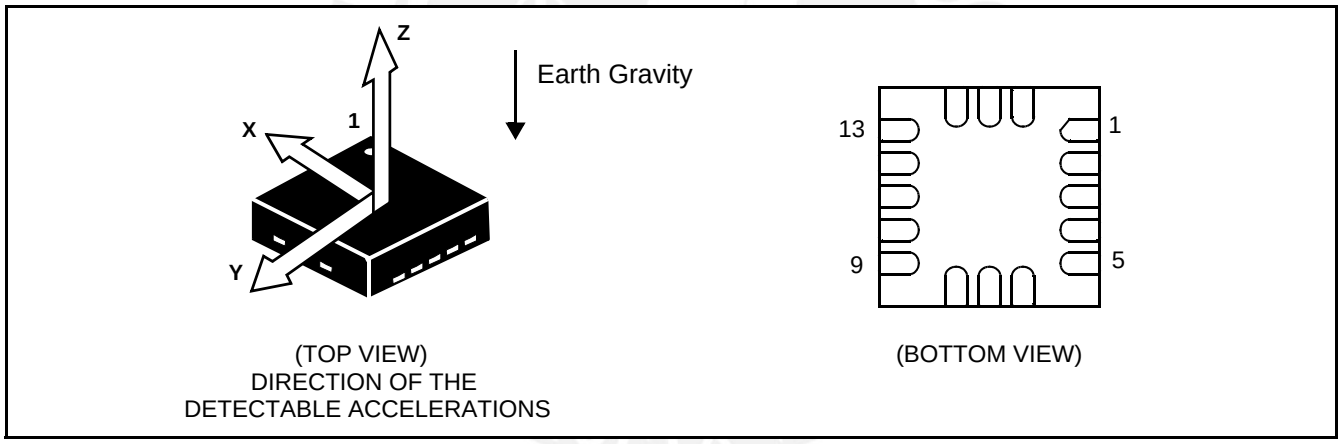


Figure 2. Direction of the Detectable Accelerations

Figure 3 shows the device configuration in the 6 different orientation modes. These orientations are defined as the following: PU = Portrait Up, LR = Landscape Right, PD = Portrait Down, LL = Landscape Left, BACK and FRONT side views. There are several registers to configure the orientation detection and are described in detail in the register setting section.

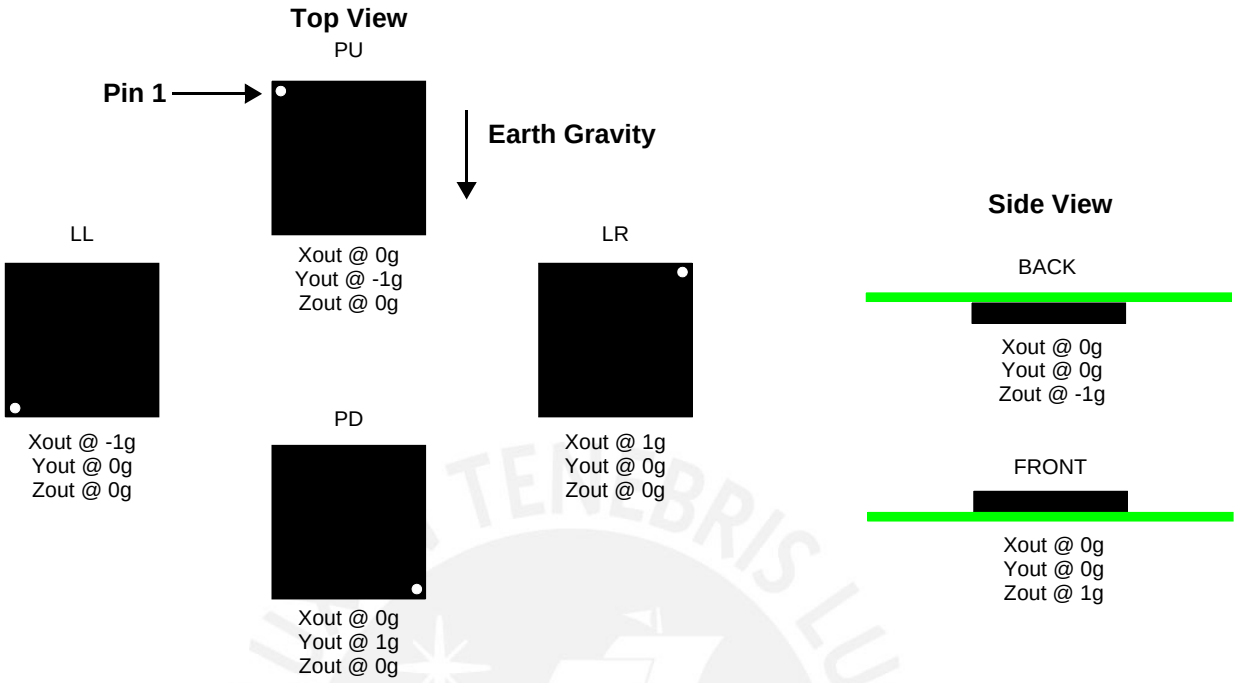


Figure 3. Landscape/Portrait Orientation

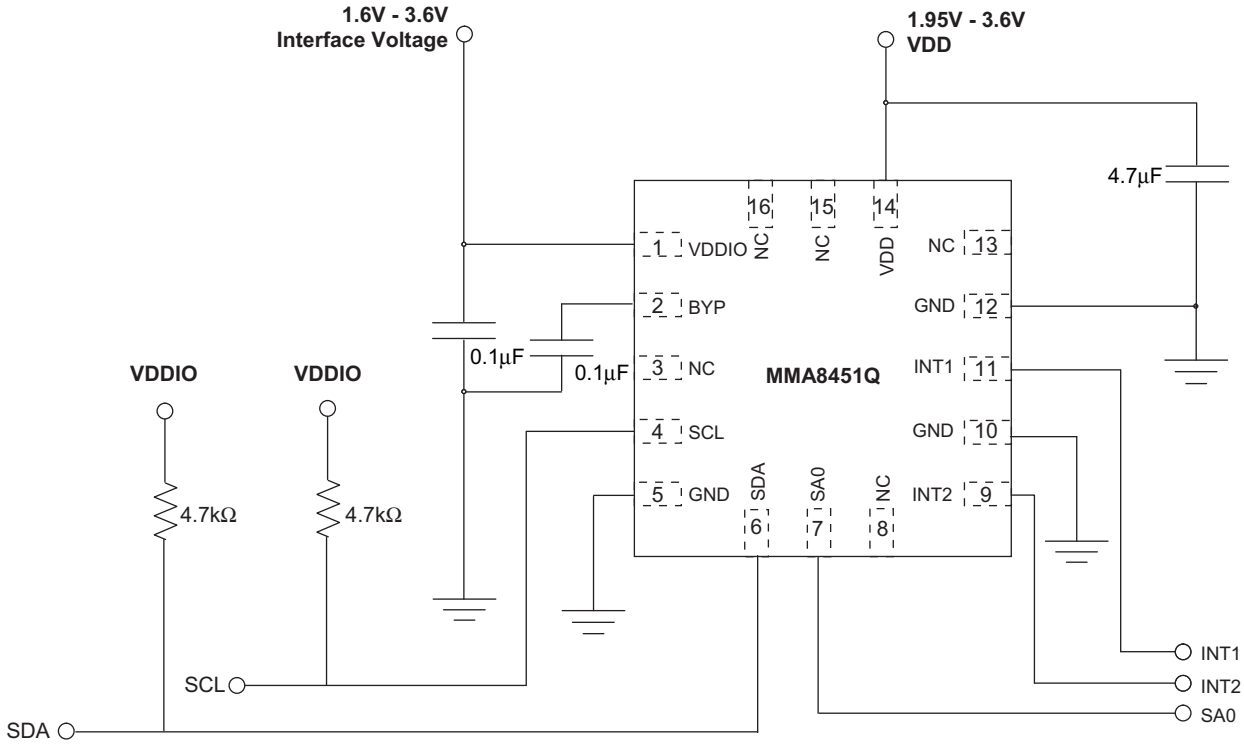


Figure 4. Application Diagram

Table 1. Pin Description

Pin #	Pin Name	Description	Pin Status
1	VDDIO	Internal Power Supply (1.62V - 3.6V)	Input
2	BYP	Bypass capacitor (0.1 µF)	Input
3	NC	Leave open. Do not connect	Open
4	SCL	I^2C Serial Clock	Open Drain
5	GND	Connect to Ground	Input
6	SDA	I^2C Serial Data	Open Drain
7	SA0	I^2C Least Significant Bit of the Device I^2C Address	Input
8	NC	Internally not connected (can be GND or VDD)	Input
9	INT2	Inertial Interrupt 2	Output
10	GND	Connect to Ground	Input
11	INT1	Inertial Interrupt 1	Output
12	GND	Connect to Ground	Input
13	NC	Internally not connected (can be GND or VDD)	Input
14	VDD	Power Supply (1.95V to 3.6V)	Input
15	NC	Internally not connected (can be GND or VDD)	Input
16	NC	Internally not connected (can be GND or VDD)	Input

The device power is supplied through VDD line. Power supply decoupling capacitors (100 nF ceramic plus 4.7 µF bulk, or a single 4.7 µF ceramic) should be placed as near as possible to the pins 1 and 14 of the device.

The control signals SCL, SDA, and SA0 are not tolerant of voltages more than VDDIO + 0.3V. If VDDIO is removed, the control signals SCL, SDA, and SA0 will clamp any logic signals with their internal ESD protection diodes.

The functions, the threshold and the timing of the two interrupt pins (INT1 and INT2) are user programmable through the I^2C interface. The SDA and SCL I^2C connections are open drain and therefore require a pullup resistor as shown in the application diagram in Figure 4.

1.1 Soldering Information

The QFN package is compliant with the RoHS standard. Please refer to AN4077.

2 Mechanical and Electrical Specifications

2.1 Mechanical Characteristics

Table 2. Mechanical Characteristics @ VDD = 2.5V, VDDIO = 1.8V, T = 25°C unless otherwise noted.

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Measurement Range(1)	FS[1:0] set to 00 2g Mode	FS		±2		g
	FS[1:0] set to 01 4g Mode			±4		
	FS[1:0] set to 10 8g Mode			±8		
Sensitivity	FS[1:0] set to 00 2g Mode	So		4096		counts/g
	FS[1:0] set to 01 4g Mode			2048		
	FS[1:0] set to 10 8g Mode			1024		
Sensitivity Accuracy(2)		Soa		±2.64		%
Sensitivity Change vs. Temperature	FS[1:0] set to 00 2g Mode	TCSo		±0.008		%/°C
	FS[1:0] set to 01 4g Mode					
	FS[1:0] set to 10 8g Mode					
Zero-g Level Offset Accuracy(3)	FS[1:0] 2g, 4g, 8g	TyOff		±17		mg
Zero-g Level Offset Accuracy Post Board Mount(4)	FS[1:0] 2g, 4g, 8g	TyOffPBM		±20		mg
Zero-g Level Change vs. Temperature	-40°C to 85°C	TCOff		±0.15		mg/°C
Self-Test Output Change(5) X Y Z	FS[1:0] set to 0 4g Mode	Vst		+181 +255 +1680		LSB
ODR Accuracy 2 MHz Clock				±2		%
Output Data Bandwidth		BW	ODR/3		ODR/2	Hz
Output Noise	Normal Mode ODR = 400 Hz	Noise		126		µg/√Hz
Output Noise Low Noise Mode(1)	Normal Mode ODR = 400 Hz	Noise		99		µg/√Hz
Operating Temperature Range		Top	-40		+85	°C

1. Dynamic Range is limited to 4g when the Low Noise bit in Register 0x2A, bit 2 is set.
2. Sensitivity remains in spec as stated, but changing Oversampling mode to Low Power causes 3% sensitivity shift. This behavior is also seen when changing from 800 Hz to any other data rate in the Normal, Low Noise + Low Power or High Resolution mode.
3. Before board mount.
4. Post Board Mount Offset Specifications are based on an 8 Layer PCB, relative to 25°C.
5. Self-Test is one direction only.

2.2 Electrical Characteristics

Table 3. Electrical Characteristics @ VDD = 2.5V, VDDIO = 1.8V, T = 25°C unless otherwise noted.

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Supply Voltage		VDD(1)	1.95	2.5	3.6	V
Interface Supply Voltage		VDDIO(1)	1.62	1.8	3.6	V
Low Power Mode	ODR = 1.56 Hz	I_{dd}LP		6		µA
	ODR = 6.25 Hz			6		
	ODR = 12.5 Hz			6		
	ODR = 50 Hz			14		
	ODR = 100 Hz			24		
	ODR = 200 Hz			44		
	ODR = 400 Hz			85		
	ODR = 800 Hz			165		
Normal Mode	ODR = 1.56 Hz	I_{dd}		24		µA
	ODR = 6.25 Hz			24		
	ODR = 12.5 Hz			24		
	ODR = 50 Hz			24		
	ODR = 100 Hz			44		
	ODR = 200 Hz			85		
	ODR = 400 Hz			165		
	ODR = 800 Hz			165		
Current during Boot Sequence, 0.5 mSec max duration using recommended Bypass Cap	VDD = 2.5V	Idd Boot			1	mA
Value of Capacitor on BYP Pin	-40°C 85°C	Cap	75	100	470	nF
STANDBY Mode Current @25°C	VDD = 2.5V, VDDIO = 1.8V STANDBY Mode	I_{dd}Stby		1.8	5	µA
Digital High Level Input Voltage SCL, SDA, SA0		VIH	0.75*VDDIO			V
Digital Low Level Input Voltage SCL, SDA, SA0		VIL			0.3*VDDIO	V
High Level Output Voltage INT1, INT2	I_O = 500 µA	VOH	0.9*VDDIO			V
Low Level Output Voltage INT1, INT2	I_O = 500 µA	VOL			0.1*VDDIO	V
Low Level Output Voltage SDA	I_O = 500 µA	VOLS			0.1*VDDIO	V
Power on Ramp Time			0.001		1000	ms
Time from VDDIO on and VDD > Vmin until I^2C ready for operation	Cbyp = 100 nF	BT	—	350	500	µs
Turn-on time (STANDBY to ACTIVE)		Ton		2/ODR + 1 ms		s
Turn-on time (Power Down to ACTIVE Mode)		Ton		2/ODR + 2 ms		s
Operating Temperature Range		Top	-40		+85	°C

1. There is no requirement for power supply sequencing. The VDDIO input voltage can be higher than the VDD input voltage.

2.3 I²C interface characteristics

Table 4. I²C slave timing values(1)

Parameter	Symbol	I²C Fast Mode		Unit
		Min	Max	
SCL clock frequency	f_{SCL}	0	400	kHz
Bus-free time between STOP and START condition	t_{BUF}	1.3		µs
(Repeated) START hold time	$t_{HD;STA}$	0.6		µs
Repeated START setup time	$t_{SU;STA}$	0.6		µs
STOP condition setup time	$t_{SU;STO}$	0.6		µs
SDA data hold time	$t_{HD;DAT}$	0.05	0.9 (2)	µs
SDA setup time	$t_{SU;DAT}$	100		ns
SCL clock low time	t_{LOW}	1.3		µs
SCL clock high time	t_{HIGH}	0.6		µs
SDA and SCL rise time	t_r	20 + 0.1 C_b(3)	300	ns
SDA and SCL fall time	t_f	20 + 0.1 C_b(3)	300	ns
SDA valid time (4)	$t_{VD;DAT}$		0.9 (2)	µs
SDA valid acknowledge time (5)	$t_{VD;ACK}$		0.9 (2)	µs
Pulse width of spikes on SDA and SCL that must be suppressed by internal input filter	t_{SP}	0	50	ns
Capacitive load for each bus line	Cb		400	pF

1.All values referred to $V_{IH(min)}$ ($0.3V_{DD}$) and $V_{IL(max)}$ ($0.7V_{DD}$) levels.
2.This device does not stretch the LOW period (t_{LOW}) of the SCL signal.
3.C_b = total capacitance of one bus line in pF.
4.$t_{VD;DAT}$ = time for data signal from SCL LOW to SDA output (HIGH or LOW, depending on which one is worse).
5.$t_{VD;ACK}$ = time for Acknowledgement signal from SCL LOW to SDA output (HIGH or LOW, depending on which one is worse).

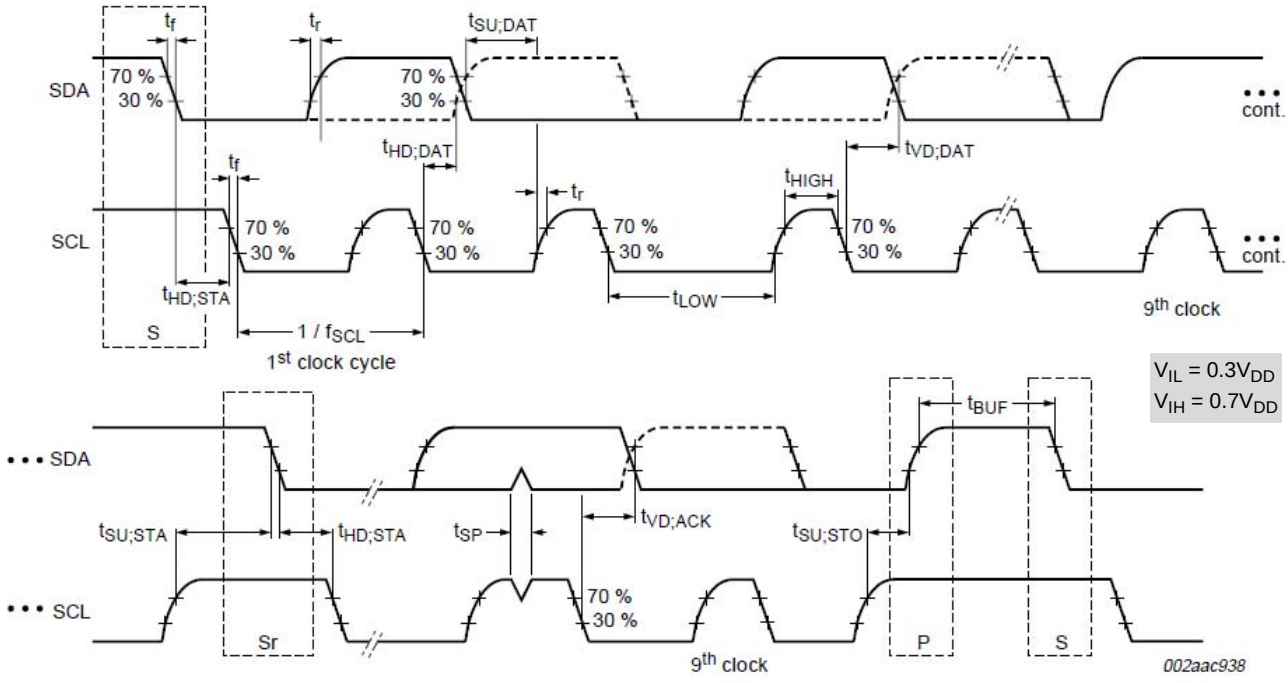


Figure 5. I²C slave timing diagram

2.4 Absolute Maximum Ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 5. Maximum Ratings

Rating	Symbol	Value	Unit
Maximum Acceleration (all axes, 100 μs)	g_{max}	5,000	g
Supply Voltage	VDD	-0.3 to + 3.6	V
Input voltage on any control pin (SA0, SCL, SDA)	Vin	-0.3 to VDDIO + 0.3	V
Drop Test	D_{drop}	1.8	m
Operating Temperature Range	T_{OP}	-40 to +85	°C
Storage Temperature Range	T_{STG}	-40 to +125	°C

Table 6. ESD and Latchup Protection Characteristics

Rating	Symbol	Value	Unit
Human Body Model	HBM	±2000	V
Machine Model	MM	±200	V
Charge Device Model	CDM	±500	V
Latchup Current at T = 85°C	—	±100	mA

This device is sensitive to mechanical shock. Improper handling can cause permanent damage of the part or cause the part to otherwise fail.

This device is sensitive to ESD, improper handling can cause permanent damage to the part.

3 Terminology

3.1 Sensitivity

The sensitivity is represented in counts/g. In 2g mode the sensitivity is 4096 counts/g. In 4g mode the sensitivity is 2048 counts/g and in 8g mode the sensitivity is 1024 counts/g.

3.2 Zero-g Offset

Zero-g Offset (TyOff) describes the deviation of an actual output signal from the ideal output signal if the sensor is stationary. A sensor stationary on a horizontal surface will measure 0g in X-axis and 0g in Y-axis whereas the Z-axis will measure 1g. The output is ideally in the middle of the dynamic range of the sensor (content of OUT Registers 0x00, data expressed as 2's complement number). A deviation from ideal value in this case is called Zero-g offset. Offset is to some extent a result of stress on the MEMS sensor and therefore the offset can slightly change after mounting the sensor onto a printed circuit board or exposing it to extensive mechanical stress.

3.3 Self-Test

Self-Test checks the transducer functionality without external mechanical stimulus. When Self-Test is activated, an electrostatic actuation force is applied to the sensor, simulating a small acceleration. In this case, the sensor outputs will exhibit a change in their DC levels which are related to the selected full scale through the device sensitivity. When Self-Test is activated, the device output level is given by the algebraic sum of the signals produced by the acceleration acting on the sensor and by the electrostatic test-force.

4 Modes of Operation

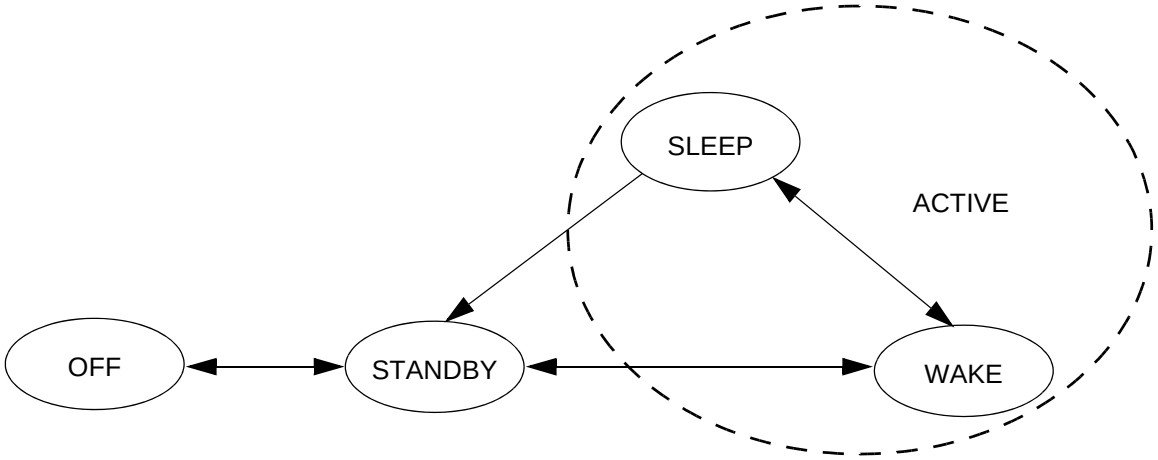


Figure 6. MMA8451Q Mode Transition Diagram

Table 7. Mode of Operation Description

Mode	I^2C Bus State	VDD	VDDIO	Function Description
OFF	Powered Down	<1.8V	VDDIO Can be > VDD	The device is powered off. All analog and digital blocks are shutdown. I^2C bus inhibited.
STANDBY	I^2C communication with MMA8451Q is possible	ON	VDDIO = High VDD = High ACTIVE bit is cleared	Only digital blocks are enabled. Analog subsystem is disabled. Internal clocks disabled.
ACTIVE (WAKE/SLEEP)	I^2C communication with MMA8451Q is possible	ON	VDDIO = High VDD = High ACTIVE bit is set	All blocks are enabled (digital, analog).

All register contents are preserved when transitioning from ACTIVE to STANDBY mode. Some registers are reset when transitioning from STANDBY to ACTIVE. These are all noted in the device memory map register table. The SLEEP and WAKE modes are ACTIVE modes. For more information on how to use the SLEEP and WAKE modes and how to transition between these modes, please refer to the functionality section of this document.

5 Functionality

The MMA8451Q is a low-power, digital output 3-axis linear accelerometer with a I^2C interface and embedded logic used to detect events and notify an external microprocessor over interrupt lines. The functionality includes the following:

- 8-bit or 14-bit data, High-Pass Filtered data, 8-bit or 14-bit configurable 32 sample FIFO
- Four different oversampling options for compromising between resolution and current consumption based on application requirements
- Additional Low Noise mode that functions independently of the Oversampling modes for higher resolution
- Low Power and Auto-WAKE/SLEEP modes for conservation of current consumption
- Single/Double tap with directional information 1 channel
- Motion detection with directional information or Freefall 1 channel
- Transient/Jolt detection based on a high-pass filter and settable threshold for detecting the change in acceleration above a threshold with directional information 1 channel
- Flexible user configurable portrait landscape detection algorithm addressing many use cases for screen orientation

All functionality is available in 2g, 4g or 8g dynamic ranges. There are many configuration settings for enabling all the different functions. Separate application notes have been provided to help configure the device for each embedded functionality.

Table 8. Features of the MMA845xQ devices

Feature List	MMA8451	MMA8452	MMA8453
Digital Resolution (Bits)	14	12	10
Digital Sensitivity (Counts/g)	4096	1024	256
Data-Ready Interrupt	Yes	Yes	Yes
Single-Pulse Interrupt	Yes	Yes	Yes
Double-Pulse Interrupt	Yes	Yes	Yes
Directional-Pulse Interrupt	Yes	Yes	Yes
Auto-WAKE	Yes	Yes	Yes
Auto-SLEEP	Yes	Yes	Yes
Freefall Interrupt	Yes	Yes	Yes
32 Level FIFO	Yes	No	No
High-Pass Filter	Yes	Yes	Yes
Low-Pass Filter	Yes	Yes	Yes
Orientation Detection Portrait/Landscape = 30°, Landscape to Portrait = 60°, and Fixed 45° Threshold	Yes	Yes	Yes
Programmable Orientation Detection	Yes	No	No
Motion Interrupt with Direction	Yes	Yes	Yes
Transient Detection with High-Pass Filter	Yes	Yes	Yes
Low Power Mode	Yes	Yes	Yes

5.1 Device Calibration

The device interface is factory calibrated for sensitivity and Zero-g offset for each axis. The trim values are stored in Non Volatile Memory (NVM). On power-up, the trim parameters are read from NVM and applied to the circuitry. In normal use, further calibration in the end application is not necessary. However, the MMA8451Q allows the user to adjust the Zero-g offset for each axis after power-up, changing the default offset values. The user offset adjustments are stored in 6 volatile registers. For more information on device calibration, refer to Freescale application note, AN4069.

5.2 8-bit or 14-bit Data

The measured acceleration data is stored in the OUT_X_MSB, OUT_X_LSB, OUT_Y_MSB, OUT_Y_LSB, OUT_Z_MSB, and OUT_Z_LSB registers as 2's complement 14-bit numbers. The most significant 8-bits of each axis are stored in OUT_X (Y, Z)_MSB, so applications needing only 8-bit results can use these 3 registers and ignore OUT_X,Y, Z_LSB. To do this, the F_READ bit in CTRL_REG1 must be set. When the F_READ bit is cleared, the fast read mode is disabled.

When the full-scale is set to 2g, the measurement range is -2g to +1.99975g, and each count corresponds to 1g/4096 (0.25 mg) at 14-bits resolution. When the full-scale is set to 8g, the measurement range is -8g to +7.999g, and each count corresponds to 1g/1024 (0.98 mg) at 14-bits resolution. The resolution is reduced by a factor of 64 if only the 8-bit results are used. For more information on the data manipulation between data formats and modes, refer to Freescale application.

5.3 Internal FIFO Data Buffer

MMA8451Q contains a 32 sample internal FIFO data buffer minimizing traffic across the I^2C bus. The FIFO can also provide power savings of the system by allowing the host processor/MCU to go into a SLEEP mode while the accelerometer independently stores the data, up to 32 samples per axis. The FIFO can run at all output data rates. There is the option of accessing the full 14-bit data or for accessing only the 8-bit data. When access speed is more important than high resolution the 8-bit data read is a better option.

The FIFO contains four modes (Fill Buffer Mode, Circular Buffer Mode, Trigger Mode, and Disabled Mode) described in the F_SETUP Register 0x09. Fill Buffer Mode collects the first 32 samples and asserts the overflow flag when the buffer is full and another sample arrives. It does not collect any more data until the buffer is read. This benefits data logging applications where all samples must be collected. The Circular Buffer Mode allows the buffer to be filled and then new data replaces the oldest sample in the buffer. The most recent 32 samples will be stored in the buffer. This benefits situations where the processor is waiting for an specific interrupt to signal that the data must be flushed to analyze the event. The trigger mode will hold the last data up to the point when the trigger occurs and can be set to keep a selectable number of samples after the event occurs.

The MMA8451Q FIFO Buffer has a configurable watermark, allowing the processor to be triggered after a configurable number of samples has filled in the buffer (1 to 32).

For details on the configurations for the FIFO buffer as well as more specific examples and application benefits, refer to Freescale application note, AN4073.

5.4 Low Power Modes vs. High Resolution Modes

The MMA8451Q can be optimized for lower power modes or for higher resolution of the output data. High resolution is achieved by setting the LNOISE bit in Register 0x2A. This improves the resolution but be aware that the dynamic range is limited to 4g when this bit is set. This will affect all internal functions and reduce noise. Another method for improving the resolution of the data is by oversampling. One of the oversampling schemes of the data can activated when MODS = 10 in Register 0x2B which will improve the resolution of the output data only. The highest resolution is achieved at 1.56 Hz.

There is a trade-off between low power and high resolution. Low Power can be achieved when the oversampling rate is reduced. The lowest power is achieved when MODS = 11 or when the sample rate is set to 1.56 Hz. For more information on how to configure the MMA8451Q in Low Power mode or High Resolution mode and to realize the benefits, refer to Freescale application note, AN4075.

5.5 Auto-WAKE/SLEEP Mode

The MMA8451Q can be configured to transition between sample rates (with their respective current consumption) based on four of the interrupt functions of the device. The advantage of using the Auto-WAKE/SLEEP is that the system can automatically transition to a higher sample rate (higher current consumption) when needed but spends the majority of the time in the SLEEP mode (lower current) when the device does not require higher sampling rates. Auto-WAKE refers to the device being triggered by one of the interrupt functions to transition to a higher sample rate. This may also interrupt the processor to transition from a SLEEP mode to a higher power mode.

SLEEP mode occurs after the accelerometer has not detected an interrupt for longer than the user definable time-out period. The device will transition to the specified lower sample rate. It may also alert the processor to go into a lower power mode to save on current during this period of inactivity.

The Interrupts that can WAKE the device from SLEEP are the following: Tap Detection, Orientation Detection, Motion/Freefall, and Transient Detection. The FIFO can be configured to hold the data in the buffer until it is flushed if the FIFO Gate bit is set in Register 0x2C but the FIFO cannot WAKE the device from SLEEP.

The interrupts that can keep the device from falling asleep are the same interrupts that can wake the device with the addition of the FIFO. If the FIFO interrupt is enabled and data is being accessed continually servicing the interrupt then the device will remain in the WAKE mode. Refer to AN4074, for more detailed information for configuring the Auto-WAKE/SLEEP.

5.6 Freefall and Motion Detection

MMA8451Q has flexible interrupt architecture for detecting either a Freefall or a Motion. Freefall can be enabled where the set threshold must be less than the configured threshold, or motion can be enabled where the set threshold must be greater than the threshold. The motion configuration has the option of enabling or disabling a high-pass filter to eliminate tilt data (static offset). The freefall does not use the high-pass filter. For details on the Freefall and Motion detection with specific application examples and recommended configuration settings, refer to Freescale application note, AN4070.

5.6.1 Freefall Detection

The detection of “Freefall” involves the monitoring of the X, Y, and Z axes for the condition where the acceleration magnitude is **below** a user specified threshold for a user definable amount of time. Normally, the usable threshold ranges are between ±100 mg and ±500 mg.

5.6.2 Motion Detection

Motion is often used to simply alert the main processor that the device is currently in use. When the acceleration exceeds a set threshold the motion interrupt is asserted. A motion can be a fast moving shake or a slow moving tilt. This will depend on the threshold and timing values configured for the event. The motion detection function can analyze static acceleration changes or faster jolts. For example, to detect that an object is spinning, all three axes would be enabled with a threshold detection of > 2g. This condition would need to occur for a minimum of 100 ms to ensure that the event wasn't just noise. The timing value is set by a configurable debounce counter. The debounce counter acts like a filter to determine whether the condition exists for configurable set of time (i.e., 100 ms or longer). There is also directional data available in the source register to detect the direction of the motion. This is useful for applications such as directional shake or flick, which assists with the algorithm for various gesture detections.

5.7 Transient Detection

The MMA8451Q has a built-in high-pass filter. Acceleration data goes through the high-pass filter, eliminating the offset (DC) and low frequencies. The high-pass filter cutoff frequency can be set by the user to four different frequencies which are dependent on the Output Data Rate (ODR). A higher cutoff frequency ensures the DC data or slower moving data will be filtered out, allowing only the higher frequencies to pass. The embedded Transient Detection function uses the high-pass filtered data allowing the user to set the threshold and debounce counter. The transient detection feature can be used in the same manner as the motion detection by bypassing the high-pass filter. There is an option in the configuration register to do this. This adds more flexibility to cover various customer use cases.

Many applications use the accelerometer’s static acceleration readings (i.e., tilt) which measure the change in acceleration due to gravity only. These functions benefit from acceleration data being filtered with a low-pass filter where high frequency data is considered noise. However, there are many functions where the accelerometer must analyze dynamic acceleration. Functions such as tap, flick, shake and step counting are based on the analysis of the change in the acceleration. It is simpler to interpret these functions dependent on dynamic acceleration data when the static component has been removed. The Transient Detection function can be routed to either interrupt pin through bit 5 in CTRL_REG5 register (0x2E). Registers 0x1D – 0x20 are the dedicated Transient Detection configuration registers. The source register contains directional data to determine the direction of the acceleration, either positive or negative. For details on the benefits of the embedded Transient Detection function along with specific application examples and recommended configuration settings, please refer to Freescale application note, AN4071.

5.8 Tap Detection

The MMA8451Q has embedded single/double and directional tap detection. This function has various customizing timers for setting the pulse time width and the latency time between pulses. There are programmable thresholds for all three axes. The tap detection can be configured to run through the high-pass filter and also through a low-pass filter, which provides more customizing and tunable tap detection schemes. The status register provides updates on the axes where the event was detected and the direction of the tap. For more information on how to configure the device for tap detection please refer to Freescale application note, AN4072.

5.9 Orientation Detection

The MMA8451Q incorporates an advanced algorithm for orientation detection (ability to detect all 6 orientations) with configurable trip points. The embedded algorithm allows the selection of the mid point with the desired hysteresis value.

The MMA8451Q Orientation Detection algorithm confirms the reliability of the function with a configurable Z-lockout angle. Based on known functionality of linear accelerometers, it is not possible to rotate the device about the Z-axis to detect change in acceleration at slow angular speeds. The angle at which the device no longer detects the orientation change is referred to as the "Z-Lockout angle". The device operates down to 14° from the flat position.

For further information on the configuration settings of the orientation detection function, including recommendations for configuring the device to support various application use cases, refer to Freescale application note, AN4068.

Figure 8 shows the definitions of the trip angles going from Landscape to Portrait (A) and then also from Portrait to Landscape (B).

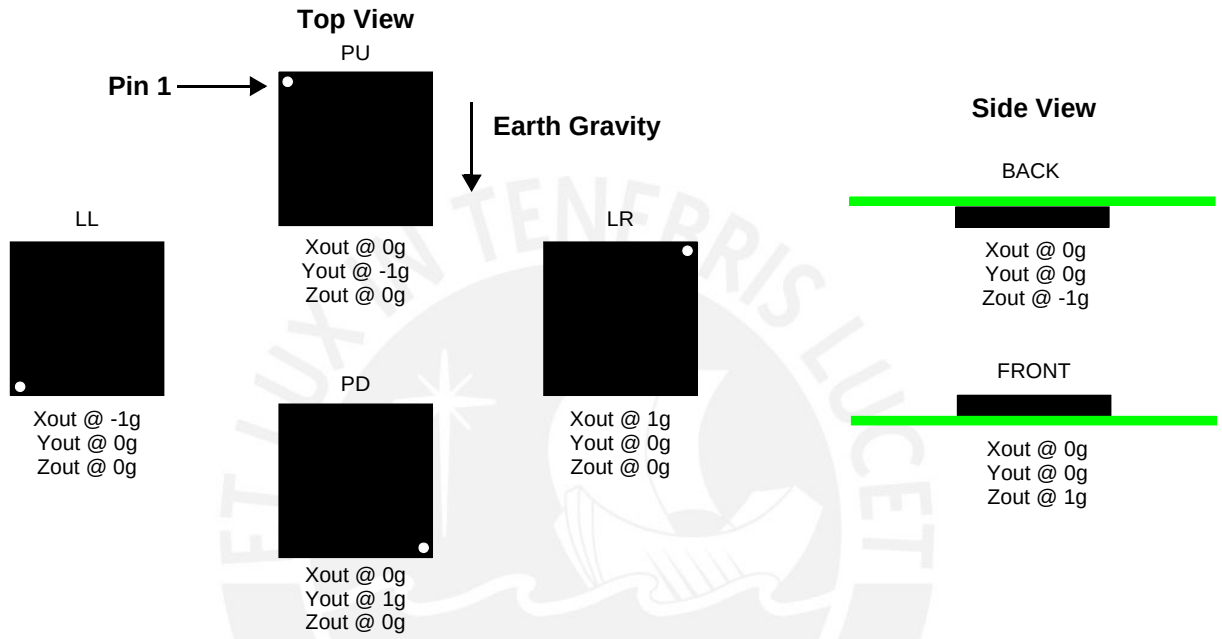


Figure 7. Landscape/Portrait Orientation

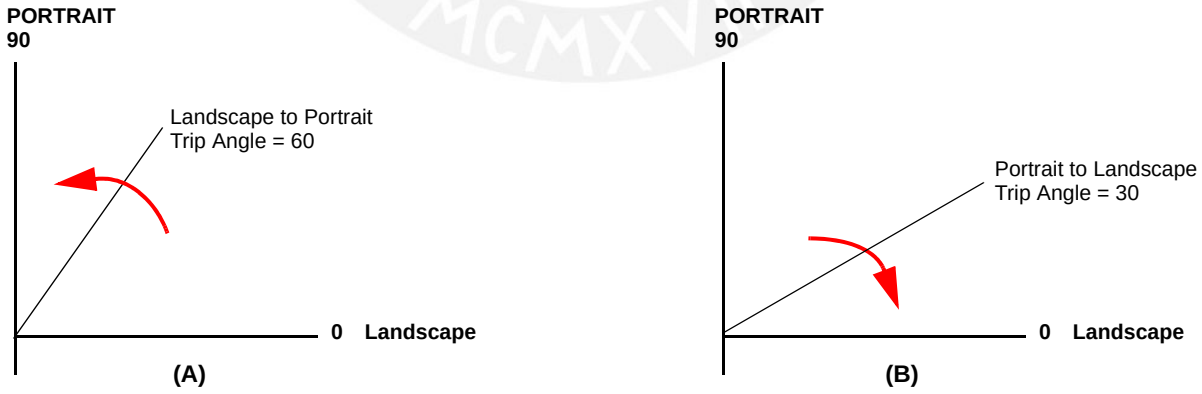


Figure 8. Illustration of Landscape to Portrait Transition (A) and Portrait to Landscape Transition (B)

Figure 9 illustrates the Z-angle lockout region. When lifting the device upright from the flat position it will be active for orientation detection as low as14° from flat. This is user configurable. The default angle is 29° but it can be set as low as 14°.

.

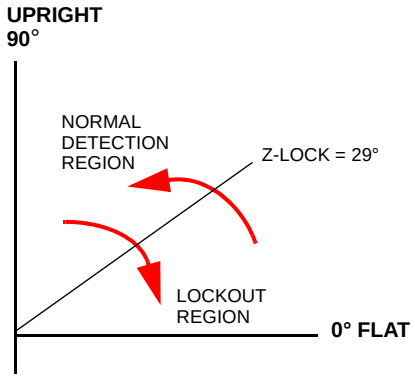


Figure 9. Illustration of Z-Tilt Angle Lockout Transition

5.10 Interrupt Register Configurations

There are seven configurable interrupts in the MMA8451Q: Data Ready, Motion/Freefall, Tap (Pulse), Orientation, Transient, FIFO and Auto-SLEEP events. These seven interrupt sources can be routed to one of two interrupt pins. The interrupt source must be enabled and configured. If the event flag is asserted because the event condition is detected, the corresponding interrupt pin, INT1 or INT2, will assert.

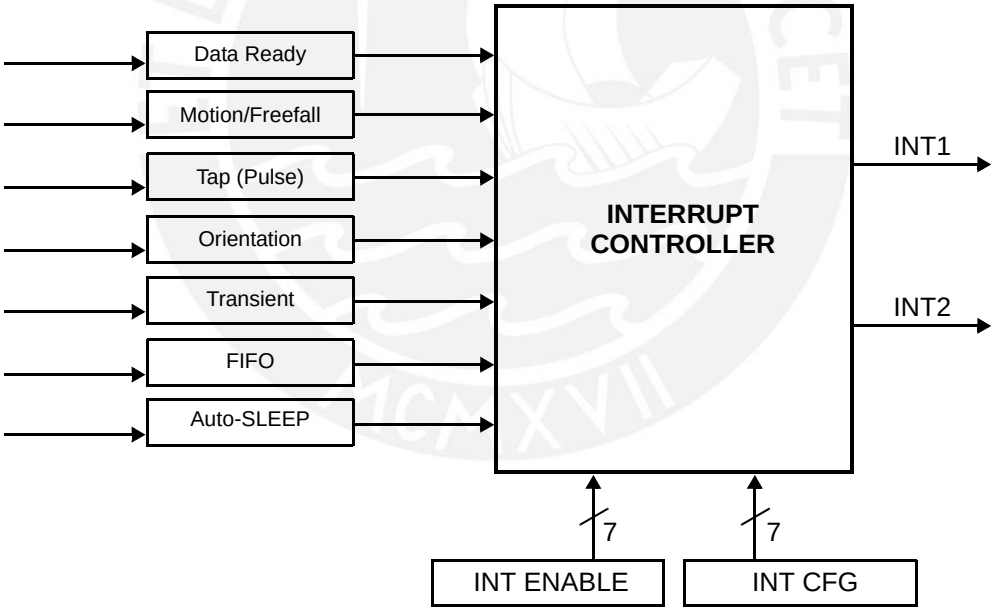


Figure 10. System Interrupt Generation Block Diagram

5.11 Serial I^2C Interface

Acceleration data may be accessed through an I^2C interface thus making the device particularly suitable for direct interfacing with a microcontroller. The MMA8451Q features an interrupt signal which indicates when a new set of measured acceleration data is available thus simplifying data synchronization in the digital system that uses the device. The MMA8451Q may also be configured to generate other interrupt signals accordingly to the programmable embedded functions of the device for Motion, Freefall, Transient, Orientation, and Tap.

The registers embedded inside the MMA8451Q are accessed through the I^2C serial interface (Table 9). To enable the I^2C interface, VDDIO line must be tied high (i.e., to the interface supply voltage). If VDD is not present and VDDIO is present, the

MMA8451Q is in off mode and communications on the I²C interface are ignored. The I²C interface may be used for communications between other I²C devices and the MMA8451Q does not affect the I²C bus.

Table 9. Serial Interface Pin Description

Pin Name	Pin Description
SCL	I²C Serial Clock
SDA	I²C Serial Data
SA0	I²C least significant bit of the device address

There are two signals associated with the I²C bus; the Serial Clock Line (SCL) and the Serial Data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. External pullup resistors connected to VDDIO are expected for SDA and SCL. When the bus is free both the lines are high. The I²C interface is compliant with Fast mode (400 kHz), and Normal mode (100 kHz) I²C standards (Table 4).

5.11.1 I²C Operation

The transaction on the bus is started through a start condition (START) signal. START condition is defined as a HIGH to LOW transition on the data line while the SCL line is held HIGH. After START has been transmitted by the Master, the bus is considered busy. The next byte of data transmitted after START contains the slave address in the first 7 bits, and the eighth bit tells whether the Master is receiving data from the slave or transmitting data to the slave. When an address is sent, each device in the system compares the first seven bits after a start condition with its address. If they match, the device considers itself addressed by the Master. The 9th clock pulse, following the slave address byte (and each subsequent byte) is the acknowledge (ACK). The transmitter must release the SDA line during the ACK period. The receiver must then pull the data line low so that it remains stable low during the high period of the acknowledge clock period.

A LOW to HIGH transition on the SDA line while the SCL line is high is defined as a stop condition (STOP). A data transfer is always terminated by a STOP. A Master may also issue a repeated START during a data transfer. The MMA8451Q expects repeated STARTs to be used to randomly read from specific registers.

The MMA8451Q's standard slave address is a choice between the two sequential addresses 0011100 and 0011101. The selection is made by the high and low logic level of the SA0 (pin 7) input respectively. The slave addresses are factory programmed and alternate addresses are available at customer request. The format is shown in Table 10.

Table 10. I²C Address Selection Table

Slave Address (SA0 = 0)	Slave Address (SA0 = 1)	Comment
0011100 (0x1C)	0011101 (0x1D)	Factory Default

Single Byte Read

The MMA8451Q has an internal ADC that can sample, convert and return sensor data on request. The transmission of an 8-bit command begins on the falling edge of SCL. After the eight clock cycles are used to send the command, note that the data returned is sent with the MSB first once the data is received. Figure 11 shows the timing diagram for the accelerometer 8-bit I²C read operation. The Master (or MCU) transmits a start condition (ST) to the MMA8451Q, slave address ($1D), with the R/W bit set to “0” for a write, and the MMA8451Q sends an acknowledgement. Then the Master (or MCU) transmits the address of the register to read and the MMA8451Q sends an acknowledgement. The Master (or MCU) transmits a repeated start condition (SR) and then addresses the MMA8451Q ($1D) with the R/W bit set to “1” for a read from the previously selected register. The Slave then acknowledges and transmits the data from the requested register. The Master does not acknowledge (NAK) the transmitted data, but transmits a stop condition to end the data transfer.

Multiple Byte Read

When performing a multi-byte read or “burst read”, the MMA8451Q automatically increments the received register address commands after a read command is received. Therefore, after following the steps of a single byte read, multiple bytes of data can be read from sequential registers after each MMA8451Q acknowledgment (AK) is received until a no acknowledge (NAK) occurs from the Master followed by a stop condition (SP) signaling an end of transmission.

Single Byte Write

To start a write command, the Master transmits a start condition (ST) to the MMA8451Q, slave address ($1D) with the R/W bit set to “0” for a write, the MMA8451Q sends an acknowledgement. Then the Master (MCU) transmits the address of the register to write to, and the MMA8451Q sends an acknowledgement. Then the Master (or MCU) transmits the 8-bit data to write to the designated register and the MMA8451Q sends an acknowledgement that it has received the data. Since this transmission is

complete, the Master transmits a stop condition (SP) to the data transfer. The data sent to the MMA8451Q is now stored in the appropriate register.

Multiple Byte Write

The MMA8451Q automatically increments the received register address commands after a write command is received. Therefore, after following the steps of a single byte write, multiple bytes of data can be written to sequential registers after each MMA8451Q acknowledgment (ACK) is received.

Table 11. I²C Device Address Sequence

Command	[6:1] Device Address	[0] SA0	[6:0] Device Address	R/W	8-bit Final Value
Read	001110	0	0x1C	1	0x39
Write	001110	0	0x1C	0	0x38
Read	001110	1	0x1D	1	0x3B
Write	001110	1	0x1D	0	0x3A

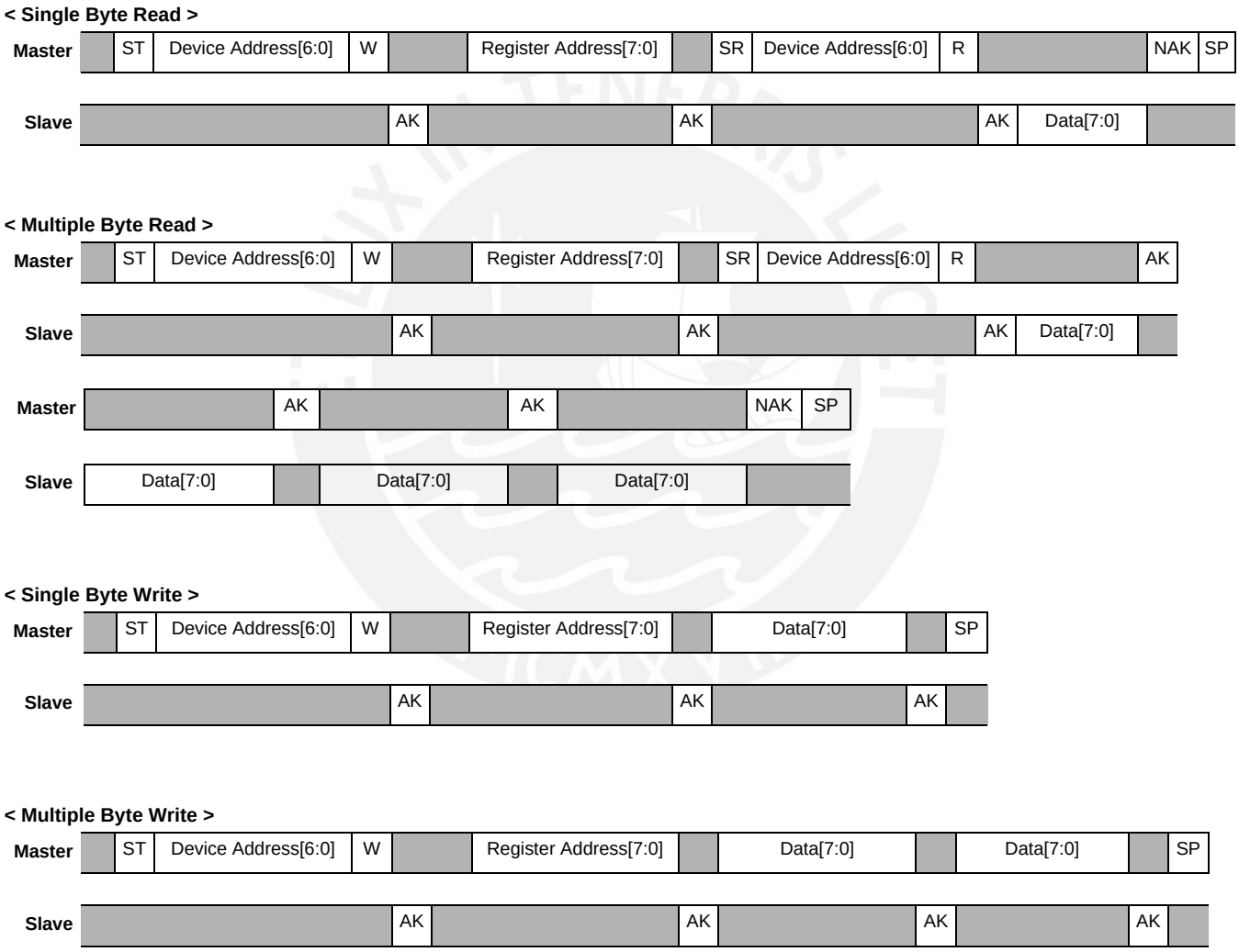


Figure 11. I²C Timing Diagram

6 Register Descriptions

Table 12. Register Address Map

Name	Type	Register Address	Auto-Increment Address				Default	Hex Value	Comment
			FMODE = 0 F_READ = 0	FMODE > 0 F_READ = 0	FMODE = 0 F_READ = 1	FMODE > 0 F_READ = 1			
STATUS/F_STATUS(1)(2)	R	0x00	0x01				00000000	0x00	FMODE = 0, real time status FMODE > 0, FIFO status
OUT_X_MSB(1)(2)	R	0x01	0x02	0x01	**0x03**	**0x01**	Output	—	[7:0] are 8 MSBs of 14-bit sample. / Root pointer to XYZ FIFO data.
OUT_X_LSB(1)(2)	R	0x02	0x03		**0x00**		Output	—	[7:2] are 6 LSBs of 14-bit real-time sample
OUT_Y_MSB(1)(2)	R	0x03	0x04		**0x05**	**0x00**	Output	—	[7:0] are 8 MSBs of 14-bit real-time sample
OUT_Y_LSB(1)(2)	R	0x04	0x05		**0x00**		Output	—	[7:2] are 6 LSBs of 14-bit real-time sample
OUT_Z_MSB(1)(2)	R	0x05	0x06		**0x00**		Output	—	[7:0] are 8 MSBs of 14-bit real-time sample
OUT_Z_LSB(1)(2)	R	0x06	**0x00**				Output	—	[7:2] are 6 LSBs of 14-bit real-time sample
Reserved	R	0x07	—	—	—	—	—	—	Reserved. Read return 0x00.
Reserved	R	0x08	—	—	—	—	—	—	Reserved. Read return 0x00.
F_SETUP(1)(3)	R/W	0x09	0x0A				00000000	0x00	FIFO setup
TRIG_CFG(1)(4)	R/W	0x0A	0x0B				00000000	0x00	Map of FIFO data capture events
SYSMOD(1)(2)	R	0x0B	0x0C				00000000	0x00	Current System Mode
INT_SOURCE(1)(2)	R	0x0C	0x0D				00000000	0x00	Interrupt status
WHO_AM_I(1)	R	0x0D	0x0E				00011010	0x1A	Device ID (0x1A)
XYZ_DATA_CFG(1)(4)	R/W	0x0E	0x0F				00000000	0x00	Dynamic Range Settings
HP_FILTER_CUTOFF(1)(4)	R/W	0x0F	0x10				00000000	0x00	Cutoff frequency is set to 16 Hz @ 800 Hz
PL_STATUS(1)(2)	R	0x10	0x11				00000000	0x00	Landscape/Portrait orientation status
PL_CFG (1)(4)	R/W	0x11	0x12				10000000	0x80	Landscape/Portrait configuration.
PL_COUNT(1)(3)	R/W	0x12	0x13				00000000	0x00	Landscape/Portrait debounce counter
PL_BF_ZCOMP(1)(4)	R/W	0x13	0x14				01000100	0x44	Back/Front, Z-Lock Trip threshold
P_L_THS_REG(1)(4)	R/W	0x14	0x15				10000100	0x84	Portrait to Landscape Trip Angle is 29°
FF_MT_CFG(1)(4)	R/W	0x15	0x16				00000000	0x00	Freefall/Motion functional block configuration
FF_MT_SRC(1)(2)	R	0x16	0x17				00000000	0x00	Freefall/Motion event source register
FF_MT_THS(1)(3)	R/W	0x17	0x18				00000000	0x00	Freefall/Motion threshold register
FF_MT_COUNT(1)(3)	R/W	0x18	0x19				00000000	0x00	Freefall/Motion debounce counter
Reserved	R	0x19	—	—	—	—	—	—	Reserved. Read return 0x00.
Reserved	R	0x1A	—	—	—	—	—	—	Reserved. Read return 0x00.
Reserved	R	0x1B	—	—	—	—	—	—	Reserved. Read return 0x00.
Reserved	R	0x1C	—	—	—	—	—	—	Reserved. Read return 0x00.
TRANSIENT_CFG(1)(4)	R/W	0x1D	0x1E				00000000	0x00	Transient functional block configuration
TRANSIENT_SCR(1)(2)	R	0x1E	0x1F				00000000	0x00	Transient event status register

Table 12. Register Address Map

TRANSIENT_THS[1][3]	R/W	0x1F	0x20	00000000	0x00	Transient event threshold
TRANSIENT_COUNT[1][3]	R/W	0x20	0x21	00000000	0x00	Transient debounce counter
PULSE_CFG[1][4]	R/W	0x21	0x22	00000000	0x00	ELE, Double_XYZ or Single_XYZ
PULSE_SRC[1][2]	R	0x22	0x23	00000000	0x00	EA, Double_XYZ or Single_XYZ
PULSE_THSX[1][3]	R/W	0x23	0x24	00000000	0x00	X pulse threshold
PULSE_THSY[1][3]	R/W	0x24	0x25	00000000	0x00	Y pulse threshold
PULSE_THSZ[1][4]	R/W	0x25	0x26	00000000	0x00	Z pulse threshold
PULSE_TMLT[1][4]	R/W	0x26	0x27	00000000	0x00	Time limit for pulse
PULSE_LTCY[1][4]	R/W	0x27	0x28	00000000	0x00	Latency time for 2nd pulse
PULSE_WIND[1][4]	R/W	0x28	0x29	00000000	0x00	Window time for 2nd pulse
ASLP_COUNT[1][4]	R/W	0x29	0x2A	00000000	0x00	Counter setting for Auto-SLEEP
CTRL_REG1[1][4]	R/W	0x2A	0x2B	00000000	0x00	ODR = 800 Hz, STANDBY Mode.
CTRL_REG2[1][4]	R/W	0x2B	0x2C	00000000	0x00	Sleep Enable, OS Modes, RST, ST
CTRL_REG3[1][4]	R/W	0x2C	0x2D	00000000	0x00	Wake from Sleep, IPOL, PP_OD
CTRL_REG4[1][4]	R/W	0x2D	0x2E	00000000	0x00	Interrupt enable register
CTRL_REG5[1][4]	R/W	0x2E	0x2F	00000000	0x00	Interrupt pin (INT1/INT2) map
OFF_X[1][4]	R/W	0x2F	0x30	00000000	0x00	X-axis offset adjust
OFF_Y[1][4]	R/W	0x30	0x31	00000000	0x00	Y-axis offset adjust
OFF_Z[1][4]	R/W	0x31	**0x0D**	00000000	0x00	Z-axis offset adjust
Reserved (do not modify)		0x40 – 7F	—	—	—	Reserved. Read return 0x00.

1. Register contents are preserved when transition from ACTIVE to STANDBY mode occurs.
2. Register contents are reset when transition from STANDBY to ACTIVE mode occurs.
3. Register contents can be modified anytime in STANDBY or ACTIVE mode. A write to this register will cause a reset of the corresponding internal system debounce counter.
4. Modification of this register's contents can only occur when device is STANDBY mode except CTRL_REG1 ACTIVE bit and CTRL_REG2 RST bit.

Note: Auto-increment addresses which are not a simple increment are highlighted in **bold**. The auto-increment addressing is only enabled when device registers are read using I^2C burst read mode. Therefore the internal storage of the auto-increment address is cleared whenever a stop-bit is detected.

6.1 Data Registers

The following are the data registers for the MMA8451Q. For more information on data manipulation of the MMA8451Q, refer to application note, AN4076.

When the F_MODE bits found in Register 0x09 (F_SETUP), bits 7 and 6 are both cleared (the FIFO is not on). Register 0x00 reflects the real-time status information of the X, Y and Z sample data. When the F_MODE value is greater than zero the FIFO is on (in either Fill, Circular or Trigger mode). In this case Register 0x00 will reflect the status of the FIFO. It is expected when the FIFO is on that the user will access the data from Register 0x01 (X_MSB) for either the 14-bit or 8-bit data. When accessing the 8-bit data the F_READ bit (Register 0x2A) is set which modifies the auto-incrementing to skip over the LSB data. When F_READ bit is cleared the 14-bit data is read accessing all 6 bytes sequentially (X_MSB, X_LSB, Y_MSB, Y_LSB, Z_MSB, Z_LSB).

F_MODE = 00: 0x00 STATUS: Data Status Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ZYXOW	ZOW	YOW	XOW	ZYXDR	ZDR	YDR	XDR

Table 13. STATUS Description

ZYXOW	X, Y, Z-axis Data Overwrite. Default value: 0 0: No data overwrite has occurred 1: Previous X, Y, or Z data was overwritten by new X, Y, or Z data before it was read
ZOW	Z-axis Data Overwrite. Default value: 0 0: No data overwrite has occurred 1: Previous Z-axis data was overwritten by new Z-axis data before it was read
YOW	Y-axis Data Overwrite. Default value: 0 0: No data overwrite has occurred 1: Previous Y-axis data was overwritten by new Y-axis data before it was read
XOW	X-axis Data Overwrite. Default value: 0 0: No data overwrite has occurred 1: Previous X-axis data was overwritten by new X-axis data before it was read
ZYXDR	X, Y, Z-axis new Data Ready. Default value: 0 0: No new set of data ready 1: A new set of data is ready
ZDR	Z-axis new Data Available. Default value: 0 0: No new Z-axis data is ready 1: A new Z-axis data is ready
YDR	Y-axis new Data Available. Default value: 0 0: No new Y-axis data ready 1: A new Y-axis data is ready
XDR	X-axis new Data Available. Default value: 0 0: No new X-axis data ready 1: A new X-axis data is ready

ZYXOW is set whenever a new acceleration data is produced before completing the retrieval of the previous set. This event occurs when the content of at least one acceleration data register (i.e., OUT_X, OUT_Y, OUT_Z) has been overwritten. ZYXOW is cleared when the high-bytes of the acceleration data (OUT_X_MSB, OUT_Y_MSB, OUT_Z_MSB) of all the active channels are read.

ZOW is set whenever a new acceleration sample related to the Z-axis is generated before the retrieval of the previous sample. When this occurs the previous sample is overwritten. ZOW is cleared anytime OUT_Z_MSB register is read.

YOW is set whenever a new acceleration sample related to the Y-axis is generated before the retrieval of the previous sample. When this occurs the previous sample is overwritten. YOW is cleared anytime OUT_Y_MSB register is read.

XOW is set whenever a new acceleration sample related to the X-axis is generated before the retrieval of the previous sample. When this occurs the previous sample is overwritten. XOW is cleared anytime OUT_X_MSB register is read.

ZYXDR signals that a new sample for any of the enabled channels is available. ZYXDR is cleared when the high-bytes of the acceleration data (OUT_X_MSB, OUT_Y_MSB, OUT_Z_MSB) of all the enabled channels are read.

ZDR is set whenever a new acceleration sample related to the Z-axis is generated. ZDR is cleared anytime OUT_Z_MSB register is read.

YDR is set whenever a new acceleration sample related to the Y-axis is generated. YDR is cleared anytime OUT_Y_MSB register is read.

XDR is set whenever a new acceleration sample related to the X-axis is generated. XDR is cleared anytime OUT_X_MSB register is read.

Data Registers: 0x01 OUT_X_MSB, 0x02 OUT_X_LSB, 0x03 OUT_Y_MSB, 0x04 OUT_Y_LSB, 0x05 OUT_Z_MSB, 0x06 OUT_Z_LSB

These registers contain the X-axis, Y-axis, and Z-axis14-bit output sample data expressed as 2's complement numbers.

Note: The sample data output registers store the current sample data if the FIFO data output register driver is disabled, but if the FIFO data output register driver is enabled (F_MODE > 00) the sample data output registers point to the head of the FIFO buffer (Register 0x01 X_MSB) which contains the previous 32 X, Y, and Z data samples. Data Registers F_MODE = 00

0x01: OUT_X_MSB: X_MSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XD13	XD12	XD11	XD10	XD9	XD8	XD7	XD6

0x02: OUT_X_LSB: X_LSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XD5	XD4	XD3	XD2	XD1	XD0	0	0

0x03: OUT_Y_MSB: Y_MSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
YD13	YD12	YD11	YD10	YD9	YD8	YD7	YD6

0x04: OUT_Y_LSB: Y_LSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
YD5	YD4	YD3	YD2	YD1	YD0	0	0

0x05: OUT_Z_MSB: Z_MSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ZD13	ZD12	ZD11	ZD10	ZD9	ZD8	ZD7	ZD6

0x06: OUT_Z_LSB: Z_LSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ZD5	ZD4	ZD3	ZD2	ZD1	ZD0	0	0

OUT_X_MSB, OUT_X_LSB, OUT_Y_MSB, OUT_Y_LSB, OUT_Z_MSB, and OUT_Z_LSB are stored in the auto-incrementing address range of 0x01 to 0x06 to reduce reading the status followed by 14-bit axis data to 7 bytes. If the F_READ bit is set (0x2A bit 1), auto increment will skip over LSB registers. This will shorten the data acquisition from 7 bytes to 4 bytes. The LSB registers can only be read immediately following the read access of the corresponding MSB register. A random read access to the LSB registers is not possible. Reading the MSB register and then the LSB register in sequence ensures that both bytes (LSB and MSB) belong to the same data sample, even if a new data sample arrives between reading the MSB and the LSB byte.

6.2 32 Sample FIFO

The following registers are used to configure the FIFO. For more information on the FIFO please refer to AN4073.

F_MODE > 0 0x00: F_STATUS FIFO Status Register

When F_MODE > 0, Register 0x00 becomes the FIFO Status Register which is used to retrieve information about the FIFO. This register has a flag for the overflow and watermark. It also has a counter that can be read to obtain the number of samples stored in the buffer when the FIFO is enabled.

0x00: F_STATUS: FIFO STATUS Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F_OVF	F_WMRK_FLAG	F_CNT5	F_CNT4	F_CNT3	F_CNT2	F_CNT1	F_CNT0

Table 14. FIFO Flag Event Description

F_OVF	F_WMRK_FLAG	Event Description
0	—	No FIFO overflow events detected.
1	—	FIFO event detected; FIFO has overflowed.
—	0	No FIFO watermark events detected.
—	1	FIFO Watermark event detected. FIFO sample count is greater than watermark value. If F_MODE = 11, Trigger Event detected.

The F_OVF and F_WMRK_FLAG flags remain asserted while the event source is still active, but the user can clear the FIFO interrupt bit flag in the interrupt source register (INT_SOURCE) by reading the F_STATUS register. In this case, the SRC_FIFO bit in the INT_SOURCE register will be set again when the next data sample enters the FIFO. Therefore the F_OVF bit flag will remain asserted while the FIFO has overflowed and the F_WMRK_FLAG bit flag will remain asserted while the F_CNT value is equal to or greater than then F_WMRK value. If the FIFO overflow flag is cleared and if F_MODE = 11 then the FIFO overflow flag will remain 0 before the trigger event even if the FIFO is full and overflows. If the FIFO overflow flag is set and if F_MODE is = 11, the FIFO has stopped accepting samples.

Table 15. FIFO Sample Count Description

F_CNT[5:0]	FIFO sample counter. Default value: 00_0000. (00_0001 to 10_0000 indicates 1 to 32 samples stored in FIFO

F_CNT[5:0] bits indicate the number of acceleration samples currently stored in the FIFO buffer. Count 000000 indicates that the FIFO is empty.

0x09: F_SETUP FIFO Setup Register

0x09 F_SETUP: FIFO Setup Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F_MODE1	F_MODE0	F_WMRK5	F_WMRK4	F_WMRK3	F_WMRK2	F_WMRK1	F_WMRK0

Table 16. F_SETUP Description

BITS	Description
F_MODE[1:0](1)(2)	FIFO buffer overflow mode. Default value: 0. **00:** FIFO is disabled. **01:** FIFO contains the most recent samples when overflowed (circular buffer). Oldest sample is discarded to be replaced by new sample. **10:** FIFO stops accepting new samples when overflowed. **11:** Trigger mode. The FIFO will be in a circular mode up to the number of samples in the watermark. The FIFO will be in a circular mode until the trigger event occurs after that the FIFO will continue to accept samples for 32-WMRK samples and then stop receiving further samples. This allows data to be collected both before and after the trigger event and it is definable by the watermark setting. The FIFO is flushed whenever the FIFO is disabled, during an automatic ODR change (Auto-WAKE/SLEEP), or transitioning from STANDBY mode to ACTIVE mode. Disabling the FIFO (F_MODE = 00) resets the F_OVF, F_WMRK_FLAG, F_CNT to zero. A FIFO overflow event (i.e., F_CNT = 32) will assert the F_OVF flag and a FIFO sample count equal to the sample count watermark (i.e., F_WMRK) asserts the F_WMRK_FLAG event flag.
F_WMRK[5:0](2)	FIFO Event Sample Count Watermark. Default value: 00_0000. These bits set the number of FIFO samples required to trigger a watermark interrupt. A FIFO watermark event flag is raised when FIFO sample count F_CNT[5:0] ≥ F_WMRK[5:0] watermark. Setting the F_WMRK[5:0] to 00_0000 will disable the FIFO watermark event flag generation. Also used to set the number of pre-trigger samples in Trigger mode.

1. Bit field can be written in ACTIVE mode.
2. Bit field can be written in STANDBY mode.

The FIFO mode can be changed while in the active state. The mode must first be disabled F_MODE = 00 then the mode can be switched between Fill mode, Circular mode and Trigger mode.

A FIFO sample count exceeding the watermark event does not stop the FIFO from accepting new data. The FIFO update rate is dictated by the selected system ODR. In ACTIVE mode the ODR is set by the DR bits in the CTRL_REG1 register. When Auto-SLEEP is active the ODR is set by the ASLP_RATE field in the CTRL_REG1 register.

When a byte is read from the FIFO buffer the oldest sample data in the FIFO buffer is returned and also deleted from the front of the FIFO buffer, while the FIFO sample count is decremented by one. It is assumed that the host application shall use the I^2C multi-byte read transaction to empty the FIFO.

0x0A: TRIG_CFG

In the trigger configuration register the bits that are set (logic '1') control which function may trigger the FIFO to its interrupt and conversely bits that are cleared (logic '0') indicate which function has not asserted its interrupt.

The bits set are rising edge sensitive, and are set by a low to high state change and reset by reading the appropriate source register.

0x0A: TRIG_CFG Trigger Configuration Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	Trig_TRANS	Trig_LNDPRT	Trig_PULSE	Trig_FF_MT	—	—

Table 17. Trigger Configuration Description

INT_SOURCE	Description
Trig_TRANS	Transient interrupt trigger bit. Default value: 0
Trig_LNDPRT	Landscape/Portrait Orientation interrupt trigger bit. Default value: 0
Trig_PULSE	Pulse interrupt trigger bit. Default value: 0
Trig_FF_MT	Freefall/Motion trigger bit. Default value: 0

0x0B: SYSMOD System Mode Register

The System mode register indicates the current device operating mode. Applications using the Auto-SLEEP/WAKE mechanism should use this register to synchronize the application with the device operating mode transitions. The System mode register also indicates the status of the FIFO gate error and number of samples since the gate error occurred.

0x0B: SYSMOD: System Mode Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FGERR	FGT_4	FGT_3	FGT_2	FGT_1	FGT_0	SYSMOD1	SYSMOD0

Table 18. SYSMOD Description

FGERR	FIFO Gate Error. Default value: 0. 0: No FIFO Gate Error detected. 1: FIFO Gate Error was detected. Emptying the FIFO buffer clears the FGERR bit in the SYS_MOD register. See section 0x2C: CTRL_REG3 Interrupt Control Register for more information on configuring the FIFO Gate function.
FGT[4:0]	Number of ODR time units since FGERR was asserted. Reset when FGERR Cleared. Default value: 0_0000
SYSMOD[1:0]	System Mode. Default value: 00. 00: STANDBY mode 01: WAKE mode 10: SLEEP mode

0x0C: INT_SOURCE System Interrupt Status Register

In the interrupt source register the status of the various embedded features can be determined. The bits that are set (logic '1') indicate which function has asserted an interrupt and conversely the bits that are cleared (logic '0') indicate which function has not asserted or has deasserted an interrupt. **The bits are set by a low to high transition and are cleared by reading the appropriate interrupt source register.** The SRC_DRDY bit is cleared by reading the X, Y and Z data. It is not cleared by simply reading the Status Register (0x00).

0x0C: INT_SOURCE: System Interrupt Status Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SRC_ASLP	SRC_FIFO	SRC_TRANS	SRC_LNDPRT	SRC_PULSE	SRC_FF_MT	—	SRC_DRDY

Table 19. INT_SOURCE Description

INT_SOURCE	Description
SRC_ASLP	Auto-SLEEP/WAKE interrupt status bit. Default value: 0. Logic '1' indicates that an interrupt event that can cause a WAKE to SLEEP or SLEEP to WAKE system mode transition has occurred. Logic '0' indicates that no WAKE to SLEEP or SLEEP to WAKE system mode transition interrupt event has occurred. **WAKE to SLEEP** transition occurs when no interrupt occurs for a time period that exceeds the user specified limit (ASLP_COUNT). This causes the system to transition to a user specified low ODR setting. **SLEEP to WAKE** transition occurs when the user specified interrupt event has woken the system; thus causing the system to transition to a user specified high ODR setting. Reading the SYSMOD register clears the SRC_ASLP bit.
SRC_FIFO	FIFO interrupt status bit. Default value: 0. Logic '1' indicates that a FIFO interrupt event such as an overflow event or watermark has occurred. Logic '0' indicates that no FIFO interrupt event has occurred. FIFO interrupt event generators: FIFO Overflow, or (Watermark: F_CNT = F_WMRK) and the interrupt has been enabled. This bit is cleared by reading the F_STATUS register.
SRC_TRANS	Transient interrupt status bit. Default value: 0. Logic '1' indicates that an acceleration transient value greater than user specified threshold has occurred. Logic '0' indicates that no transient event has occurred. This bit is asserted whenever **"**EA**"** bit in the TRANS_SRC is asserted and the interrupt has been enabled. This bit is cleared by reading the TRANS_SRC register.
SRC_LNDPRT	Landscape/Portrait Orientation interrupt status bit. Default value: 0. Logic '1' indicates that an interrupt was generated due to a change in the device orientation status. Logic '0' indicates that no change in orientation status was detected. This bit is asserted whenever **"**NEWLP**"** bit in the PL_STATUS is asserted and the interrupt has been enabled. This bit is cleared by reading the PL_STATUS register.
SRC_PULSE	Pulse interrupt status bit. Default value: 0. Logic '1' indicates that an interrupt was generated due to single and/or double pulse event. Logic '0' indicates that no pulse event was detected. This bit is asserted whenever "EA" bit in the PULSE_SRC is asserted and the interrupt has been enabled. This bit is cleared by reading the PULSE_SRC register.
SRC_FF_MT	Freefall/Motion interrupt status bit. Default value: 0. Logic '1' indicates that the Freefall/Motion function interrupt is active. Logic '0' indicates that no Freefall or Motion event was detected. This bit is asserted whenever "EA" bit in the FF_MT_SRC register is asserted and the FF_MT interrupt has been enabled. This bit is cleared by reading the FF_MT_SRC register.
SRC_DRDY	Data Ready Interrupt bit status. Default value: 0. Logic '1' indicates that the X, Y, Z data ready interrupt is active indicating the presence of new data and/or data overrun. Otherwise if it is a logic '0' the X, Y, Z interrupt is not active. This bit is asserted when the ZYXOW and/or ZYXDR is set and the interrupt has been enabled. This bit is cleared by reading the X, Y, and Z data.

0x0D: WHO_AM_I Device ID Register

The device identification register identifies the part. The default value is 0x1A. This value is factory programmed. Consult the factory for custom alternate values.

0x0D: WHO_AM_I Device ID Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	1	0	1	0

0x0E: XYZ_DATA_CFG Register

The XYZ_DATA_CFG register sets the dynamic range and sets the high-pass filter for the output data. When the HPF_OUT bit is set, both the FIFO and DATA registers will contain high-pass filtered data.

0x0E: XYZ_DATA_CFG (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	HPF_OUT	0	0	FS1	FS0

Table 20. XYZ Data Configuration Descriptions

HPF_OUT	Enable High-pass output data 1 = output data High-pass filtered. Default value: 0.
FS[1:0]	Output buffer data format full scale. Default value: 00 (2g).

The default full scale value range is 2g and the high-pass filter is disabled.

Table 21. Full Scale Range

FS1	FS0	Full Scale Range
0	0	2
0	1	4
1	0	8
1	1	Reserved

0x0F: HP_FILTER_CUTOFF High-Pass Filter Register

This register sets the high-pass filter cutoff frequency for removal of the offset and slower changing acceleration data. The output of this filter is indicated by the data registers (0x01-0x06) when bit 4 (HPF_OUT) of Register 0x0E is set. The filter cutoff options change based on the data rate selected as shown in Table 23. For details of implementation on the high-pass filter, refer to Freescale application note, AN4071.

0x0F: HP_FILTER_CUTOFF: High-Pass Filter Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	Pulse_HPF_BYP	Pulse_LPF_EN	0	0	SEL1	SEL0

Table 22. High-Pass Filter Cutoff Register Descriptions

Pulse_HPF_BYP	Bypass High-Pass Filter (HPF) for Pulse Processing Function. 0: HPF enabled for Pulse Processing, 1: HPF Bypassed for Pulse Processing Default value: 0.
Pulse_LPF_EN	Enable Low-Pass Filter (LPF) for Pulse Processing Function. 0: LPF disabled for Pulse Processing, 1: LPF Enabled for Pulse Processing Default value: 0.
SEL[1:0]	HPF Cutoff frequency selection. Default value: 00 (see Table 23).

Table 23. High-Pass Filter Cutoff Options

Oversampling Mode = Normal									
SEL1	SEL0	800 Hz	400 Hz	200 Hz	100 Hz	50 Hz	12.5 Hz	6.25 Hz	1.56 Hz
0	0	16 Hz	16 Hz	8 Hz	4 Hz	2 Hz	2 Hz	2 Hz	2 Hz
0	1	8 Hz	8 Hz	4 Hz	2 Hz	1 Hz	1 Hz	1 Hz	1 Hz
1	0	4 Hz	4 Hz	2 Hz	1 Hz	0.5 Hz	0.5 Hz	0.5 Hz	0.5 Hz
1	1	2 Hz	2 Hz	1 Hz	0.5 Hz	0.25 Hz	0.25 Hz	0.25 Hz	0.25 Hz
Oversampling Mode = Low Noise Low Power									
0	0	16 Hz	16 Hz	8 Hz	4 Hz	2 Hz	0.5 Hz	0.5 Hz	0.5 Hz
0	1	8 Hz	8 Hz	4 Hz	2 Hz	1 Hz	0.25 Hz	0.25 Hz	0.25 Hz
1	0	4 Hz	4 Hz	2 Hz	1 Hz	0.5 Hz	0.125 Hz	0.125 Hz	0.125 Hz
1	1	2 Hz	2 Hz	1 Hz	0.5 Hz	0.25 Hz	0.063 Hz	0.063 Hz	0.063 Hz
Oversampling Mode = High Resolution									
0	0	16 Hz	16 Hz	16 Hz	16 Hz	16 Hz	16 Hz	16 Hz	16 Hz
0	1	8 Hz	8 Hz	8 Hz	8 Hz	8 Hz	8 Hz	8 Hz	8 Hz
1	0	4 Hz	4 Hz	4 Hz	4 Hz	4 Hz	4 Hz	4 Hz	4 Hz
1	1	2 Hz	2 Hz	2 Hz	2 Hz	2 Hz	2 Hz	2 Hz	2 Hz
Oversampling Mode = Low Power									
0	0	16 Hz	8 Hz	4 Hz	2 Hz	1 Hz	0.25 Hz	0.25 Hz	0.25 Hz
0	1	8 Hz	4 Hz	2 Hz	1 Hz	0.5 Hz	0.125 Hz	0.125 Hz	0.125 Hz
1	0	4 Hz	2 Hz	1 Hz	0.5 Hz	0.25 Hz	0.063 Hz	0.063 Hz	0.063 Hz
1	1	2 Hz	1 Hz	0.5 Hz	0.25 Hz	0.125 Hz	0.031 Hz	0.031 Hz	0.031 Hz

6.3 Portrait/Landscape Embedded Function Registers

For more details on the meaning of the different user configurable settings and for example code refer to Freescale application note, AN4068.

0x10: PL_STATUS Portrait/Landscape Status Register

This status register can be read to get updated information on any change in orientation by reading Bit 7, or on the specifics of the orientation by reading the other bits. For further understanding of Portrait Up, Portrait Down, Landscape Left, Landscape Right, Back and Front orientations please refer to Figure 3. The interrupt is cleared when reading the PL_STATUS register.

0x10: PL_STATUS Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NEWLP	LO	—	—	—	LAPO[1]	LAPO[0]	BAFRO

Table 24. PL_STATUS Register Description

NEWLP	Landscape/Portrait status change flag. Default value: 0. 0: No change, 1: BAFRO and/or LAPO and/or Z-Tilt lockout value has changed
LO	Z-Tilt Angle Lockout. Default value: 0. 0: Lockout condition has not been detected. 1: Z-Tilt lockout trip angle has been exceeded. Lockout has been detected.
LAPO[1:0](1)	Landscape/Portrait orientation. Default value: 00 00: Portrait Up: Equipment standing vertically in the normal orientation 01: Portrait Down: Equipment standing vertically in the inverted orientation 10: Landscape Right: Equipment is in landscape mode to the right 11: Landscape Left: Equipment is in landscape mode to the left.
BAFRO	Back or Front orientation. Default value: 0 0: Front: Equipment is in the front facing orientation. 1: Back: Equipment is in the back facing orientation.

1. *The default power up state is BAFRO = 0, LAPO = 0, and LO = 0.*

NEWLP is set to 1 after the first orientation detection after a STANDBY to ACTIVE transition, and whenever a change in LO, BAFRO, or LAPO occurs. NEWLP bit is cleared anytime PL_STATUS register is read. The Orientation mechanism state change is limited to a maximum 1.25g. LAPO BAFRO and LO continue to change when NEWLP is set. The current position is locked if the absolute value of the acceleration experienced on any of the three axes is greater than 1.25g.

0x11: Portrait/Landscape Configuration Register

This register enables the Portrait/Landscape function and sets the behavior of the debounce counter.

0x11: PL_CFG Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBCNTM	PL_EN	—	—	—	—	—	—

Table 25. PL_CFG Description

DBCNTM	Debounce counter mode selection. Default value: 1 0: Decrements debounce whenever condition of interest is no longer valid. 1: Clears counter whenever condition of interest is no longer valid.
PL_EN	Portrait/Landscape Detection Enable. Default value: 0 0: Portrait/Landscape Detection is Disabled. 1: Portrait/Landscape Detection is Enabled.

0x12: Portrait/Landscape Debounce Counter

This register sets the debounce count for the orientation state transition. The minimum debounce latency is determined by the data rate set by the product of the selected system ODR and PL_COUNT registers. Any transition from WAKE to SLEEP or vice versa resets the internal Landscape/Portrait debounce counter. **Note:** The debounce counter weighting (time step) changes based on the ODR and the Oversampling mode. Table 27 explains the time step value for all sample rates and all Oversampling modes.

0x12: PL_COUNT Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBNCE[7]	DBNCE[6]	DBNCE[5]	DBNCE[4]	DBNCE[3]	DBNCE[2]	DBNCE[1]	DBNCE[0]

Table 26. PL_COUNT Description

DBCNE[7:0]	Debounce Count value. Default value: 0000_0000.

Table 27. PL_COUNT Relationship with the ODR

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.319	0.319	0.319	0.319	1.25	1.25	1.25	1.25
400	0.638	0.638	0.638	0.638	2.5	2.5	2.5	2.5
200	1.28	1.28	0.638	1.28	5	5	2.5	5
100	2.55	2.55	0.638	2.55	10	10	2.5	10
50	5.1	5.1	0.638	5.1	20	20	2.5	20
12.5	5.1	20.4	0.638	20.4	20	80	2.5	80
6.25	5.1	20.4	0.638	40.8	20	80	2.5	160
1.56	5.1	20.4	0.638	40.8	20	80	2.5	160

0x13: PL_BF_ZCOMP Back/Front and Z Compensation Register

The Z-Lock angle compensation bits allow the user to adjust the Z-lockout region from 14° up to 43°. The default Z-lockout angle is set to the default value of 29° upon power up. The Back to Front trip angle is set by default to ±75° but this angle also can be adjusted from a range of 65° to 80° with 5° step increments.

0x13: PL_BF_ZCOMP Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BKFR[1]	BKFR[0]	—	—	—	ZLOCK[2]	ZLOCK[1]	ZLOCK[0]

Table 28. PL_BF_ZCOMP Description

BKFR[7:6]	Back/Front Trip Angle Threshold. Default: **01** ≥ **±75°**. Step size is 5°. Range: ±(65° to 80°).
ZLOCK[2:0]	Z-Lock Angle Threshold. Range is from 14° to 43°. Step size is 4°. Default value: **100** ≥ **29°**. Maximum value: **111** ≥ **43°**.

Note: All angles are accurate to ±2°.

Table 29. Z-Lock Threshold Angles

Z-Lock Value	Threshold Angle
0x00	14°
0x01	18°
0x02	21°
0x03	25°
0x04	29°
0x05	33°
0x06	37°
0x07	42°

Table 30. Back/Front Orientation Definition

BKFR	Back/Front Transition	Front/Back Transition
00	Z < 80° or Z > 280°	Z > 100° and Z < 260°
01	Z < 75° or Z > 285°	Z > 105° and Z < 255°
10	Z < 70° or Z > 290°	Z > 110° and Z < 250°
11	Z < 65° or Z > 295°	Z > 115° and Z < 245°

0x14: P_L_THS_REG Portrait/Landscape Threshold and Hysteresis Register

This register represents the Portrait to Landscape trip threshold register used to set the trip angle for transitioning from Portrait to Landscape and Landscape to Portrait. This register includes a value for the hysteresis.

0x14: P_L_THS_REG Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P_L_THS[4]	P_L_THS[3]	P_L_THS[2]	P_L_THS[1]	P_L_THS[0]	HYS[2]	HYS[1]	HYS[0]

Table 31. P_L_THS_REG Description

P_L_THS[7:3]	Portrait/Landscape trip threshold angle from 15° to 75°. See Table 32 for the values with the corresponding approximate threshold angle. Default value: 1_0000 (45°).
HYS[2:0]	This angle is added to the threshold angle for a smoother transition from Portrait to Landscape and Landscape to Portrait. This angle ranges from 0° to ±24°. The default is 100 (±14°).

Table 32 is a lookup table to set the threshold. This is the center value that will be set for the trip point from Portrait to Landscape and Landscape to Portrait. The default Trip Angle is 45° (0x10). The default hysteresis is ±14°.

Note: The condition THS + HYS > 0 and THS + HYS < 32 must be met in order for Landscape/Portrait detection to work properly. The value of 32 represents the sum of both P_L_THS and HYS register values in decimal. For example, THS angle = 75°, P_L_THS = 25(dec) then max HYS must be set to 6 to meet the condition THS+HYS < 32. To configure correctly the hysteresis (HYS) angle must be smaller than the threshold angle (P_L_THS).

Table 32. Threshold Angle Thresholds Lookup Table

Threshold Angle (approx.)	5-bit Register value
15°	0x07
20°	0x09
30°	0x0C
35°	0x0D
40°	0x0F
45°	**0x10**
55°	0x13
60°	0x14
70°	0x17
75°	0x19

Table 33. Trip Angles with Hysteresis for 45° Angle

Hysteresis Register Value	Hysteresis ± Angle Range	Landscape to Portrait Trip Angle	Portrait to Landscape Trip Angle
0	±0	45°	45°
1	±4	49°	41°
2	±7	52°	38°
3	±11	56°	34°
4	**±14**	**59°**	**31°**

Table 33. Trip Angles with Hysteresis for 45° Angle

5	±17	62°	28°
6	±21	66°	24°
7	±24	69°	21°

6.4 Motion and Freefall Embedded Function Registers

The freefall/motion function can be configured in either Freefall or Motion Detection mode via the **OAE** configuration bit (0x15 bit 6). The freefall/motion detection block can be disabled by setting all three bits ZEFE, YEFE, and XEFE to zero.

Depending on the register bits **ELE** (0x15 bit 7) and **OAE** (0x15 bit 6), each of the freefall and motion detection block can operate in four different modes:

Mode 1: Freefall Detection with ELE = 0, OAE = 0

In this mode, the **EA** bit (0x16 bit 7) indicates a freefall event after the debounce counter is complete. The ZEFE, YEFE, and XEFE control bits determine which axes are considered for the freefall detection. Once the EA bit is set, and DBCNTM = 0, the EA bit can get cleared only after the delay specified by FF_MT_COUNT. This is because the counter is in decrement mode. If DBCNTM = 1, the EA bit is cleared as soon as the freefall condition disappears, and will not be set again before the delay specified by FF_MT_COUNT has passed. Reading the FF_MT_SRC register does not clear the EA bit. The event flags (0x16) ZHE, ZHP, YHE, YHP, XHE, and XHP reflect the motion detection status (i.e. high g event) without any debouncing, provided that the corresponding bits ZEFE, YEFE, and/or XEFE are set.

Mode 2: Freefall Detection with ELE = 1, OAE = 0

In this mode, the **EA** event bit indicates a freefall event after the debounce counter. Once the debounce counter reaches the time value for the set threshold, the EA bit is set, and remains set until the FF_MT_SRC register is read. When the FF_MT_SRC register is read, the EA bit and the debounce counter are cleared and a new event can only be generated after the delay specified by FF_MT_CNT. The ZEFE, YEFE, and XEFE control bits determine which axes are considered for the freefall detection. While EA = 0, the event flags ZHE, ZHP, YHE, YHP, XHE, and XHP reflect the motion detection status (i.e., high g event) without any debouncing, provided that the corresponding bits ZEFE, YEFE, and/or XEFE are set. The event flags ZHE, ZHP, YHE, YHP, XHE, and XHP are latched when the EA event bit is set. The event flags ZHE, ZHP, YHE, YHP, XHE, and XHP will start changing only after the FF_MT_SRC register has been read.

Mode 3: Motion Detection with ELE = 0, OAE = 1

In this mode, the **EA** bit indicates a motion event after the debounce counter time is reached. The ZEFE, YEFE, and XEFE control bits determine which axes are taken into consideration for motion detection. Once the **EA** bit is set, and DBCNTM = 0, the EA bit can get cleared only after the delay specified by FF_MT_COUNT. If DBCNTM = 1, the **EA** bit is cleared as soon as the motion high g condition disappears. The event flags ZHE, ZHP, YHE, YHP, XHE, and XHP reflect the motion detection status (i.e., high g event) without any debouncing, provided that the corresponding bits ZEFE, YEFE, and/or XEFE are set. Reading the FF_MT_SRC does not clear any flags, nor is the debounce counter reset.

Mode 4: Motion Detection with ELE = 1, OAE = 1

In this mode, the EA bit indicates a motion event after debouncing. The ZEFE, YEFE, and XEFE control bits determine which axes are taken into consideration for motion detection. Once the debounce counter reaches the threshold, the EA bit is set, and remains set until the FF_MT_SRC register is read. When the FF_MT_SRC register is read, all register bits are cleared and the debounce counter are cleared and a new event can only be generated after the delay specified by FF_MT_CNT. While the bit EA is zero, the event flags ZHE, ZHP, YHE, YHP, XHE, and XHP reflect the motion detection status (i.e., high g event) without any debouncing, provided that the corresponding bits ZEFE, YEFE, and/or XEFE are set. When the EA bit is set, these bits keep their current value until the FF_MT_SRC register is read.

0x15: FF_MT_CFG Freefall/Motion Configuration Register

This is the Freefall/Motion configuration register for setting up the conditions of the freefall or motion function.

0x15: FF_MT_CFG Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ELE	OAE	ZEFE	YEFE	XEFE	—	—	—

Table 34. FF_MT_CFG Description

ELE	Event Latch Enable: Event flags are latched into FF_MT_SRC register. Reading of the FF_MT_SRC register clears the event flag EA and all FF_MT_SRC bits. Default value: 0. 0: Event flag latch disabled; 1: event flag latch enabled
OAE	Motion detect / Freefall detect flag selection. Default value: 0. (Freefall Flag) 0: Freefall Flag (Logical AND combination) 1: Motion Flag (Logical OR combination)
ZEFE	Event flag enable on Z Default value: 0. 0: event detection disabled; 1: raise event flag on measured acceleration value beyond preset threshold
YEFE	Event flag enable on Y event. Default value: 0. 0: Event detection disabled; 1: raise event flag on measured acceleration value beyond preset threshold
XEFE	Event flag enable on X event. Default value: 0. 0: event detection disabled; 1: raise event flag on measured acceleration value beyond preset threshold

OAE bit allows the selection between Motion (logical OR combination) and Freefall (logical AND combination) detection.
ELE denotes whether the enabled event flag will to be latched in the FF_MT_SRC register or the event flag status in the FF_MT_SRC will indicate the real-time status of the event. If ELE bit is set to a logic '1', then the event flags are frozen when the EA bit gets set, and are cleared by reading the FF_MT_SRC source register.
ZHFE, YEFE, XEFE enable the detection of a motion or freefall event when the measured acceleration data on X, Y, Z channel is beyond the threshold set in FF_MT_THS register. If the ELE bit is set to logic '1' in the FF_MT_CFG register new event flags are blocked from updating the FF_MT_SRC register.
FF_MT_THS is the threshold register used to detect freefall motion events. The unsigned 7-bit **FF_MT_THS** threshold register holds the threshold for the freefall detection where the magnitude of the X and Y and Z acceleration values is lower or equal than the threshold value. Conversely, the **FF_MT_THS** also holds the threshold for the motion detection where the magnitude of the X or Y or Z acceleration value is higher than the threshold value.

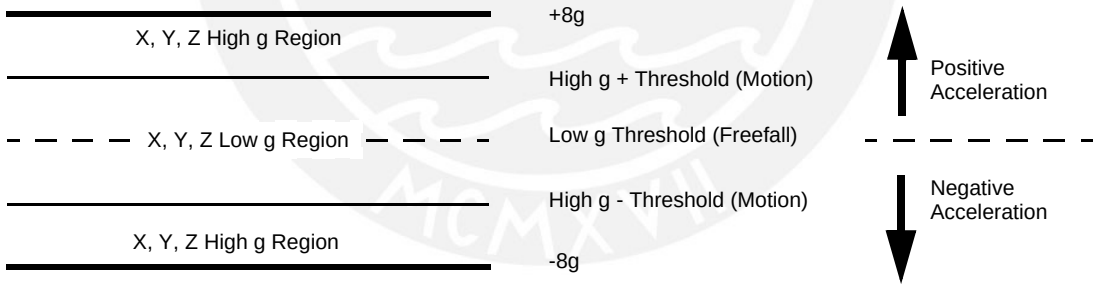


Figure 12. FF_MT_CFG High and Low g Level

0x16: FF_MT_SRC Freefall/Motion Source Register

0x16: FF_MT_SRC Freefall and Motion Source Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	—	ZHE	ZHP	YHE	YHP	XHE	XHP

Table 35. Freefall/Motion Source Description

EA	Event Active Flag. Default value: 0. 0: No event flag has been asserted; 1: one or more event flag has been asserted. See the description of the OAE bit to determine the effect of the 3-axis event flags on the EA bit.
ZHE	Z Motion Flag. Default value: 0. 0: No Z Motion event detected, 1: Z Motion has been detected This bit reads always zero if the ZEFE control bit is set to zero
ZHP	Z Motion Polarity Flag. Default value: 0. 0: Z event was Positive g, 1: Z event was Negative g This bit read always zero if the ZEFE control bit is set to zero
YHE	Y Motion Flag. Default value: 0. 0: No Y Motion event detected, 1: Y Motion has been detected This bit read always zero if the YEFE control bit is set to zero
YHP	Y Motion Polarity Flag. Default value: 0 0: Y event detected was Positive g, 1: Y event was Negative g This bit reads always zero if the YEFE control bit is set to zero
XHE	X Motion Flag. Default value: 0 0: No X Motion event detected, 1: X Motion has been detected This bit reads always zero if the XEFE control bit is set to zero
XHP	X Motion Polarity Flag. Default value: 0 0: X event was Positive g, 1: X event was Negative g This bit reads always zero if the XEFE control bit is set to zero

This register keeps track of the acceleration event which is triggering (or has triggered, in case of ELE bit in FF_MT_CFG register being set to 1) the event flag. In particular EA is set to a logic ‘1’ when the logical combination of acceleration events flags specified in FF_MT_CFG register is true. This bit is used in combination with the values in INT_EN_FF_MT and INT_CFG_FF_MT register bits to generate the freefall/motion interrupts.

An X,Y, or Z motion is true when the acceleration value of the X or Y or Z channel is higher than the preset threshold value defined in the FF_MT_THS register.

Conversely an X, Y, and Z low event is true when the acceleration value of the X and Y and Z channel is lower than or equal to the preset threshold value defined in the FF_MT_THS register.

0x17: FF_MT_THS Freefall and Motion Threshold Register

0x17: FF_MT_THS Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBCNTM	THS6	THS5	THS4	THS3	THS2	THS1	THS0

Table 36. FF_MT_THS Description

DBCNTM	Debounce counter mode selection. Default value: 0. 0: increments or decrements debounce, 1: increments or clears counter.
THS[6:0]	Freefall /Motion Threshold: Default value: 000_0000.

The threshold resolution is 0.063g/LSB and the threshold register has a range of 0 to 127 counts. The maximum range is to 8g. Note that even when the full scale value is set to 2g or 4g the motion detects up to 8g. If the Low Noise bit is set in Register 0x2A then the maximum threshold will be limited to 4g regardless of the full scale range.

DBCNTM bit configures the way in which the debounce counter is reset when the inertial event of interest is momentarily not true.

When DBCNTM bit is a logic ‘1’, the debounce counter is cleared to 0 whenever the inertial event of interest is no longer true as shown in Figure 13, (b). While the DBCNTM bit is set to logic ‘0’ the debounce counter is decremented by 1 whenever the inertial event of interest is no longer true (Figure 13, (c)) until the debounce counter reaches 0 or the inertial event of interest becomes active.

Decrementing the debounce counter acts as a median enabling the system to filter out irregular spurious events which might impede the detection of inertial events.

0x18: FF_MT_COUNT Debounce Register

This register sets the number of debounce sample counts for the event trigger.

0x18: FF_MT_COUNT_Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

Table 37. FF_MT_COUNT Description

D[7:0]	Count value. Default value: 0000_0000

This register sets the minimum number of debounce sample counts of continuously matching the detection condition user selected for the freefall, motion event.

When the internal debounce counter reaches the FF_MT_COUNT value a Freefall/Motion event flag is set. The debounce counter will never increase beyond the FF_MT_COUNT value. Time step used for the debounce sample count depends on the ODR chosen and the Oversampling mode as shown in Table 38.

Table 38. FF_MT_COUNT Relationship with the ODR

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.319	0.319	0.319	0.319	1.25	1.25	1.25	1.25
400	0.638	0.638	0.638	0.638	2.5	2.5	2.5	2.5
200	1.28	1.28	0.638	1.28	5	5	2.5	5
100	2.55	2.55	0.638	2.55	10	10	2.5	10
50	5.1	5.1	0.638	5.1	20	20	2.5	20
12.5	5.1	20.4	0.638	20.4	20	80	2.5	80
6.25	5.1	20.4	0.638	40.8	20	80	2.5	160
1.56	5.1	20.4	0.638	40.8	20	80	2.5	160

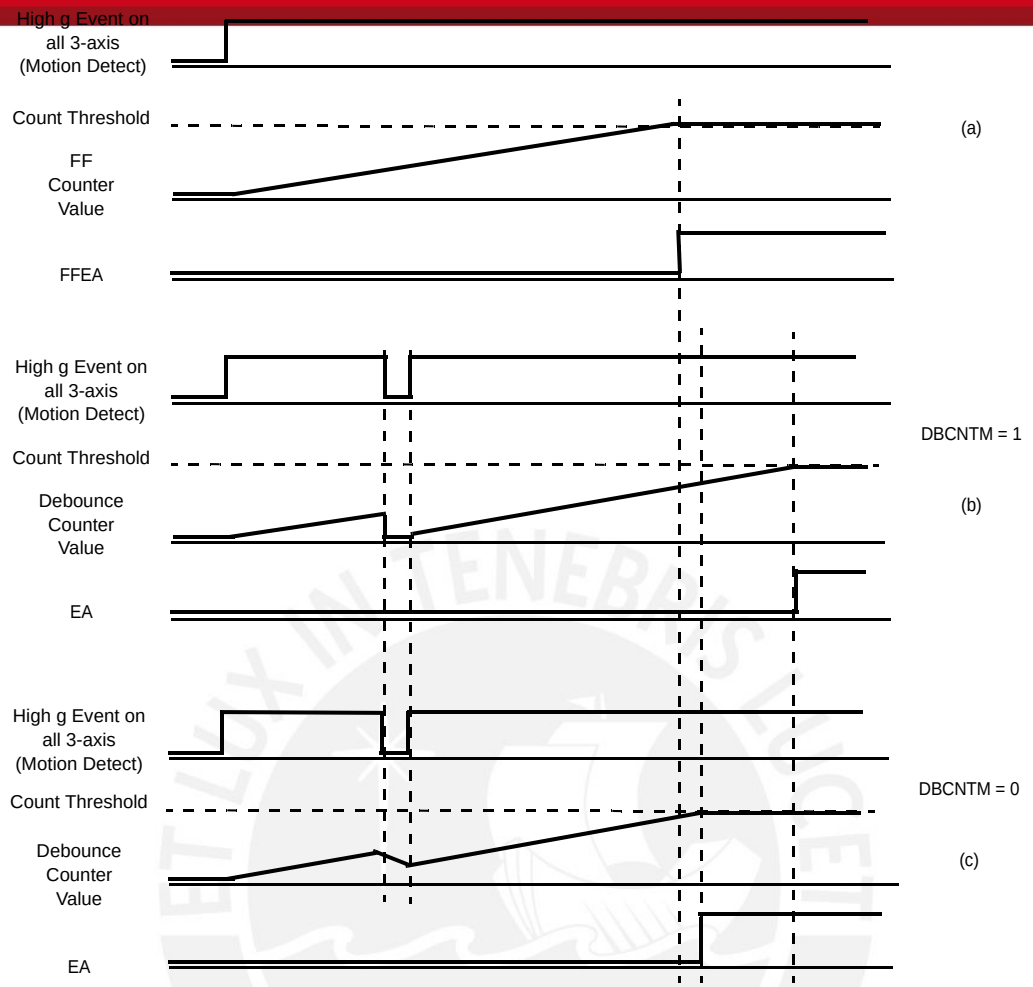


Figure 13. DBCNTM Bit Function

6.5 Transient (HPF) Acceleration Detection

For more information on the uses of the transient function please review Freescale application note, AN4071. This function is similar to the motion detection except that high-pass filtered data is compared. There is an option to disable the high-pass filter through the function. In this case the behavior is the same as the motion detection. This allows for the device to have 2 motion detection functions.

0x1D: Transient_CFG Register

The transient detection mechanism can be configured to raise an interrupt when the magnitude of the high-pass filtered acceleration threshold is exceeded. The TRANSIENT_CFG register is used to enable the transient interrupt generation mechanism for the 3 axes (X, Y, Z) of acceleration. There is also an option to bypass the high-pass filter. When the high-pass filter is bypassed, the function behaves similar to the motion detection.

0x1D: TRANSIENT_CFG Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	ELE	ZTEFE	YTEFE	XTEFE	HPF_BYP

Table 39. TRANSIENT_CFG Description

ELE	Transient event flags are latched into the TRANSIENT_SRC register. Reading of the TRANSIENT_SRC register clears the event flag. Default value: 0. 0: Event flag latch disabled; 1: Event flag latch enabled
ZTEFE	Event flag enable on Z transient acceleration greater than transient threshold event. Default value: 0. 0: Event detection disabled; 1: Raise event flag on measured acceleration delta value greater than transient threshold.
YTEFE	Event flag enable on Y transient acceleration greater than transient threshold event. Default value: 0. 0: Event detection disabled; 1: Raise event flag on measured acceleration delta value greater than transient threshold.
XTEFE	Event flag enable on X transient acceleration greater than transient threshold event. Default value: 0. 0: Event detection disabled; 1: Raise event flag on measured acceleration delta value greater than transient threshold.
HPF_BYP	Bypass High-Pass filter Default value: 0. 0: Data to transient acceleration detection block is through HPF 1: Data to transient acceleration detection block is NOT through HPF (similar to motion detection function)

0x1E: TRANSIENT_SRC Register

The Transient Source register provides the status of the enabled axes and the polarity (directional) information. When this register is read it clears the interrupt for the transient detection. When new events arrive while EA = 1, additional *TRANSE bits may get set, and the corresponding *_Trans_Pol flag become updated. However, no *TRANSE bit may get cleared before the TRANSIENT_SRC register is read.

0x1E: TRANSIENT_SRC Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	EA	ZTRANSE	Z_Trans_Pol	YTRANSE	Y_Trans_Pol	XTRANSE	X_Trans_Pol

Table 40. TRANSIENT_SRC Description

EA	Event Active Flag. Default value: 0. 0: no event flag has been asserted; 1: one or more event flag has been asserted.
ZTRANSE	Z transient event. Default value: 0. 0: no interrupt, 1: Z Transient acceleration greater than the value of TRANSIENT_THS event has occurred
Z_Trans_Pol	Polarity of Z Transient Event that triggered interrupt. Default value: 0. 0: Z event was Positive g, 1: Z event was Negative g
YTRANSE	Y transient event. Default value: 0. 0: no interrupt, 1: Y Transient acceleration greater than the value of TRANSIENT_THS event has occurred
Y_Trans_Pol	Polarity of Y Transient Event that triggered interrupt. Default value: 0. 0: Y event was Positive g, 1: Y event was Negative g
XTRANSE	X transient event. Default value: 0. 0: no interrupt, 1: X Transient acceleration greater than the value of TRANSIENT_THS event has occurred
X_Trans_Pol	Polarity of X Transient Event that triggered interrupt. Default value: 0. 0: X event was Positive g, 1: X event was Negative g

When the EA bit gets set while ELE = 1, all other status bits get frozen at their current state. By reading the TRANSIENT_SRC register, all bits get cleared.

0x1F: TRANSIENT_THS Register

The Transient Threshold register sets the threshold limit for the detection of the transient acceleration. The value in the TRANSIENT_THS register corresponds to a g value which is compared against the values of High-Pass Filtered Data. If the High-Pass Filtered acceleration value exceeds the threshold limit, an event flag is raised and the interrupt is generated if enabled.

0x1F: TRANSIENT_THS Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBCNTM	THS6	THS5	THS4	THS3	THS2	THS1	THS0

Table 41. TRANSIENT_THS Description

DBCNTM	Debounce counter mode selection. Default value: 0. 0: increments or decrements debounce; 1: increments or clears counter.
THS[6:0]	Transient Threshold: Default value: 000_0000.

The threshold THS[6:0] is a 7-bit unsigned number, 0.063g/LSB. The maximum threshold is 8g. Even if the part is set to full scale at 2g or 4g this function will still operate up to 8g. If the Low Noise bit is set in Register 0x2A, the maximum threshold to be reached is 4g.

0x20: TRANSIENT_COUNT

The TRANSIENT_COUNT sets the minimum number of debounce counts continuously matching the condition where the unsigned value of high-pass filtered data is greater than the user specified value of TRANSIENT_THS.

0x20: TRANSIENT_COUNT Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

Table 42. TRANSIENT_COUNT Description

D[7:0]	Count value. Default value: 0000_0000.

The time step for the transient detection debounce counter is set by the value of the system ODR and the Oversampling mode.

Table 43. TRANSIENT_COUNT Relationship with the ODR

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.319	0.319	0.319	0.319	1.25	1.25	1.25	1.25
400	0.638	0.638	0.638	0.638	2.5	2.5	2.5	2.5
200	1.28	1.28	0.638	1.28	5	5	2.5	5
100	2.55	2.55	0.638	2.55	10	10	2.5	10
50	5.1	5.1	0.638	5.1	20	20	2.5	20
12.5	5.1	20.4	0.638	20.4	20	80	2.5	80
6.25	5.1	20.4	0.638	40.8	20	80	2.5	160
1.56	5.1	20.4	0.638	40.8	20	80	2.5	160

6.6 Single, Double and Directional Tap Detection Registers

For more details of how to configure the tap detection and sample code, please refer to Freescale application note, AN4072. The tap detection registers are referred to as “Pulse”.

0x21: PULSE_CFG Pulse Configuration Register

This register configures the event flag for the tap detection for enabling/disabling the detection of a single and double pulse on each of the axes.

0x21: PULSE_CFG Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPA	ELE	ZDPEFE	ZSPEFE	YDPEFE	YSPEFE	XDPEFE	XSPEFE

Table 44. PULSE_CFG Description

DPA	Double Pulse Abort. Default value: 0. 0: Double Pulse detection is not aborted if the start of a pulse is detected during the time period specified by the PULSE_LTCY register. 1: Setting the DPA bit momentarily suspends the double tap detection if the start of a pulse is detected during the time period specified by the PULSE_LTCY register and the pulse ends before the end of the time period specified by the PULSE_LTCY register.
ELE	Pulse event flags are latched into the PULSE_SRC register. Reading of the PULSE_SRC register clears the event flag. Default value: 0. 0: Event flag latch disabled; 1: Event flag latch enabled
ZDPEFE	Event flag enable on double pulse event on Z-axis. Default value: 0. 0: Event detection disabled; 1: Event detection enabled
ZSPEFE	Event flag enable on single pulse event on Z-axis. Default value: 0. 0: Event detection disabled; 1: Event detection enabled
YDPEFE	Event flag enable on double pulse event on Y-axis. Default value: 0. 0: Event detection disabled; 1: Event detection enabled
YSPEFE	Event flag enable on single pulse event on Y-axis. Default value: 0. 0: Event detection disabled; 1: Event detection enabled
XDPEFE	Event flag enable on double pulse event on X-axis. Default value: 0. 0: Event detection disabled; 1: Event detection enabled
XSPEFE	Event flag enable on single pulse event on X-axis. Default value: 0. 0: Event detection disabled; 1: Event detection enabled

0x22: PULSE_SRC Pulse Source Register

This register indicates a double or single pulse event has occurred and also which direction. The corresponding axis and event must be enabled in Register 0x21 for the event to be seen in the source register.

0x22: PULSE_SRC Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	AxZ	AxY	AxX	DPE	PolZ	PolY	PolX

Table 45. PULSE_SRC Description

EA	Event Active Flag. Default value: 0. (0: No interrupt has been generated; 1: One or more interrupt events have been generated)
AxZ	Z-axis event. Default value: 0. (0: No interrupt; 1: Z-axis event has occurred)
AxY	Y-axis event. Default value: 0. (0: No interrupt; 1: Y-axis event has occurred)
AxX	X-axis event. Default value: 0. (0: No interrupt; 1: X-axis event has occurred)
DPE	Double pulse on first event. Default value: 0. (0: Single Pulse Event triggered interrupt; 1: Double Pulse Event triggered interrupt)
PolZ	Pulse polarity of Z-axis Event. Default value: 0. (0: Pulse Event that triggered interrupt was Positive; 1: Pulse Event that triggered interrupt was negative)
PolY	Pulse polarity of Y-axis Event. Default value: 0. (0: Pulse Event that triggered interrupt was Positive; 1: Pulse Event that triggered interrupt was negative)
PolX	Pulse polarity of X-axis Event. Default value: 0. (0: Pulse Event that triggered interrupt was Positive; 1: Pulse Event that triggered interrupt was negative)

When the EA bit gets set while ELE = 1, all status bits (AxZ, AxY, AxZ, DPE, and PolX, PolY, PolZ) are frozen. Reading the PULSE_SRC register clears all bits. Reading the source register will clear the interrupt.

0x23 - 0x25: PULSE_THSX, Y, Z Pulse Threshold for X, Y & Z Registers

The pulse threshold can be set separately for the X, Y and Z axes. The PULSE_THSX, PULSE_THSY and PULSE_THSZ registers define the threshold which is used by the system to start the pulse detection procedure.

0x23: PULSE_THSX Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	THSX6	THSX5	THSX4	THSX3	THSX2	THSX1	THSX0

Table 46. PULSE_THSX Description

THSX[6:0]	Pulse Threshold on X-axis. Default value: 000_0000.

0x24: PULSE_THSY Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	THSY6	THSY5	THSY4	THSY3	THSY2	THSY1	THSY0

Table 47. PULSE_THSY Description

THSY[6:0]	Pulse Threshold on Y-axis. Default value: 000_0000.

0x25: PULSE_THSZ Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	THSZ6	THSZ5	THSZ4	THSZ3	THSZ2	THSZ1	THSZ0

Table 48. PULSE_THSZ Description

THSZ[6:0]	Pulse Threshold on Z-axis. Default value: 000_0000.

The threshold values range from 1 to 127 with steps of 0.63g/LSB at a fixed 8g acceleration range, thus the minimum resolution is always fixed at 0.063g/LSB. If the Low Noise bit in Register 0x2A is set then the maximum threshold will be 4g. The PULSE_THSX, PULSE_THSY and PULSE_THSZ registers define the threshold which is used by the system to start the pulse detection procedure. The threshold value is expressed over 7-bits as an unsigned number.

0x26: PULSE_TMLT Pulse Time Window 1 Register

0x26: PULSE_TMLT Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMLT7	TMLT6	TMLT5	TMLT4	TMLT3	TMLT2	TMLT1	TMLT0

Table 49. PULSE_TMLT Description

TMLT[7:0]	Pulse Time Limit. Default value: 0000_0000.

The bits TMLT7 through TMLT0 define the maximum time interval that can elapse between the start of the acceleration on the selected axis exceeding the specified threshold and the end when the acceleration on the selected axis must go below the specified threshold to be considered a valid pulse.

The minimum time step for the pulse time limit is defined in Table 50 and Table 51. Maximum time for a given ODR and Oversampling mode is the time step pulse multiplied by 255. The time steps available are dependent on the Oversampling mode and whether the Pulse Low-Pass Filter option is enabled or not. The Pulse Low Pass Filter is set in Register 0x0F.

Table 50. Time Step for PULSE Time Limit (Reg 0x0F) Pulse_LPF_EN = 1

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.319	0.319	0.319	0.319	1.25	1.25	1.25	1.25
400	0.638	0.638	0.638	0.638	2.5	2.5	2.5	2.5
200	1.28	1.28	0.638	1.28	5	5	2.5	5
100	2.55	2.55	0.638	2.55	10	10	2.5	10
50	5.1	5.1	0.638	5.1	20	20	2.5	20
12.5	5.1	20.4	0.638	20.4	20	80	2.5	80
6.25	5.1	20.4	0.638	40.8	20	80	2.5	160
1.56	5.1	20.4	0.638	40.8	20	80	2.5	160

Table 51. Time Step for PULSE Time Limit (Reg 0x0F) Pulse_LPF_EN = 0

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.159	0.159	0.159	0.159	0.625	0.625	0.625	0.625
400	0.159	0.159	0.159	0.319	0.625	0.625	0.625	1.25
200	0.319	0.319	0.159	0.638	1.25	1.25	0.625	2.5
100	0.638	0.638	0.159	1.28	2.5	2.5	0.625	5
50	1.28	1.28	0.159	2.55	5	5	0.625	10
12.5	1.28	5.1	0.159	10.2	5	20	0.625	40
6.25	1.28	5.1	0.159	10.2	5	20	0.625	40
1.56	1.28	5.1	0.159	10.2	5	20	0.625	40

0x27: PULSE_LTCY Pulse Latency Timer Register

0x27: PULSE_LTCY Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LTCY7	LTCY6	LTCY5	LTCY4	LTCY3	LTCY2	LTCY1	LTCY0

Table 52. PULSE_LTCY Description

LTCY[7:0]	Latency Time Limit. Default value: 0000_0000

The bits LTCY7 through LTCY0 define the time interval that starts after the first pulse detection. During this time interval, all pulses are ignored. **Note:** This timer must be set for single pulse and for double pulse.

The minimum time step for the pulse latency is defined in Table 53 and Table 54. The maximum time is the time step at the ODR and Oversampling mode multiplied by 255. The timing also changes when the Pulse LPF is enabled or disabled.

Table 53. Time Step for PULSE Latency @ ODR and Power Mode (Reg 0x0F) Pulse_LPF_EN = 1

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.638	0.638	0.638	0.638	2.5	2.5	2.5	2.5
400	1.276	1.276	1.276	1.276	5	5	5	5
200	2.56	2.56	1.276	2.56	10	10	5	10
100	5.1	5.1	1.276	5.1	20	20	5	20
50	10.2	10.2	1.276	10.2	40	40	5	40
12.5	10.2	40.8	1.276	40.8	40	160	5	160
6.25	10.2	40.8	1.276	81.6	40	160	5	320
1.56	10.2	40.8	1.276	81.6	40	160	5	320

Table 54. Time Step for PULSE Latency @ ODR and Power Mode (Reg 0x0F) Pulse_LPF_EN = 0

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.318	0.318	0.318	0.318	1.25	1.25	1.25	1.25
400	0.318	0.318	0.318	0.638	1.25	1.25	1.25	2.5
200	0.638	0.638	0.318	1.276	2.5	2.5	1.25	5
100	1.276	1.276	0.318	2.56	5	5	1.25	10
50	2.56	2.56	0.318	5.1	10	10	1.25	20
12.5	2.56	10.2	0.318	20.4	10	40	1.25	80
6.25	2.56	10.2	0.318	20.4	10	40	1.25	80
1.56	2.56	10.2	0.318	20.4	10	40	1.25	80

0x28: PULSE_WIND Register (Read/Write)

0x28: PULSE_WIND Second Pulse Time Window Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WIND7	WIND6	WIND5	WIND4	WIND3	WIND2	WIND1	WIND0

Table 55. PULSE_WIND Description

WIND[7:0]	Second Pulse Time Window. Default value: 0000_0000.

The bits WIND7 through WIND0 define the maximum interval of time that can elapse after the end of the latency interval in which the start of the second pulse event must be detected provided the device has been configured for double pulse detection. The detected second pulse width must be shorter than the time limit constraints specified by the PULSE_TMLT register, but the end of the double pulse need not finish within the time specified by the PULSE_WIND register.

The minimum time step for the pulse window is defined in Table 56 and Table 57. The maximum time is the time step at the ODR, Oversampling mode and LPF Filter Option multiplied by 255.

Table 56. Time Step for PULSE Detection Window @ ODR and Power Mode (Reg 0x0F) Pulse_LPF_EN = 1

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.638	0.638	0.638	0.638	2.5	2.5	2.5	2.5
400	1.276	1.276	1.276	1.276	5	5	5	5
200	2.56	2.56	1.276	2.56	10	10	5	10
100	5.1	5.1	1.276	5.1	20	20	5	20
50	10.2	10.2	1.276	10.2	40	40	5	40
12.5	10.2	40.8	1.276	40.8	40	160	5	160
6.25	10.2	40.8	1.276	81.6	40	160	5	320
1.56	10.2	40.8	1.276	81.6	40	160	5	320

Table 57. Time Step for PULSE Detection Window @ ODR and Power Mode (Reg 0x0F) Pulse_LPF_EN = 0

ODR (Hz)	Max Time Range (s)				Time Step (ms)			
	Normal	LPLN	HighRes	LP	Normal	LPLN	HighRes	LP
800	0.318	0.318	0.318	0.318	1.25	1.25	1.25	1.25
400	0.318	0.318	0.318	0.638	1.25	1.25	1.25	2.5
200	0.638	0.638	0.318	1.276	2.5	2.5	1.25	5
100	1.276	1.276	0.318	2.56	5	5	1.25	10
50	2.56	2.56	0.318	5.1	10	10	1.25	20
12.5	2.56	10.2	0.318	20.4	10	40	1.25	80
6.25	2.56	10.2	0.318	20.4	10	40	1.25	80
1.56	2.56	10.2	0.318	20.4	10	40	1.25	80

6.7 Auto-WAKE/SLEEP Detection

The ASLP_COUNT register sets the minimum time period of inactivity required to change current ODR value from the value specified in the **DR[2:0]** register to **ASLP_RATE** register value, provided the **SLPE** bit is set to a logic '1' in the **CTRL_REG2** register. See Table 59 for functional blocks that may be monitored for inactivity in order to trigger the "return to SLEEP" event.

0x29: ASLP_COUNT Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

Table 58. ASLP_COUNT Description

D[7:0]	Duration value. Default value: 0000_0000.

D7-D0 defines the minimum duration time to change current ODR value from **DR** to **ASLP_RATE**. Time step and maximum value depend on the ODR chosen as shown in Table 59.

Table 59. ASLP_COUNT Relationship with ODR

Output Data Rate (ODR)	Duration	ODR Time Step	ASLP_COUNT Step
800 Hz	0 to 81s	1.25 ms	320 ms
400 Hz	0 to 81s	2.5 ms	320 ms
200 Hz	0 to 81s	5 ms	320 ms
100 Hz	0 to 81s	10 ms	320 ms
50 Hz	0 to 81s	20 ms	320 ms
12.5 Hz	0 to 81s	80 ms	320 ms
6.25 Hz	0 to 81s	160 ms	320 ms
1.56 Hz	0 to 162s	640 ms	640 ms

Table 60. SLEEP/WAKE Mode Gates and Triggers

Interrupt Source	Event restarts timer and delays Return to SLEEP	Event will WAKE from SLEEP
FIFO_GATE	Yes	No
SRC_TRANS	Yes	Yes
SRC_LNDPRT	Yes	Yes
SRC_PULSE	Yes	Yes
SRC_FF_MT	Yes	Yes
SRC_ASLP	No*	No*
SRC_DRDY	No	No

* If the FIFO_GATE bit is set to logic '1', the assertion of the SRC_ASLP interrupt does not prevent the system from transitioning to SLEEP or from WAKE mode; instead it prevents the FIFO buffer from accepting new sample data until the host application flushes the FIFO buffer.

In order to wake the device, the desired function or functions must be enabled in CTRL_REG4 and set to WAKE to SLEEP in CTRL_REG3. All enabled functions will still function in SLEEP mode at the SLEEP ODR. Only the functions that have been selected for WAKE from SLEEP will **WAKE** the device.

MMA8451Q has four functions that can be used to keep the sensor from falling asleep; Transient, Orientation, Tap and Motion/Freefall. One or more of these functions can be enabled. In order to WAKE the device, four functions are provided; Transient, Orientation, Tap, and the Motion/Freefall. Note that the FIFO does not WAKE the device. The Auto-WAKE/SLEEP interrupt does not affect the WAKE/SLEEP, nor does the data ready interrupt. The FIFO gate (bit 7) in Register 0x2C, when set, will hold the last data in the FIFO before transitioning to a different ODR. After the buffer is flushed, it will accept new sample data at the current ODR. See Register 0x2C for the WAKE from SLEEP bits.

If the Auto-SLEEP bit is disabled, then the device can only toggle between STANDBY and WAKE mode. If Auto-SLEEP interrupt is enabled, transitioning from ACTIVE mode to Auto-SLEEP mode and vice versa generates an interrupt.

6.8 Control Registers

Note: Except for STANDBY mode selection, the device must be in STANDBY mode to change any of the fields within CTRL_REG1 (0x2A).

0x2A: CTRL_REG1 System Control 1 Register

0x2A: CTRL_REG1 Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ASLP_RATE1	ASLP_RATE0	DR2	DR1	DR0	LNOISE	F_READ	ACTIVE

Table 61. CTRL_REG1 Description

ASLP_RATE[1:0]	Configures the Auto-WAKE sample frequency when the device is in SLEEP Mode. Default value: 00. See Table 62 for more information.
DR[2:0]	Data rate selection. Default value: 000. See Table 63 for more information.
LNOISE	Reduced noise reduced Maximum range mode. Default value: 0. (0: Normal mode; 1: Reduced Noise mode)
F_READ	Fast Read mode: Data format limited to single Byte Default value: 0. (0: Normal mode 1: Fast Read Mode)
ACTIVE	Full Scale selection. Default value: 00. (0: STANDBY mode; 1: ACTIVE mode)

Table 62. SLEEP Mode Rate Description

ASLP_RATE1	ASLP_RATE0	Frequency (Hz)
0	0	50
0	1	12.5
1	0	6.25
1	1	1.56

It is important to note that when the device is Auto-SLEEP mode, the system ODR and the data rate for all the system functional blocks are overridden by the data rate set by the **ASLP_RATE** field.

DR[2:0] bits select the Output Data Rate (ODR) for acceleration samples. The default value is **000 for a data rate of 800 Hz.**

Table 63. System Output Data Rate Selection

DR2	DR1	DR0	ODR	Period
0	0	0	800 Hz	1.25 ms
0	0	1	400 Hz	2.5 ms
0	1	0	200 Hz	5 ms
0	1	1	100 Hz	10 ms
1	0	0	50 Hz	20 ms
1	0	1	12.5 Hz	80 ms
1	1	0	6.25 Hz	160 ms
1	1	1	1.56 Hz	640 ms

ACTIVE bit selects between STANDBY mode and ACTIVE mode. The default value is 0 for STANDBY mode.

Table 64. Full Scale Selection

Active	Mode
0	STANDBY
1	ACTIVE

LNOISE bit selects between normal full dynamic range mode and a high sensitivity, Low Noise mode. In Low Noise mode, the maximum signal that can be measured is ±4g. **Note:** Any thresholds set above 4g will not be reached.

F_READ bit selects between normal and Fast Read mode. When selected, the auto increment counter will skip over the LSB data bytes. Data read from the FIFO will skip over the LSB data, reducing the acquisition time. Note F_READ can only be changed when FMODE = 00. The F_READ bit applies for both the output registers and the FIFO.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ST	RST	0	SMODS1	SMODS0	SLPE	MODS1	MODS0

Table 65. CTRL_REG2 Description

ST	Self-Test Enable. Default value: 0. 0: Self-Test disabled; 1: Self-Test enabled
RST	Software Reset. Default value: 0. 0: Device reset disabled; 1: Device reset enabled.
SMODS[1:0]	SLEEP mode power scheme selection. Default value: 00. See Table 66 and Table 67
SLPE	Auto-SLEEP enable. Default value: 0. 0: Auto-SLEEP is not enabled; 1: Auto-SLEEP is enabled.
MODS[1:0]	ACTIVE mode power scheme selection. Default value: 00. See Table 66 and Table 67

ST bit activates the self-test function. When ST is set, X, Y, and Z outputs will shift. **RST** bit is used to activate the software reset. The reset mechanism can be enabled in STANDBY and ACTIVE mode.

When the reset bit is enabled, all registers are rest and are loaded with default values. Writing '1' to the RST bit immediately resets the device, no matter whether it is in ACTIVE/WAKE, ACTIVE/SLEEP, or STANDBY mode.

The I^2C communication system is reset to avoid accidental corrupted data access.

At the end of the boot process, the RST bit is deasserted to 0. Reading this bit will return a value of zero.

The **(S)MODS[1:0]** bits select which Oversampling mode is to be used shown in Table 66. The Oversampling modes are available in both WAKE Mode MOD[1:0] and also in the SLEEP Mode SMOD[1:0].

Table 66. MODS Oversampling Modes

(S)MODS1	(S)MODS0	Power Mode
0	0	Normal
0	1	Low Noise Low Power
1	0	High Resolution
1	1	Low Power

Table 67. MODS Oversampling Modes Current Consumption and Averaging Values at each ODR

Mode	Normal (00)		Low Noise Low Power (01)		High Resolution (10)		Low Power (11)	
ODR	Current µA	OS Ratio	Current µA	OS Ratio	Current µA	OS Ratio	Current µA	OS Ratio
1.56 Hz	24	128	8	32	165	1024	6	16
6.25 Hz	24	32	8	8	165	256	6	4
12.5 Hz	24	16	8	4	165	128	6	2
50 Hz	24	4	24	4	165	32	14	2
100 Hz	44	4	44	4	165	16	24	2
200 Hz	85	4	85	4	165	8	44	2
400 Hz	165	4	165	4	165	4	85	2
800 Hz	165	2	165	2	165	2	165	2

0x2C: CTRL_REG3 Interrupt Control Register

0x2C: CTRL_REG3 Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIFO_GATE	WAKE_TRANS	WAKE_LNDPRT	WAKE_PULSE	WAKE_FF_MT	—	IPOL	PP_OD

Table 68. CTRL_REG3 Description

FIFO_GATE	0: FIFO gate is bypassed. FIFO is flushed upon the system mode transitioning from WAKE to SLEEP mode or from SLEEP to WAKE mode. Default value: 0. 1: The FIFO input buffer is blocked when transitioning from WAKE to SLEEP mode or from SLEEP to WAKE mode until the FIFO is flushed. Although the system transitions from WAKE to SLEEP or from SLEEP to WAKE the contents of the FIFO buffer are preserved, new data samples are ignored until the FIFO is emptied by the host application. If the FIFO_GATE bit is set to logic '1' and the FIFO buffer is not emptied before the arrival of the next sample, then the FGERR bit in the SYS_MOD register (0x0B) will be asserted. The FGERR bit remains asserted as long as the FIFO buffer remains un-emptied. Emptying the FIFO buffer clears the FGERR bit in the SYS_MOD register.
WAKE_TRANS	0: Transient function is bypassed in SLEEP mode. Default value: 0. 1: Transient function interrupt can wake up system
WAKE_LNDPRT	0: Orientation function is bypassed in SLEEP mode. Default value: 0. 1: Orientation function interrupt can wake up system
WAKE_PULSE	0: Pulse function is bypassed in SLEEP mode. Default value: 0. 1: Pulse function interrupt can wake up system
WAKE_FF_MT	0: Freefall/Motion function is bypassed in SLEEP mode. Default value: 0. 1: Freefall/Motion function interrupt can wake up
IPOL	Interrupt polarity ACTIVE high, or ACTIVE low. Default value: 0. 0: ACTIVE low; 1: ACTIVE high
PP_OD	Push-Pull/Open Drain selection on interrupt pad. Default value: 0. 0: Push-Pull; 1: Open Drain

IPOL bit selects the polarity of the interrupt signal. When IPOL is '0' (default value) any interrupt event will signaled with a logical 0.
PP_OD bit configures the interrupt pin to Push-Pull or in Open Drain mode. The default value is 0 which corresponds to Push-Pull mode. The Open Drain configuration can be used for connecting multiple interrupt signals on the same interrupt line.

0x2D: CTRL_REG4 Register (Read/Write)

0x2D: CTRL_REG4 Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT_EN_ASLP	INT_EN_FIFO	INT_EN_TRANS	INT_EN_LNDPR	INT_EN_PULSE	INT_EN_FF_MT	—	INT_EN_DRDY

Table 69. Interrupt Enable Register Description

Interrupt Enable	Description
INT_EN_ASLP	Interrupt Enable. Default value: 0. 0: Auto-SLEEP/WAKE interrupt disabled; 1: Auto-SLEEP/WAKE interrupt enabled.
INT_EN_FIFO	Interrupt Enable. Default value: 0. 0: FIFO interrupt disabled; 1: FIFO interrupt enabled.
INT_EN_TRANS	Interrupt Enable. Default value: 0. 0: Transient interrupt disabled; 1: Transient interrupt enabled.
INT_EN_LNDPRT	Interrupt Enable. Default value: 0. 0: Orientation (Landscape/Portrait) interrupt disabled. 1: Orientation (Landscape/Portrait) interrupt enabled.
INT_EN_PULSE	Interrupt Enable. Default value: 0. 0: Pulse Detection interrupt disabled; 1: Pulse Detection interrupt enabled
INT_EN_FF_MT	Interrupt Enable. Default value: 0. 0: Freefall/Motion interrupt disabled; 1: Freefall/Motion interrupt enabled
INT_EN_DRDY	Interrupt Enable. Default value: 0. 0: Data Ready interrupt disabled; 1: Data Ready interrupt enabled

The corresponding functional block interrupt enable bit allows the functional block to route its event detection flags to the system's interrupt controller. The interrupt controller routes the enabled functional block interrupt to the INT1 or INT2 pin.

0x2E: CTRL_REG5 Register (Read/Write)

0x2E: CTRL_REG5 Interrupt Configuration Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT_CFG_ASLP	INT_CFG_FIFO	INT_CFG_TRANS	INT_CFG_LNDPRT	INT_CFG_PULSE	INT_CFG_FF_MT	—	INT_CFG_DRDY

Table 70. Interrupt Configuration Register Description

Interrupt Configuration	Description
INT_CFG_ASLP	INT1/INT2 Configuration. Default value: 0. 0: Interrupt is routed to INT2 pin; 1: Interrupt is routed to INT1 pin
INT_CFG_FIFO	INT1/INT2 Configuration. Default value: 0 0: Interrupt is routed to INT2 pin; 1: Interrupt is routed to INT1 pin
INT_CFG_TRANS	INT1/INT2 Configuration. Default value: 0. 0: Interrupt is routed to INT2 pin; 1: Interrupt is routed to INT1 pin
INT_CFG_LNDPRT	INT1/INT2 Configuration. Default value: 0. 0: Interrupt is routed to INT2 pin; 1: Interrupt is routed to INT1 pin
INT_CFG_PULSE	INT1/INT2 Configuration. Default value: 0. 0: Interrupt is routed to INT2 pin; 1: Interrupt is routed to INT1 pin
INT_CFG_FF_MT	INT1/INT2 Configuration. Default value: 0. 0: Interrupt is routed to INT2 pin; 1: Interrupt is routed to INT1 pin
INT_CFG_DRDY	INT1/INT2 Configuration. Default value: 0. 0: Interrupt is routed to INT2 pin; 1: Interrupt is routed to INT1 pin

The system's interrupt controller shown in Figure 10 uses the corresponding bit field in the CTRL_REG5 register to determine the routing table for the INT1 and INT2 interrupt pins. If the bit value is logic '0', the functional block's interrupt is routed to INT2, and if the bit value is logic '1', then the interrupt is routed to INT1. One or more functions can assert an interrupt pin; therefore a host application responding to an interrupt should read the INT_SOURCE (0x0C) register to determine the appropriate sources of the interrupt.

6.9 User Offset Correction Registers

For more information on how to calibrate the 0g offset, refer to application note AN4069. The 2's complement offset correction registers values are used to realign the Zero-g position of the X, Y, and Z-axis after device board mount. The resolution of the offset registers is 2 mg per LSB. The 2's complement 8-bit value would result in an offset compensation range ±256 mg.

0x2F: OFF_X Offset Correction X Register

0x2F: OFF_X Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

Table 71. OFF_X Description

D[7:0]	X-axis offset value. Default value: 0000_0000.

0x30: OFF_Y Offset Correction Y Register

0x30: OFF_Y Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

Table 72. OFF_Y Description

D[7:0]	Y-axis offset value. Default value: 0000_0000.

0x31: OFF_Z Offset Correction Z Register

0x31: OFF_Z Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

Table 73. OFF_Z Description

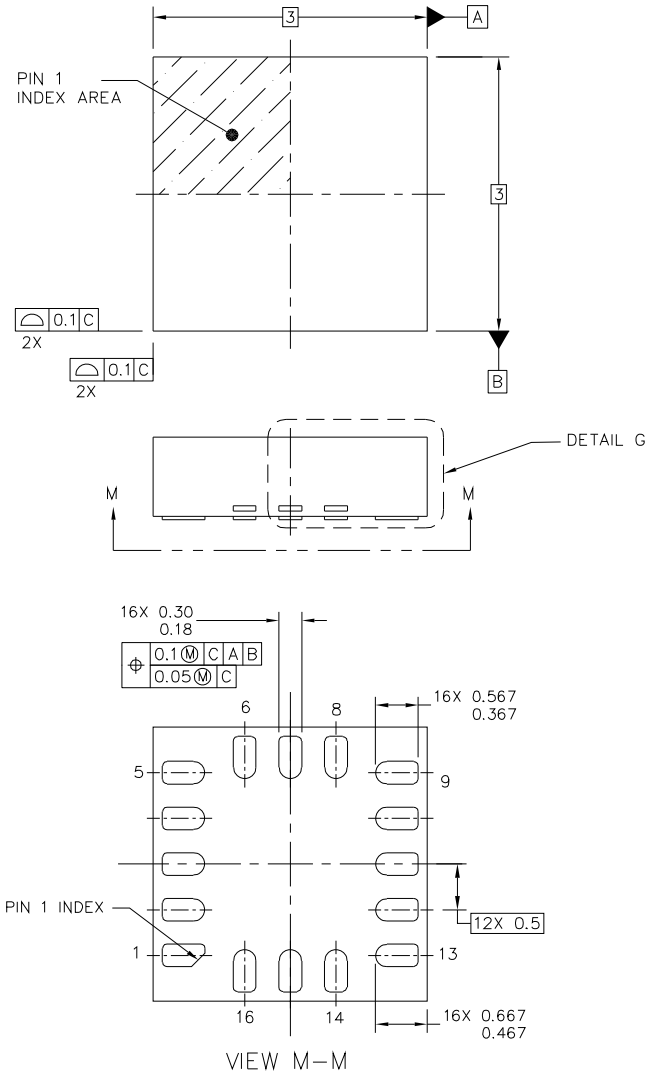
D[7:0]	Z-axis offset value. Default value: 0000_0000.

Table 74. MMA8451Q Register Map

Reg	Name	Definition	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00	STATUS/F_STATUS	Data Status R	ZYXOW	ZOW	YOW	XOW	ZYXDR	ZDR	YDR	XDR
01	OUT_X_MSB	14 bit X Data R	XD13	XD12	XD11	XD10	XD9	XD8	XD7	XD6
02	OUT_X_LSB	14 bit X Data R	XD5	XD4	XD3	XD2	XD1	XD0	0	0
03	OUT_Y_MSB	14 bit Y Data R	YD13	YD12	YD11	YD10	YD9	YD8	YD7	YD6
04	OUT_Y_LSB	14 bit Y Data R	YD5	YD4	YD3	YD2	YD1	YD0	0	0
05	OUT_Z_MSB	14 bit Z Data R	ZD13	ZD12	ZD11	ZD10	ZD9	ZD8	ZD7	ZD6
06	OUT_Z_LSB	14 bit Z Data R	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0	0	0
09	F_SETUP	FIFO Setup R/W	F_MODE1	F_MODE0	F_WMRK5	F_WMRK4	F_WMRK3	F_WMRK2	F_WMRK1	F_WMRK0
0A	TRIG_CFG	FIFO Triggers R/W	—	—	Trig_TRANS	Trig_LNDPRT	Trig_PULSE	Trig_FF_MT	—	—
0B	SYSMOD	System Mode R	FGERR	FGT_4	FGT_3	FGT_2	FGT_1	FGT_0	SYSMOD1	SYSMOD0
0C	INT_SOURCE	Interrupt Status R	SRC_ASLP	SRC_FIFO	SRC_TRANS	SRC_LNDPRT	SRC_PULSE	SRC_FF_MT	—	SRC_DRDY
0D	WHO_AM_I	ID Register R	0	0	0	1	1	0	1	0
0E	XYZ_DATA_CFG	Data Config R/W	—	—	—	HPF_Out	—	—	FS1	FS0
0F	HP_FILTER_CUTOFF	HP Filter Setting R/W	—	—	Pulse_HPF_BYP	Pulse_LPF_EN	—	—	SEL1	SEL0
10	PL_STATUS	PL Status R	NEWLP	LO	—	—	—	LAPO[1]	LAPO[0]	BAFRO
11	PL_CFG	PL Configuration R/W	DBCNTM	PL_EN	—	—	—	—	—	—
12	PL_COUNT	PL DEBOUNCE R/W	DBNCE[7]	DBNCE[6]	DBNCE[5]	DBNCE[4]	DBNCE[3]	DBNCE[2]	DBNCE[1]	DBNCE[0]
13	PL_BF_ZCOMP	PL Back/Front Z Comp R/W	BKFR[1]	BKFR[0]	—	—	—	ZLOCK[2]	ZLOCK[1]	ZLOCK[0]
14	P_L_THS_REG	PL THRESHOLD R/W	P_L_THS[4]	P_L_THS[3]	P_L_THS[2]	P_L_THS[1]	P_L_THS[0]	HYS[2]	HYS[1]	HYS[0]
15	FF_MT_CFG	Freefall/Motion Config R/W	ELE	OAE	ZEFE	YEFE	XEFE	—	—	—
16	FF_MT_SRC	Freefall/Motion Source R	EA	—	ZHE	ZHP	YHE	YHP	XHE	XHP
17	FF_MT_THS	Freefall/Motion Threshold R/W	DBCNTM	THS6	THS5	THS4	THS3	THS2	THS1	THS0
18	FF_MT_COUNT	Freefall/Motion Debounce R/W	D7	D6	D5	D4	D3	D2	D1	D0
1D	TRANSIENT_CFG	Transient Config R/W	—	—	—	ELE	ZTEFE	YTEFE	XTEFE	HPF_BYP
1E	TRANSIENT_SRC	Transient Source R	—	EA	ZTRANSE	Z_Trans_Pol	YTRANSE	Y_Trans_Pol	XTRANSE	X_Trans_Pol
1F	TRANSIENT_THS	Transient Threshold R/W	DBCNTM	THS6	THS5	THS4	THS3	THS2	THS1	THS0
20	TRANSIENT_COUNT	Transient Debounce R/W	D7	D6	D5	D4	D3	D2	D1	D0
21	PULSE_CFG	Pulse Config R/W	DPA	ELE	ZDPEFE	ZSPEFE	YDPEFE	YSPEFE	XDPEFE	XSPEFE
22	PULSE_SRC	Pulse Source R	EA	AxZ	AxY	AxX	DPE	Pol_Z	Pol_Y	Pol_X
23	PULSE_THSX	Pulse X Threshold R/W	—	THSX6	THSX5	THSX4	THSX3	THSX2	THSX1	THSX0
24	PULSE_THSY	Pulse Y Threshold R/W	—	THSY6	THSY5	THSY4	THSY3	THSY2	THSY1	THSY0
25	PULSE_THSZ	Pulse Z Threshold R/W	—	THSZ6	THSZ5	THSZ4	THSZ3	THSZ2	THSZ1	THSZ0
26	PULSE_TMLT	Pulse First Timer R/W	TMLT7	TMLT6	TMLT5	TMLT4	TMLT3	TMLT2	TMLT1	TMLT0
27	PULSE_LTCY	Pulse Latency R/W	LTCY7	LTCY6	LTCY5	LTCY4	LTCY3	LTCY2	LTCY1	LTCY0
28	PULSE_WIND	Pulse 2nd Window R/W	WIND7	WIND6	WIND5	WIND4	WIND3	WIND2	WIND1	WIND0
29	ASLP_COUNT	Auto-SLEEP Counter R/W	D7	D6	D5	D4	D3	D2	D1	D0
2A	CTRL_REG1	Control Reg1 R/W	ASLP_RATE1	ASLP_RATE0	DR2	DR1	DR0	LNOISE	F_READ	ACTIVE
2B	CTRL_REG2	Control Reg2 R/W	ST	RST	—	SMODS1	SMODS0	SLPE	MODS1	MODS0
2C	CTRL_REG3	Control Reg3 (WAKE Interrupts from SLEEP) R/W	FIFO_GATE	WAKE_TRANS	WAKE_LNDPRT	WAKE_PULSE	WAKE_FF_MT	—	IPOL	PP_OD
2D	CTRL_REG4	Control Reg4 (Interrupt Enable Map) R/W	INT_EN_ASLP	INT_EN_FIFO	INT_EN_TRANS	INT_EN_LNDPRT	INT_EN_PULSE	INT_EN_FF_MT	—	INT_EN_DRDY
2E	CTRL_REG5	Control Reg5 (Interrupt Configuration) R/W	INT_CFG_ASLP	INT_CFG_FIFO	INT_CFG_TRANS	INT_CFG_LNDPRT	INT_CFG_PULSE	INT_CFG_FF_MT	—	INT_CFG_DRDY
2F	OFF_X	X 8-bit offset R/W	D7	D6	D5	D4	D3	D2	D1	D0
30	OFF_Y	Y 8-bit offset R/W	D7	D6	D5	D4	D3	D2	D1	D0
31	OFF_Z	Z 8-bit offset R/W	D7	D6	D5	D4	D3	D2	D1	D0

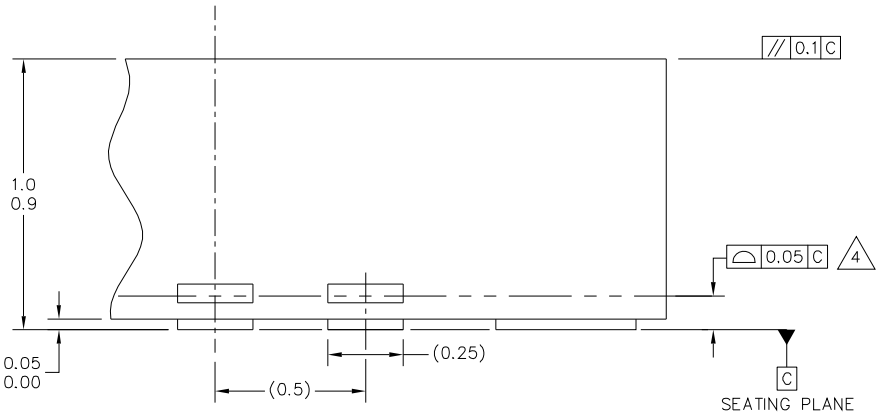
Table 75. Accelerometer Output Data

14-bit Data	Range ±2g (0.25 mg)	Range ±4g (0.5 mg)	Range ±8g (1.0 mg)
01 1111 1111 1111	1.99975g	+3.9995g	+7.999g
01 1111 1111 1110	1.99950g	+3.9990g	+7.998g
…	…	…	…
00 0000 0000 0001	0.00025g	+0.0005g	+0.001g
00 0000 0000 0000	0.00000g	0.00000g	0.000g
11 1111 1111 1111	-0.00025g	-0.0005g	-0.001g
…	…	…	…
10 0000 0000 0001	-1.99975g	-3.9995g	-7.999g
10 0000 0000 0000	-2.00000g	-4.0000g	-8.000g
8-bit Data	**Range ±2g (15.6 mg)**	**Range ±4g (31.25 mg)**	**Range ±8g (62.5 mg)**
0111 1111	1.9844g	+3.9688g	+7.9375g
0111 1110	1.9688g	+3.9375g	+7.8750g
…	…	…	…
0000 0001	+0.0156g	+0.0313g	+0.0625g
0000 0000	0.000g	0.0000g	0.0000g
1111 1111	-0.0156g	-0.0313g	-0.0625g
…	…	…	…
1000 0001	-1.9844g	-3.9688g	-7.9375g
1000 0000	-2.0000g	-4.0000g	-8.0000g



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QUAD FLAT NO LEAD COL PACKAGE (QFN–COL) 16 TERMINAL, 0.5 PITCH (3 X 3 X 1.0)	DOCUMENT NO: 98ASA00063D	REV: A
	CASE NUMBER: 2077–02	20 OCT 2011
	STANDARD: NON JEDEC	

CASE 2077-02
ISSUE A
16-LEAD QFN



DETAIL G

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QUAD FLAT NO LEAD COL PACKAGE (QFN–COL) 16 TERMINAL, 0.5 PITCH (3 X 3 X 1.0)	DOCUMENT NO: 98ASA00063D	REV: A
	CASE NUMBER: 2077–02	20 OCT 2011
	STANDARD: NON JEDEC	

CASE 2077-02
ISSUE A
16-LEAD QFN

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M–1994.
3. THIS IS NON JEDEC REGISTERED PACKAGE.
4. COPLANARITY APPLIES TO ALL LEADS.
5. MIN. METAL GAP SHOULD BE 0.2MM.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QUAD FLAT NO LEAD COL PACKAGE (QFN–COL) 16 TERMINAL, 0.5 PITCH (3 X 3 X 1.0)	DOCUMENT NO: 98ASA00063D	REV: A
	CASE NUMBER: 2077–02	20 OCT 2011
	STANDARD: NON JEDEC	

CASE 2077-02
ISSUE A
16-LEAD QFN

Table 76. Revision History

Revision number	Revision date	Description of changes
7	03/2012	• Table 2. Updated Typ values for Sensitivity Accuracy from 2.5% to 2.64%; Zero-g Level Offset Accuracy from ±20 mg to ±17 mg and Zero-g Level Offset Accuracy Post Board Mount from ±30 mg to ±20 mg. • Table 4. Updated Min value from 50 µs to 0.05 µs and added Max value of 0.9 µs • Added Table 8. Features of the MMA845xQ devices. • Removed FIFO paragraph at the end of Section 6.1. • Updated Note preceding Table 32 from "THS + HYS > 0 and THS + HYS < 49... " to "The condition THS + HYS > 0 and THS + HYS < 32 must be met in order for Landscape/Portrait detection to work properly...." • Updated Case outline with current version.
7.1	05/2012	• Updated Figure 5 to correspond to table.
8	02/2013	• Replaced Section 2.3 I^2C interface characteristics, including Table 4 and Figure 5.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: MMA8451Q
Rev. 8
02/2013

Energy Efficient Solutions by Freescale

freescale™

How Many Bits are Enough?

The Trade-off Between High Resolution and Low Power Using Oversampling Modes

by: Kimberly Tuck
Applications Engineer

1.0 Introduction

This application note is targeted to try to help answer the question “How many bits are enough” as well as several others that fall out of this same question. The intent is to educate customers on the details to understand more about their application, to help extract meaningful information from the data sheet and learn some of the techniques to get the most out of the sensor for the application while realizing the limitations of the sensor.

Often the customer will be looking for the resolution limitations of the accelerometer to determine if the sensor is capable of the application. Resolution is defined as the smallest detectable change in acceleration. This information can be extracted out of the number of bits of the accelerometer. This is part of the information required but it is not the only piece of information. The system noise performance must also be taken into account. Without a device to do noise measurements and experimentally test the resolution how can one calculate the expected values and what are some of the important parameters to consider that will affect the results?

If an accelerometer is advertised as being 12-bit, what does that mean? The first thing one should ask is what range is the 12 bits divided into. Twelve bits over an 8g range is not the same as 12 bits over a 2g range. Over an 8g range, 12 bits will give you 256 counts/g and over a 2g range, 12 bits will give you 1024 counts per g, which is 4 times more sensitivity! It is important to first understand what the *sensitivity* is of the device by looking at the total dynamic range that the accelerometer is measuring.

TABLE OF CONTENTS

1.1 Key Words

Accelerometer, Static Acceleration, Tilt Angles, Portrait/Landscape Orientation, Embedded Algorithm MMA8450Q, Z-Angle Lockout, XYZ Output Data, Low Current Consumption, Motion and Tap Detection, Design Flexibility, Hysteresis, 3-axis Accelerometer, Offset Considerations, Sample Rate, Debounce

1.2 Summary

A. The effective number of bit calculations provides better insight into the resolution of the digitized signal.

B. Noise affects the resolution of the device. Calculations for the Root Mean Square (RMS) noise and Power Spectral Density (PSD) are valuable to understand the effective number of bits.

C. Oversampling can be used to improve the resolution of the device. The MMA8451, 2, 3Q provides 4 different oversampling schemes which trade-off between resolution and current consumption at varying sample rates.

2.0 MMA8451, 2, 3Q Consumer 3-axis Accelerometer 3 x 3 x 1 mm

The MMA8451, 2, 3Q has a selectable dynamic range of ±2g, ±4g, ±8g. The device has 8 different output data rates, selectable high pass filter cut-off frequencies, and high pass filtered data. The available resolution of the data and the embedded features is dependant on the specific device.

Note: The MMA8450Q has a different memory map and has a slightly different pin-out configuration.

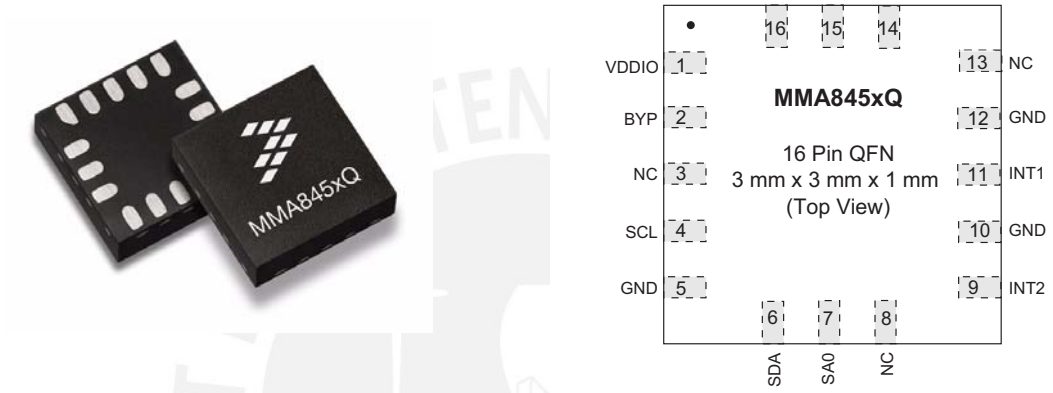


Figure 1. MMA8451, 2, 3Q Consumer 3-axis Accelerometer 3 x 3 x 1 mm

2.1 Output Data, Sample Rates and Dynamic Ranges of all Three Products

2.1.1 MMA8451Q

1. **14-bit data**
   2g (4096 counts/g = 0.25 mg/LSB) **4g** (2048 counts/g = 0.5 mg/LSB) **8g** (1024 counts/g = 1 mg/LSB)
2. **8-bit data**
   2g (64 counts/g = 15.6 mg/LSB) **4g** (32 counts/g = 31.25 mg/LSB) **8g** (16 counts/g = 62.5 mg/LSB)
3. **Embedded 32 sample FIFO (MMA8451Q)**

2.1.2 MMA8452Q

1. **12-bit data**
   2g (1024 counts/g = 1 mg/LSB) **4g** (512 counts/g = 2 mg/LSB) **8g** (256 counts/g = 3.9 mg/LSB)
2. **8-bit data**
   2g (64 counts/g = 15.6 mg/LSB) **4g** (32 counts/g = 31.25 mg/LSB) **8g** (16 counts/g = 62.5 mg/LSB)

2.1.3 MMA8453Q Note: No HPF Data

1. **10-bit data**
   2g (256 counts/g = 3.9 mg/LSB) **4g** (128 counts/g = 7.8 mg/LSB) **8g** (64 counts/g = 15.6 mg/LSB)
2. **8-bit data**
   2g (64 counts/g = 15.6 mg/LSB) **4g** (32 counts/g = 31.25 mg/LSB) **8g** (16 counts/g = 62.5 mg/LSB)

2.2 Application Notes for the MMA8451, 2, 3Q

The following is a list of all the application notes available for the MMA8451, 2, 3Q:

- **AN4068**, *Embedded Orientation Detection Using the MMA8451, 2, 3Q*
- **AN4069**, *Offset Calibration of the MMA8451, 2, 3Q*
- **AN4070**, *Motion and Freefall Detection Using the MMA8451, 2, 3Q*
- **AN4071**, *High Pass Data and Functions Using the MMA8451, 2,3Q*
- **AN4072**, *MMA8451, 2, 3Q Single/Double and Directional Tap Detection*
- **AN4073**, *Using the 32 Sample First In First Out (FIFO) in the MMA8451Q*
- **AN4074**, *Auto-Wake/Sleep Using the MMA8451, 2, 3Q*
- **AN4075, *How Many Bits are Enough? The Trade-off Between High Resolution and Low Power Using Oversampling Modes***
- **AN4076**, *Data Manipulation and Basic Settings of the MMA8451, 2, 3Q*
- **AN4077**, *MMA8451, 2, 3Q Design Checklist and Board Mounting Guidelines*

3.0 Number of Bits verses Effective Bits

The number of bits is normally specified as the number of bits of the digitizer. Any data acquisition system or ADC has inherent performance limitations. The data sheet may advertise that a part has a particular number of bits but the number of *effective* bits provides the insight into the resolution of the part. Every accelerometer will have a certain amount of system noise that will limit the number of effective bits. By understanding how to calculate and measure the noise performance of the part and by doing a few calculations the effective number of bits can be calculated to provide the resolution limitations of the device.

4.0 Noise

There are two types of noise in the accelerometer. There is electronic noise in the ASIC and there is mechanical noise from the MEMS g-cell. The electronic noise has been minimized as much as possible in the ASIC while the mechanical noise is due to thermo-mechanical noise of the moving parts. The noise in the MEMS g-cell is minimized by good design practices as well. The mechanical noise is a function of the resonant frequency, the mass m the damping Q and the temperature T as shown in the following equation:

$$N_{Mech} = \frac{\sqrt{\frac{4k_bT\omega}{mQ}}}{g}$$

The overall system noise is measured as a function of frequency. Often the Power Spectral Density (PSD) is given in the data sheet representing the total internal system noise. Spectral density captures the frequency content of a stochastic process (noise). This describes how the power is distributed with frequency. The noise in the accelerometer is predominantly considered Gaussian white noise and therefore is a constant value across all frequencies. The PSD given in the data sheet is given in units of μg per square root Hz. The PSD relates to the RMS noise by the following equations.

$$N^2_{RMS} = \int_0^{\infty} PSD(f)df$$

$$N^2_{RMS} = \int_0^{BW} PSD(f)df$$

$$N^2_{RMS} = PSD(BW - 0)$$

$$N_{RMS} = \sqrt{PSD \times BW}$$

From the above equations you can see that the PSD which is often given in the data sheet is actually the square root PSD. To calculate the RMS noise simply multiply the bandwidth by the PSD value. Note this is the RMS noise of the system but does not include the quantization noise from digitizing the signal.

4.1 Quantization Error

Quantization error occurs because the analog signal is sampled and must be divided into a finite number. The output of the ADC has 2^n divisions or counts, where n is the number of converter bits. A 12-bit ADC has 2^{12} or 4096 counts. The bit that represents the smallest change is the least significant bit (LSB). Each sample has a quantization error of ±0.5 of the LSB due to the difference between the true analog output and the count represented by the ADC output. The quantization error is the minimum noise level to resolve the signal in an ideal perfect system.

The “mean square” of the quantization error is found by integrating over the quantization voltage error range as follows:

$$QE_{mean^2} = \int_{-V_q/2}^{+V_q/2} \frac{E^2}{V_q} dE = \frac{V_q^2}{12}$$

The RMS value of the quantization error is found by taking the square root of the mean square error where *Vq* is the quantization voltage and is equal to 1 bit.

$$QE_{RMS} = \sqrt{\frac{V_q^2}{12}} = \frac{V_q}{\sqrt{12}} = \frac{1LSB}{\sqrt{12}} = 0.288 \cdot mg/LSB$$

The quantization noise is often so small that it is negligible in the calculation, but it is part of the overall noise that is measured.

4.2 Measuring Noise from the Accelerometer

The noise of the accelerometer can be *measured* by the following steps:

1. Place the accelerometer and ensure that it remains motionless away from any kind of vibration.
2. Datalog the raw data output from the XYZ axes of the accelerometer
3. Calculate the standard deviation from the data.
4. Calculate the RMS noise based on the known sensitivity.
5. Calculate the pk-pk noise to quantify the resolution

The **standard deviation** is from the following formula:

$$\sum_{i=0}^{n} \frac{1}{n} \sqrt{(x_i - \mu)^2} \qquad \text{where } \mu \text{ is the mean}$$

If the sensitivity of the part is 1024 counts/g then this equates to 0.976 mg/count. As an example if the standard deviation is 4.1 counts then from this value we can calculate the RMS Noise from the following equation:

N_{RMS} = STD* 1/Sensitivity

N_{RMS} = 4.1counts* 0.976 mg/count = 4.0016 mg

Note: The measured noise includes all noise of the system. This will include the analog noise and the quantization noise as well as any mechanical noise.

Knowing the N_{RMS} the PSD can be estimated if the bandwidth is known.

$$PSD = \frac{N_{RMS}}{\sqrt{BW}}$$

In this example if the bandwidth is 200 Hz (400 Hz sample rate) the PSD estimate would be $283\mu g/\sqrt{Hz}$.

If we multiply the N_{RMS} by 4 we get an estimate of the pk-pk noise which provides the resolution. Therefore the resolution estimate in this example would be 4.0016 mg*4, which is 16 mg.

To estimate the resolution from the data sheet, use the PSD value from the data sheet.

$$N_{RMS} = PSD \times \sqrt{BW}$$

The PSD value in the data sheet as mentioned is actually the square root PSD from the derivation above. This will give you the RMS noise of the system but does not include the quantization noise.

For example, if the accelerometer has a $PSD = 85\mu g/\sqrt{Hz}$ and the bandwidth is 200 Hz (Sample Rate = 400 Hz) then the RMS noise would be 1.2 mg and the pk-pk estimate would be 1.2*4 = 4.8 mg.

It follows that assuming the PSD is constant and the sensitivity is constant that the standard deviation measured should decrease as the bandwidth decreases and ultimately the RMS noise decreases.

5.0 Calculating the Effective Number of Bits

Data sheets on the website will advertise 16-bit resolution, but often times this is not true. The device may have 16 bits but the real question is how many bits are noise free? Calculating the effective number of bits the Signal to Noise Ratio (SNR) is equated. The RMS value of the signal is divided by the quantization noise.

$$\frac{S_{RMS}}{N_{RMS}} = \frac{\frac{2^n}{2\sqrt{2}}}{\frac{1}{\sqrt{12}}} = 2^n \cdot \frac{\sqrt{3}}{\sqrt{2}}$$

Then the signal to noise ratio is converted to the decibel system.

$$\left(\frac{S_{RMS}}{N_{RMS}}\right)db = 20\log\left(2^n \cdot \frac{\sqrt{3}}{\sqrt{2}}\right)$$

$$\left(\frac{S_{RMS}}{N_{RMS}}\right)db = 20\log 2^n + 20\log 1.225$$

$$\left(\frac{S_{RMS}}{N_{RMS}}\right)db = 602n + 1.76$$

$$n = \frac{SNR(db) - 1.76}{6.02}$$

This results in the equation for finding the effective number of bits “n”.

5.1 Example Finding Effective Number of Bits

If the accelerometer has a dynamic range of 2g, then the full signal is ±2g which is a total of 4g. The next equation relates how to calculate the SNR. The noise can be calculated based on measured data from the device from the steps provided in Section 4.2, or an estimate can be made if the PSD value and bandwidth is known.

$$SNR(db) = 20\log\left(\frac{\frac{4g}{2\sqrt{2}}}{N_{RMS}}\right)$$

$$n = \frac{SNR(db) - 1.76}{6.02}$$

If the result is that the device has 8.5 effective bits then by taking the full scale and dividing by 2^BEff, results in the true resolution (noise free).

- 4g/ 2^8.5 = 11.05 mg resolution

Now that there is knowledge on how to calculate resolution the question becomes more an understanding of the needs for the application.

This is something that the user needs to determine about their application. A lot of times this comes about from experimentation and creative use-case analysis. Freescale offers the Sensor Toolbox demo board system to allow customers to try any of our accelerometers to datalog and experiment with their application. It allows for data collection to analyze the performance and this application note is intended to be used to determine the true resolution of the devices.

6.0 Improving the Resolution using Oversampling and Decimation

The oversampling factor k is equal to 2^{2n}, where n is the number of bits to gain by oversampling. The signal to noise ratio increases as follows by adding on the 10logk term.

$$SNR[dB] = 6.02N + 1.76 + 10\log k$$

The following table shows the values for k with the SNR improvement and the added extra effective number of bits. This isn't an endless process. There is a trade-off with reaction time. As more and more samples are average the reaction time slows down by a factor proportional to "k".

Table 1. Oversampling Factor for Added Resolution

Oversampling Factor (k)	SNR Improvement (dB)	Extra Resolution Bits (n)
2	3	0.5
4	6	1.0
8	9	1.5
16	12	2.0
32	15	2.5
64	18	3.0

There is a trade-off between current consumption and resolution. The MMA8451, 2, 3 accelerometers have been designed to fit as many applications and markets as possible. The MMA845xQ series of accelerometers have been designed such that the same accelerometer can be used for high resolution applications or for low power applications simply by switching the device into a different mode. There is also an option for trading off the full dynamic range for improved noise performance.

The internal sampling rate is 1600 Hz. The oversampling ratio "OS ratio" represents the number of averaged samples. Note in High Resolution mode as many samples as possible are averaged. In Low Power mode only 2 samples are averaged. The device is power cycled at 400 Hz and below to minimize current consumption in Low Power mode. There are four different modes that the device can be in. These are Normal Mode, High Resolution Mode, Low Noise and Low Power Mode and Low Power Mode. The difference between the modes is seen in Tables 3, 4, 5 and 6 by the number of averaged samples at each ODR. The oversampling mode can be set for the Wake Mode as well as for the Sleep Mode. The Oversampling Mode register is found in CTRL_REG2. The MMA845xQ also has a low noise bit that can be set in CTRL_REG1. The low noise bit will improve the noise performance and this is separate from averaging samples. This bit will increase the sensitivity but the dynamic range will be limited from 8g to approximately 4g for all functions.

Table 2. MODS Oversampling Mode

(S)MODS1	(S)MODS0	Power Mode
0	0	Normal
0	1	Low Noise and Low Power
1	0	High Resolution
1	1	Low Power

The typical noise calculated in mg RMS is given for all different available sample rates when the Low Noise bit is set and also when it is cleared. This is done at all sample rates and at all oversampling modes. The equivalent effective number of noise free bits of the 14-bit data is given for each based on tested results. For an application requiring the highest resolution with the lowest current consumption, the low noise bit should be set and a trade-off will need to be made depending on acceptable requirements.

Note: The data in Tables 3, 4, 5 and 6 were taken in a typical applications lab environment without any sophisticated noise isolation during measurements.

Table 3. Normal Mode: MODS = 00 Oversampling Effective Bits and Noise Calculations All ODRs MMA8451Q

ODR	Noise RMS (Low Noise Bit = 1)	BEff (2g) (Low Noise Bit = 1)	BEff (4g) (Low Noise Bit = 1)	Noise RMS (Low Noise Bit = 0)	BEff (2g) (Low Noise Bit = 0)	BEff (4g) (Low Noise Bit = 0)	BEff (8g) (Low Noise Bit = 0)	OS Ratio	Current µA
1.56	0.233 mg	12.2	13.2	0.307 mg	11.8	12.8	13.8	128	24
6.25	0.466 mg	11.2	12.2	0.613 mg	10.8	11.8	12.8	32	24
12.5	0.659 mg	10.7	11.7	0.867 mg	10.3	11.3	12.3	16	24
50	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	24
100	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	44
200	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	85
400	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	165
800	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	165

Table 4. High Resolution Mode: MODS = 10 Oversampling Effective Bits and Noise Calculations All ODRs MMA8451Q

ODR	Noise RMS (Low Noise Bit = 1)	BEff (2g) (Low Noise Bit = 1)	BEff (4g) (Low Noise Bit = 1)	Noise RMS (Low Noise Bit = 0)	BEff (2g) (Low Noise Bit = 0)	BEff (4g) (Low Noise Bit = 0)	BEff (8g) (Low Noise Bit = 0)	OS Ratio	Current µA
1.56	0.0824 mg	13.7	14.7	0.108 mg	13.3	14.3	15.3	1024	165
6.25	0.164 mg	12.7	13.7	0.217 mg	12.3	13.3	14.3	256	165
12.5	0.233 mg	12.2	13.2	0.307 mg	11.8	12.8	13.8	128	165
50	0.466 mg	11.2	12.2	0.614 mg	10.8	11.8	12.8	32	165
100	0.659 mg	10.7	11.7	0.868 mg	10.3	11.3	12.3	16	165
200	0.932 mg	10.2	11.2	1.23 mg	9.8	10.8	11.8	8	165
400	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	165
800	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	165

Table 5. Low Noise and Low Power: MODS=01 Oversampling Effective Bits and Noise Calculations All ODRs MMA8451Q

ODR	Noise RMS (Low Noise Bit = 1)	BEff (2g) (Low Noise Bit = 1)	BEff (4g) (Low Noise Bit = 1)	Noise RMS (Low Noise Bit = 0)	BEff (2g) (Low Noise Bit = 0)	BEff (4g) (Low Noise Bit = 0)	BEff (8g) (Low Noise Bit = 0)	OS Ratio	Current µA
1.56	0.466 mg	11.2	12.2	0.614 mg	10.8	11.8	12.8	32	8
6.25	0.932 mg	10.2	11.2	1.23 mg	9.8	10.8	11.8	8	8
12.5	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	8
50	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	24
100	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	44
200	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	85
400	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.3	4	165
800	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	165

Table 6. Low Power: MODS = 11 Oversampling Effective Bits and Noise Calculations All ODRs MMA8451Q

ODR	Noise RMS (Low Noise Bit = 1)	BEff (2g) (Low Noise Bit = 1)	BEff (4g) (Low Noise Bit = 1)	Noise RMS (Low Noise Bit = 0)	BEff (2g) (Low Noise Bit = 0)	BEff (4g) (Low Noise Bit = 0)	BEff (8g) (Low Noise Bit = 0)	OS Ratio	Current µA
1.56	0.659 mg	10.7	11.7	0.868 mg	10.3	11.3	12.3	16	6
6.25	1.32 mg	9.7	10.7	1.74 mg	9.3	10.3	11.2	4	6
12.5	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	6
50	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	14
100	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	24
200	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	44
400	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	85
800	1.86 mg	9.2	10.2	2.45 mg	8.8	9.8	10.8	2	165

The trade-offs between low power and high resolution can be made by choosing the appropriate oversampling mode. These embedded options save the applications processor from averaging data to get improved performance and adds additional flexibility into the performance of the sensor. Example code for setting different oversampling modes is given in Freescale application note AN4074, Auto-Wake/Sleep Using the MMA8451, 2, 3Q.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN4075
Rev. 1
09/2010

MCP73831/2

Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers

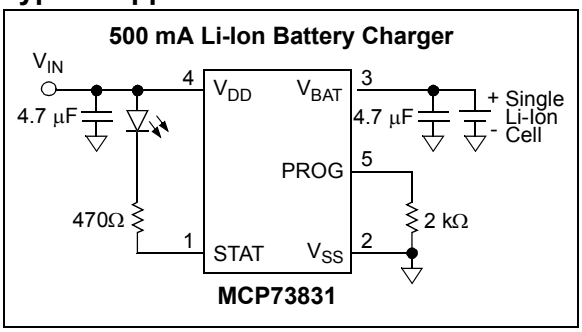
Features:

- Linear Charge Management Controller:
  - Integrated Pass Transistor
  - Integrated Current Sense
  - Reverse Discharge Protection
- High Accuracy Preset Voltage Regulation: ± 0.75%
- Four Voltage Regulation Options:
  - 4.20V, 4.35V, 4.40V, 4.50V
- Programmable Charge Current: 15 mA to 500 mA
- Selectable Preconditioning:
  - 10%, 20%, 40%, or Disable
- Selectable End-of-Charge Control:
  - 5%, 7.5%, 10%, or 20%
- Charge Status Output
  - Tri-State Output - MCP73831
  - Open-Drain Output - MCP73832
- Automatic Power-Down
- Thermal Regulation
- Temperature Range: -40°C to +85°C
- Packaging:
  - 8-Lead, 2 mm x 3 mm DFN
  - 5-Lead, SOT-23

Applications:

- Lithium-Ion/Lithium-Polymer Battery Chargers
- Personal Data Assistants
- Cellular Telephones
- Digital Cameras
- MP3 Players
- Bluetooth Headsets
- USB Chargers

Typical Application



Description:

The MCP73831/2 devices are highly advanced linear charge management controllers for use in space-limited, cost-sensitive applications. The MCP73831/2 are available in an 8-Lead, 2 mm x 3 mm DFN package or a 5-Lead, SOT-23 package. Along with their small physical size, the low number of external components required make the MCP73831/2 ideally suited for portable applications. For applications charging from a USB port, the MCP73831/2 adhere to all the specifications governing the USB power bus.

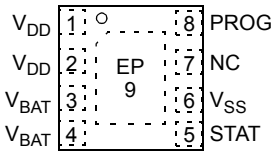
The MCP73831/2 employ a constant-current/constant-voltage charge algorithm with selectable preconditioning and charge termination. The constant voltage regulation is fixed with four available options: 4.20V, 4.35V, 4.40V or 4.50V, to accommodate new, emerging battery charging requirements. The constant current value is set with one external resistor. The MCP73831/2 devices limit the charge current based on die temperature during high power or high ambient conditions. This thermal regulation optimizes the charge cycle time while maintaining device reliability.

Several options are available for the preconditioning threshold, preconditioning current value, charge termination value and automatic recharge threshold. The preconditioning value and charge termination value are set as a ratio or percentage of the programmed constant current value. Preconditioning can be disabled. Refer to **Section 1.0 "Electrical Characteristics"** for available options and the **Product Identification System** for standard options.

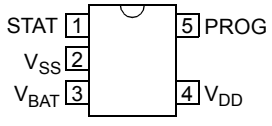
The MCP73831/2 devices are fully specified over the ambient temperature range of -40°C to +85°C.

Package Types

MCP73831/2
2×3 DFN*

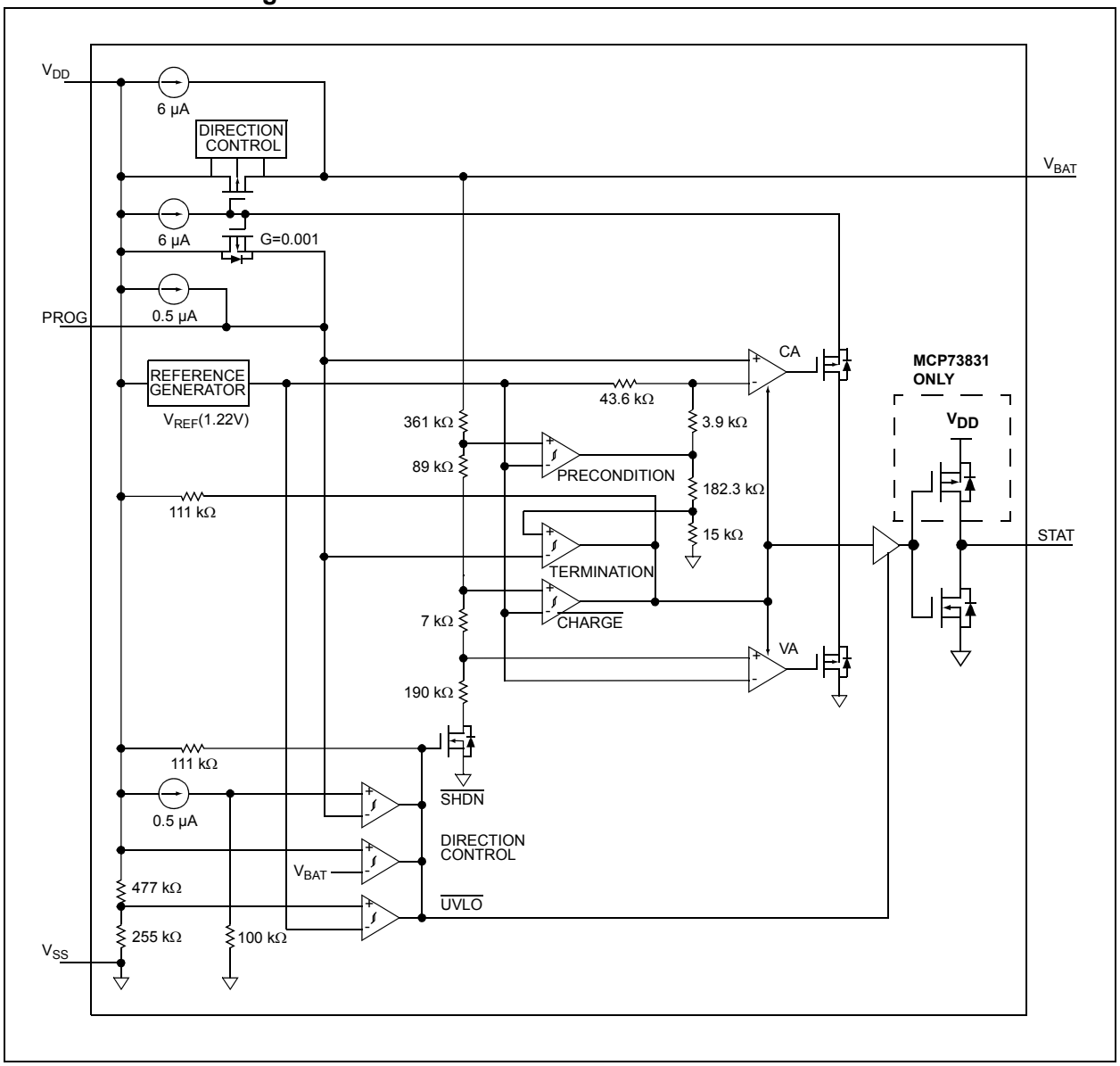


MCP73831/2
SOT-23-5



* Includes Exposed Thermal Pad (EP); see Table 3-1.

Functional Block Diagram



1.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings†

V_{DD}.....7.0V
All Inputs and Outputs w.r.t. V_{SS} ..... -0.3 to (V_{DD}+0.3)V
Maximum Junction Temperature, T_J .....Internally Limited
Storage temperature .....-65°C to +150°C
ESD protection on all pins:
Human Body Model (1.5 kΩ in Series with 100 pF).....≥ 4 kV
Machine Model (200 pF, No Series Resistance).....400V

† Notice: Stresses above those listed under "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 operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

Electrical Specifications: Unless otherwise indicated, all limits apply for V_{DD}= [V_{REG}(typical) + 0.3V] to 6V, T_A = -40°C to +85°C. Typical values are at +25°C, V_{DD} = [V_{REG} (typical) + 1.0V]

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Supply Input						
Supply Voltage	V_{DD}	3.75	—	6	V	
Supply Current	I_{SS}	—	510	1500	µA	Charging
		—	53	200	µA	Charge Complete, No Battery
		—	25	50	µA	PROG Floating
		—	1	5	µA	$V_{DD} \leq (V_{BAT}$ - 50 mV)
		—	0.1	2	µA	$V_{DD} < V_{STOP}$
UVLO Start Threshold	V_{START}	3.3	3.45	3.6	V	V_{DD} Low-to-High
UVLO Stop Threshold	V_{STOP}	3.2	3.38	3.5	V	V_{DD} High-to-Low
UVLO Hysteresis	V_{HYS}	—	70	—	mV	
Voltage Regulation (Constant-Voltage Mode)						
Regulated Output Voltage	V_{REG}	4.168	4.20	4.232	V	MCP7383X-2
		4.317	4.35	4.383	V	MCP7383X-3
		4.367	4.40	4.433	V	MCP7383X-4
		4.466	4.50	4.534	V	MCP7383X-5
						V_{DD} = [V_{REG}(typical)+1V] I_{OUT} = 10 mA T_A = -5°C to +55°C
Line Regulation	$\lvert(\Delta V_{BAT}/V_{BAT})/\Delta V_{DD}\rvert$	—	0.09	0.30	%/V	V_{DD} = [V_{REG}(typical)+1V] to 6V, I_{OUT} = 10 mA
Load Regulation	$\lvert\Delta V_{BAT}/V_{BAT}\rvert$	—	0.05	0.30	%	I_{OUT} = 10 mA to 50 mA V_{DD} = [V_{REG}(typical)+1V]
Supply Ripple Attenuation	PSRR	—	52	—-	dB	I_{OUT}=10 mA, 10Hz to 1 kHz
		—	47	—	dB	I_{OUT}=10 mA, 10Hz to 10 kHz
		—	22	—	dB	I_{OUT}=10 mA, 10Hz to 1 MHz
Current Regulation (Fast Charge Constant-Current Mode)						
Fast Charge Current Regulation	I_{REG}	90	100	110	mA	PROG = 10 kΩ
		450	505	550	mA	PROG = 2.0 kΩ, **Note 1**
		12.5	14.5	16.5	mA	PROG = 67 kΩ
						T_A = -5°C to +55°C

Note 1: Not production tested. Ensured by design.

DC CHARACTERISTICS (CONTINUED)

Electrical Specifications: Unless otherwise indicated, all limits apply for V_{DD}= [V_{REG}(typical) + 0.3V] to 6V, T_A = -40°C to +85°C. Typical values are at +25°C, V_{DD} = [V_{REG} (typical) + 1.0V]

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Preconditioning Current Regulation (Trickle Charge Constant-Current Mode)						
Precondition Current Ratio	I_{PREG} / I_{REG}	7.5	10	12.5	%	PROG = 2.0 kΩ to 10 kΩ
		15	20	25	%	PROG = 2.0 kΩ to 10 kΩ
		30	40	50	%	PROG = 2.0 kΩ to 10 kΩ
		—	100	—	%	No Preconditioning
						T_A = -5°C to +55°C
Precondition Voltage Threshold Ratio	V_{PTH} / V_{REG}	64	66.5	69	%	V_{BAT} Low-to-High
		69	71.5	74	%	V_{BAT} Low-to-High
Precondition Hysteresis	V_{PHYS}	—	110	—	mV	V_{BAT} High-to-Low
Charge Termination						
Charge Termination Current Ratio	I_{TERM} / I_{REG}	3.75	5	6.25	%	PROG = 2.0 kΩ to 10 kΩ
		5.6	7.5	9.4	%	PROG = 2.0 kΩ to 10 kΩ
		8.5	10	11.5	%	PROG = 2.0 kΩ to 10 kΩ
		15	20	25	%	PROG = 2.0 kΩ to 10 kΩ
						T_A = -5°C to +55°C
Automatic Recharge						
Recharge Voltage Threshold Ratio	V_{RTH} / V_{REG}	91.5	94.0	96.5	%	V_{BAT} High-to-Low
		94	96.5	99	%	V_{BAT} High-to-Low
Pass Transistor ON-Resistance						
ON-Resistance	R_{DSON}	—	350	—	mΩ	V_{DD} = 3.75V, T_J = 105°C
Battery Detection						
Battery Detection Current	I_{BAT_DET}	—	6	—	µA	V_{BAT} Source Current
No-Battery-Present Threshold	V_{NO_BAT}	—	V_{REG} + 100 mV	—	V	V_{BAT} Voltage ≥ V_{NO_BAT} for No Battery condition
No-Battery-Present Impedance	Z_{NO_BAT}	2	—	—	MΩ	V_{BAT} Impedance ≥ Z_{NO_BAT} for No Battery condition, **Note 1**
Battery Discharge Current						
Output Reverse Leakage Current	$I_{DISCHARGE}$	—	0.15	2	µA	PROG Floating
		—	0.25	2	µA	V_{DD} Floating
		—	0.15	2	µA	V_{DD} < V_{STOP}
		—	-5.5	-15	µA	Charge Complete
Status Indicator – STAT						
Sink Current	I_{SINK}	—	—	25	mA	
Low Output Voltage	V_{OL}	—	0.4	1	V	I_{SINK} = 4 mA
Source Current	I_{SOURCE}	—	—	35	mA	
High Output Voltage	V_{OH}	—	V_{DD}-0.4	V_{DD} - 1	V	I_{SOURCE} = 4 mA (MCP73831)
Input Leakage Current	I_{LK}	—	0.03	1	µA	High-Impedance
PROG Input						
Charge Impedance Range	R_{PROG}	2	—	67	kΩ	
Minimum Shutdown Impedance	R_{PROG}	70	—	200	kΩ	
Automatic Power Down						
Automatic Power Down Entry Threshold	$V_{PDENTER}$	V_{DD}<(V_{BAT} +20 mV)	V_{DD}<(V_{BAT} +50 mV)	—		3.5V ≤ V_{BAT} ≤ V_{REG} V_{DD} Falling

Note 1: Not production tested. Ensured by design.

DC CHARACTERISTICS (CONTINUED)

Electrical Specifications: Unless otherwise indicated, all limits apply for V_{DD}= [V_{REG}(typical) + 0.3V] to 6V, T_A = -40°C to +85°C. Typical values are at +25°C, V_{DD} = [V_{REG} (typical) + 1.0V]

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Automatic Power Down Exit Threshold	V_{PDEXIT}	—	V_{DD}<(V_{BAT} +150 mV)	V_{DD}<(V_{BAT} +200 mV)		3.5V ≤ V_{BAT} ≤ V_{REG} V_{DD} Rising
Thermal Shutdown						
Die Temperature	T_{SD}	—	150	—	°C	
Die Temperature Hysteresis	T_{SDHYS}	—	10	—	°C	

Note 1: Not production tested. Ensured by design.

AC CHARACTERISTICS

Electrical Specifications: Unless otherwise indicated, all limits apply for V_{DD} = [V_{REG} (typical) + 0.3V] to 12V, T_A = -40°C to +85°C. Typical values are at +25°C, V_{DD} = [V_{REG} (typical) + 1.0V]

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
UVLO Start Delay	t_{START}	—	—	5	ms	V_{DD} Low-to-High
Constant-Current Regulation						
Transition Time Out of Preconditioning	t_{DELAY}	—	—	1	ms	V_{BAT} < V_{PTH} to V_{BAT} > V_{PTH}
Current Rise Time Out of Preconditioning	t_{RISE}	—	—	1	ms	I_{OUT} Rising to 90% of I_{REG}
Termination Comparator Filter	t_{TERM}	0.4	1.3	3.2	ms	Average I_{OUT} Falling
$\overline{\text{Charge}}$ Comparator Filter	t_{CHARGE}	0.4	1.3	3.2	ms	Average V_{BAT}
Status Indicator						
Status Output turn-off	t_{OFF}	—	—	200	µs	I_{SINK} = 1 mA to 0 mA
Status Output turn-on	t_{ON}	—	—	200	µs	I_{SINK} = 0 mA to 1 mA

TEMPERATURE SPECIFICATIONS

Electrical Specifications: Unless otherwise indicated, all limits apply for V_{DD} = [V_{REG} (typical) + 0.3V] to 12V. Typical values are at +25°C, V_{DD} = [V_{REG} (typical) + 1.0V]

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Temperature Ranges						
Specified Temperature Range	T_A	-40	—	+85	°C	
Operating Temperature Range	T_J	-40	—	+125	°C	
Storage Temperature Range	T_A	-65	—	+150	°C	
Thermal Package Resistances						
5-Lead, SOT-23	θ_{JA}	—	230	—	°C/W	4-Layer JC51-7 Standard Board, Natural Convection (**Note 2**)
8-Lead, 2 mm x 3 mm, DFN	θ_{JA}	—	76	—	°C/W	4-Layer JC51-7 Standard Board, Natural Convection (**Note 1**)

Note 1: This represents the minimum copper condition on the PCB (Printed Circuit Board).

2: With large copper area on the PCB, the SOT-23-5 thermal resistance (θ_{JA}) can reach a typical value of 130°C/W or better.

NOTES:

2.0 TYPICAL PERFORMANCE CURVES

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore outside the warranted range.

Note: Unless otherwise indicated, V_{DD} = [V_{REG}(typical) + 1V], I_{OUT} = 10 mA and T_A= +25°C, Constant-Voltage mode.

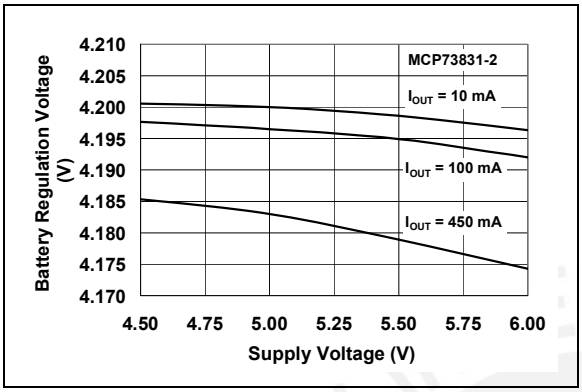


FIGURE 2-1: *Battery Regulation Voltage (V_{BAT}) vs. Supply Voltage (V_{DD}).*

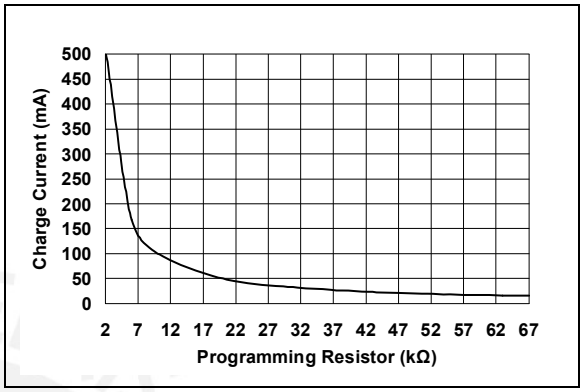


FIGURE 2-4: *Charge Current (I_{OUT}) vs. Programming Resistor (R_{PROG}).*

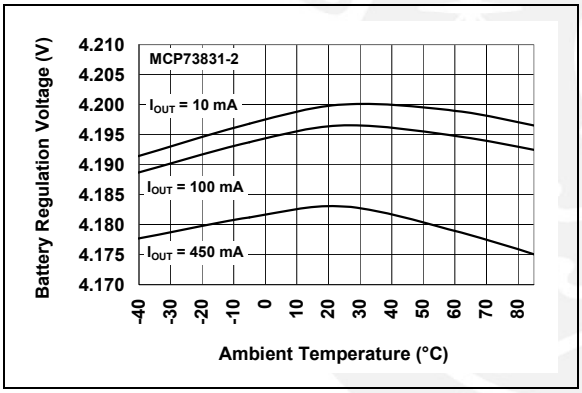


FIGURE 2-2: *Battery Regulation Voltage (V_{BAT}) vs. Ambient Temperature (T_A).*

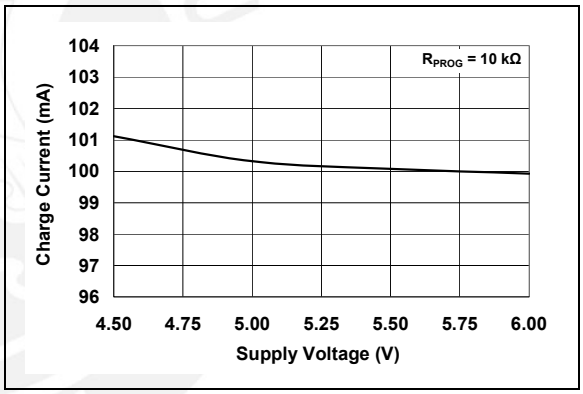


FIGURE 2-5: *Charge Current (I_{OUT}) vs. Supply Voltage (V_{DD}).*

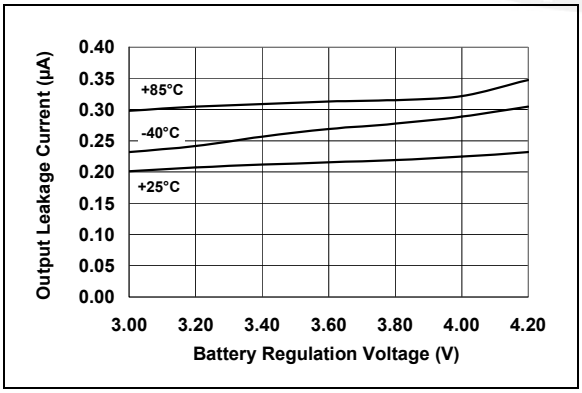


FIGURE 2-3: *Output Leakage Current ($I_{DISCHARGE}$) vs. Battery Regulation Voltage (V_{BAT}).*

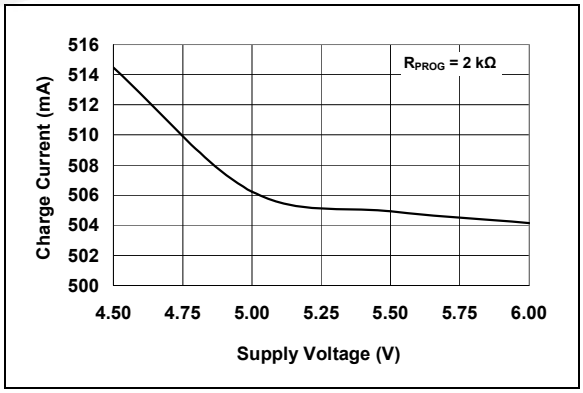


FIGURE 2-6: *Charge Current (I_{OUT}) vs. Supply Voltage (V_{DD}).*

TYPICAL PERFORMANCE CURVES (CONTINUED)

Note: Unless otherwise indicated, V_{DD} = [V_{REG}(typical) + 1V], I_{OUT} = 10 mA and T_A= +25°C, Constant-Voltage mode.

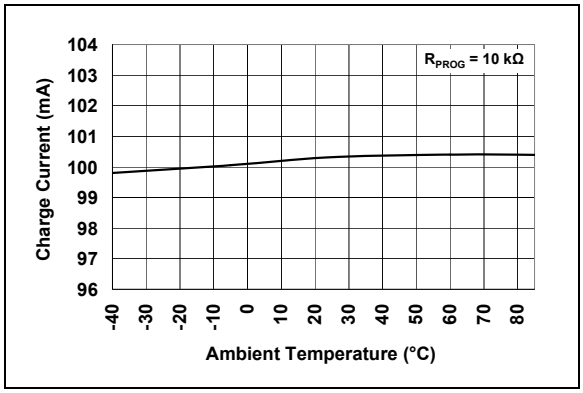


FIGURE 2-7: *Charge Current (I_{OUT}) vs. Ambient Temperature (T_A).*

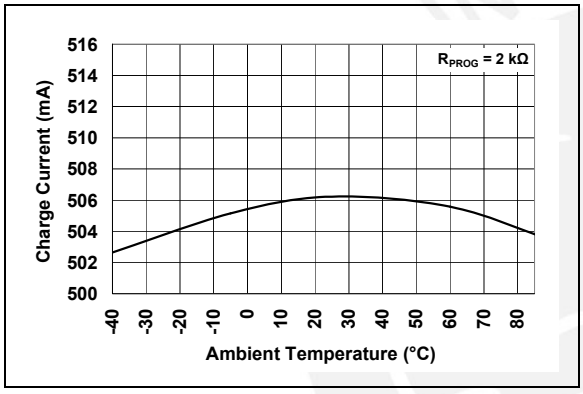


FIGURE 2-8: *Charge Current (I_{OUT}) vs. Ambient Temperature (T_A).*

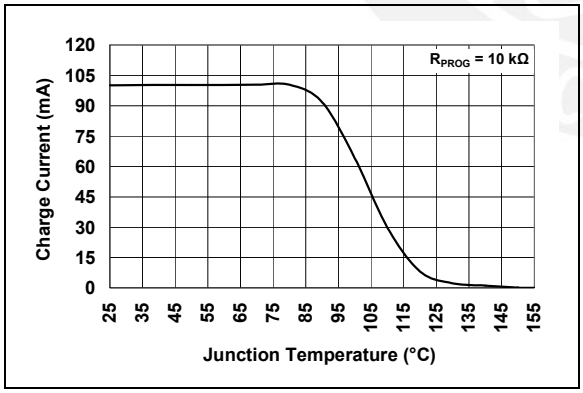


FIGURE 2-9: *Charge Current (I_{OUT}) vs. Junction Temperature (T_J).*

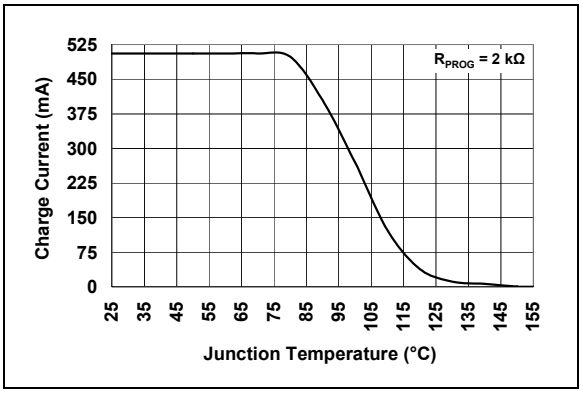


FIGURE 2-10: *Charge Current (I_{OUT}) vs. Junction Temperature (T_J).*

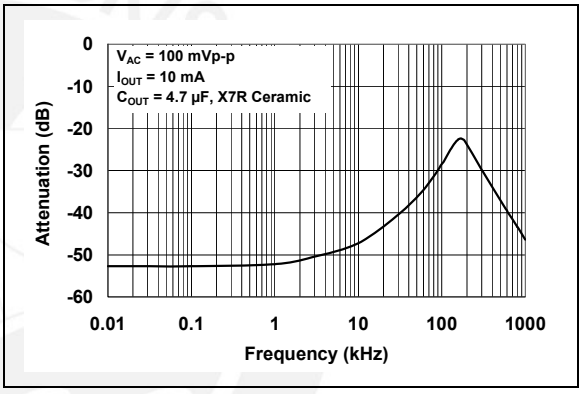


FIGURE 2-11: *Power Supply Ripple Rejection (PSRR).*

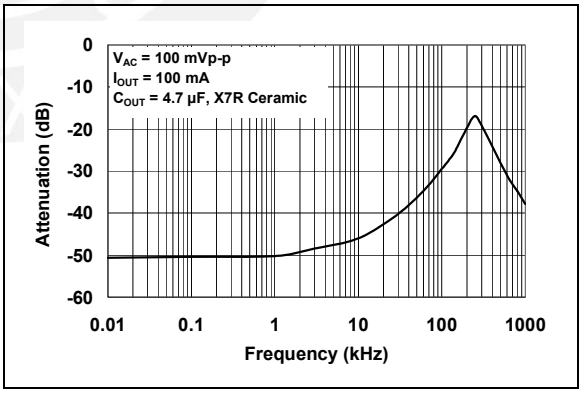


FIGURE 2-12: *Power Supply Ripple Rejection (PSRR).*

TYPICAL PERFORMANCE CURVES (CONTINUED)

Note: Unless otherwise indicated, V_{DD} = [V_{REG}(typical) + 1V], I_{OUT} = 10 mA and T_A= +25°C, Constant-Voltage mode.

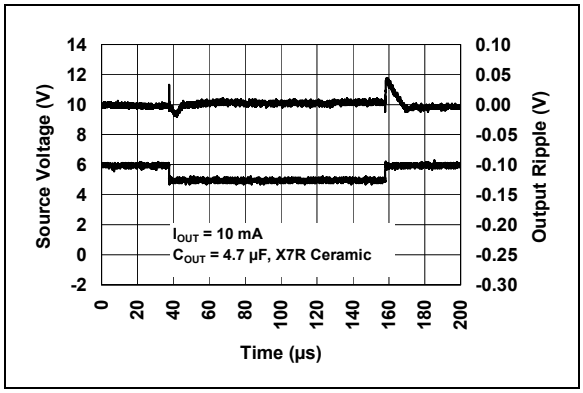


FIGURE 2-13: *Line Transient Response.*

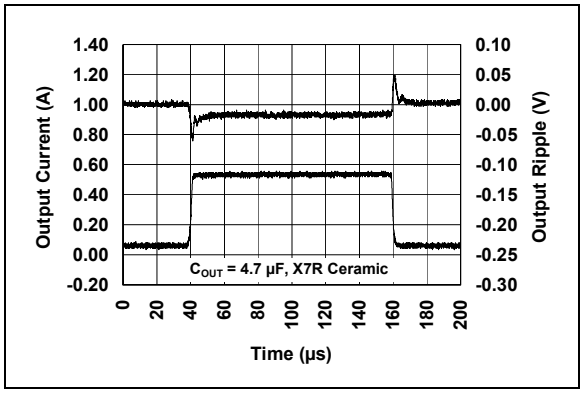


FIGURE 2-16: *Load Transient Response.*

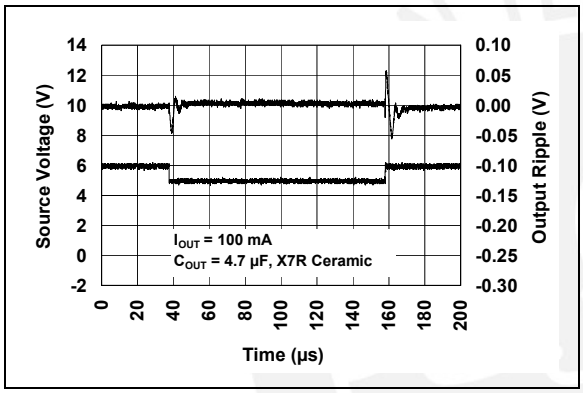


FIGURE 2-14: *Line Transient Response.*

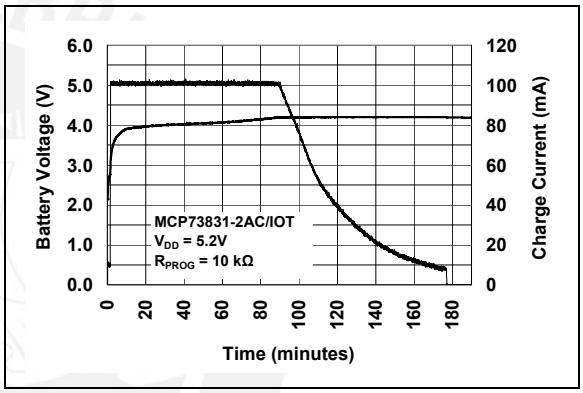


FIGURE 2-17: *Complete Charge Cycle (180 mAh Li-Ion Battery).*

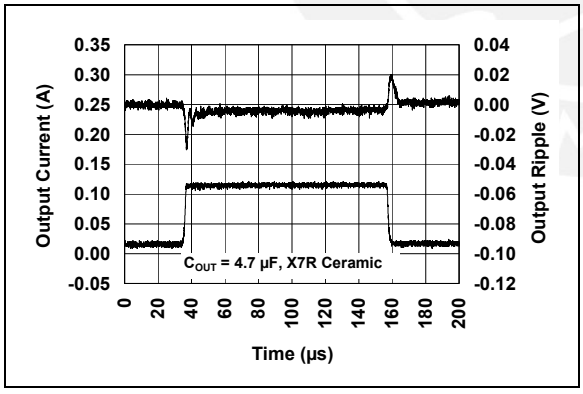


FIGURE 2-15: *Load Transient Response.*

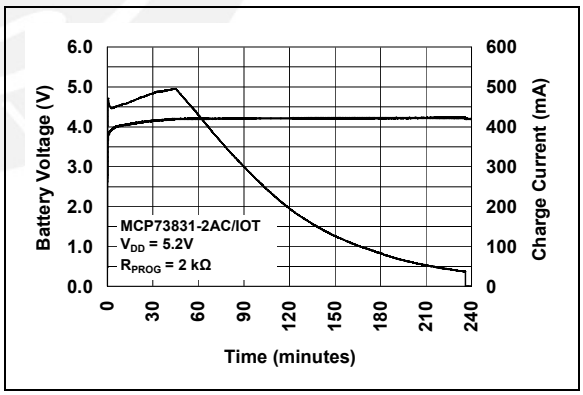


FIGURE 2-18: *Complete Charge Cycle (1000 mAh Li-Ion Battery).*

NOTES:

3.0 PIN DESCRIPTION

The descriptions of the pins are listed in Table 3-1.

TABLE 3-1: PIN FUNCTION TABLES

Pin No.		Symbol	Function
DFN	SOT-23-5		
1	4	V_{DD}	Battery Management Input Supply
2	—	V_{DD}	Battery Management Input Supply
3	3	V_{BAT}	Battery Charge Control Output
4	—	V_{BAT}	Battery Charge Control Output
5	1	STAT	Charge Status Output
6	2	V_{SS}	Battery Management 0V Reference
7	—	NC	No Connection
8	5	PROG	Current Regulation Set and Charge Control Enable
9	—	EP	Exposed Thermal Pad (EP); must be connected to V_{SS}.

3.1 Battery Management Input Supply (V_{DD})

A supply voltage of [V_{REG} (typical) + 0.3V] to 6V is recommended. Bypass to V_{SS} with a minimum of 4.7 µF.

3.2 Battery Charge Control Output (V_{BAT})

Connect to positive terminal of battery. Drain terminal of internal P-channel MOSFET pass transistor. Bypass to V_{SS} with a minimum of 4.7 µF to ensure loop stability when the battery is disconnected.

3.3 Charge Status Output (STAT)

STAT is an output for connection to an LED for charge status indication. Alternatively, a pull-up resistor can be applied for interfacing to a host microcontroller.

STAT is a tri-state logic output on the MCP73831 and an open-drain output on the MCP73832.

3.4 Battery Management 0V Reference (V_{SS})

Connect to negative terminal of battery and input supply.

3.5 Current Regulation Set (PROG)

Preconditioning, fast charge and termination currents are scaled by placing a resistor from PROG to V_{SS}.

The charge management controller can be disabled by allowing the PROG input to float.

3.6 Exposed Thermal Pad (EP)

An internal electrical connection exists between the Exposed Thermal Pad (EP) and the V_{SS} pin. They must be connected to the same potential on the Printed Circuit Board (PCB).

For better thermal performance, it is recommended to add vias from the land area of EP to a copper layer on the other side of the PCB.

NOTES:

4.0 DEVICE OVERVIEW

The MCP73831/2 are highly advanced linear charge management controllers. Figure 4-1 depicts the operational flow algorithm from charge initiation to completion and automatic recharge.

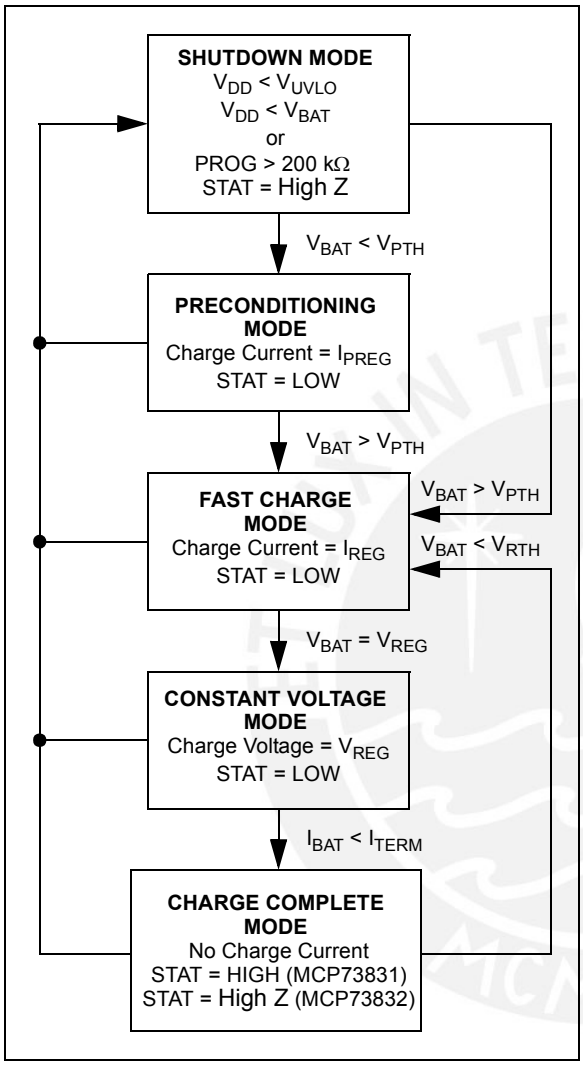


FIGURE 4-1: *Flowchart.*

4.1 Undervoltage Lockout (UVLO)

An internal UVLO circuit monitors the input voltage and keeps the charger in Shutdown mode until the input supply rises above the UVLO threshold. The UVLO circuitry has a built in hysteresis of 100 mV.

In the event a battery is present when the input power is applied, the input supply must rise to a level 150 mV above the battery voltage before the MCP73831/2 become operational.

The UVLO circuit places the device in Shutdown mode if the input supply falls to within +50 mV of the battery voltage. Again, the input supply must rise to a level 150 mV above the battery voltage before the MCP73831/2 become operational.

The UVLO circuit is always active. Whenever the input supply is below the UVLO threshold or within +50 mV of the voltage at the V_{BAT} pin, the MCP73831/2 are placed in Shutdown mode.

During any UVLO condition, the battery reverse discharge current is less than 2 µA.

4.2 Battery Detection

A 6 µA (typical) current is sourced by the V_{BAT} pin to determine if a battery is present or not. If the voltage at V_{BAT} rises to V_{REG} + 100 mV (typical), the device assumes that a battery is not present. If the voltage stays below V_{REG} + 100 mV (typical), the device assumes that a battery is detected. In order to correctly detect a battery insertion, the impedance seen by the V_{BAT} pin before the battery is connected must be greater than 2 MΩ.

4.3 Charge Qualification

For a charge cycle to begin, all UVLO conditions must be met and a battery or output load must be present. A charge current programming resistor must be connected from PROG to V_{SS}. If the PROG pin is open or floating, the MCP73831/2 are disabled and the battery reverse discharge current is less than 2 µA. In this manner, the PROG pin acts as a charge enable and can be used as a manual shutdown.

4.4 Preconditioning

If the voltage at the V_{BAT} pin is less than the preconditioning threshold, the MCP73831/2 enter a preconditioning or Trickle Charge mode. The preconditioning threshold is factory set. Refer to **Section 1.0 "Electrical Characteristics"** for preconditioning threshold options and the **Product Identification System** for standard options.

In this mode, the MCP73831/2 supply a percentage of the charge current (established with the value of the resistor connected to the PROG pin) to the battery. The percentage or ratio of the current is factory set. Refer to **Section 1.0 "Electrical Characteristics"** for preconditioning current options and the **Product Identification System** for standard options.

When the voltage at the V_{BAT} pin rises above the preconditioning threshold, the MCP73831/2 enter the Constant-Current or Fast Charge mode.

4.5 Fast Charge Constant-Current Mode

During the Constant-Current mode, the programmed charge current is supplied to the battery or load. The charge current is established using a single resistor from PROG to V_{SS}. Constant-Current mode is maintained until the voltage at the V_{BAT} pin reaches the regulation voltage, V_{REG}.

4.6 Constant-Voltage Mode

When the voltage at the V_{BAT} pin reaches the regulation voltage, V_{REG}, constant voltage regulation begins. The regulation voltage is factory set to 4.2V, 4.35V, 4.40V or 4.50V with a tolerance of ±0.75%.

4.7 Charge Termination

The charge cycle is terminated when, during Constant-Voltage mode, the average charge current diminishes below a percentage of the programmed charge current (established with the value of the resistor connected to the PROG pin). A 1 ms filter time on the termination comparator ensures that transient load conditions do not result in premature charge cycle termination. The percentage or ratio of the current is factory set. Refer to **Section 1.0 "Electrical Characteristics"** for charge termination current options and the **Product Identification System** for standard options.

The charge current is latched off and the MCP73831/2 enter a Charge Complete mode.

4.8 Automatic Recharge

The MCP73831/2 continuously monitor the voltage at the V_{BAT} pin in the Charge Complete mode. If the voltage drops below the recharge threshold, another charge cycle begins and current is once again supplied to the battery or load. The recharge threshold is factory set. Refer to **Section 1.0 "Electrical Characteristics"** for recharge threshold options and the **Product Identification System** for standard options.

4.9 Thermal Regulation

The MCP73831/2 limit the charge current based on the die temperature. The thermal regulation optimizes the charge cycle time while maintaining device reliability. Figure 4-2 depicts the thermal regulation for the MCP73831/2.

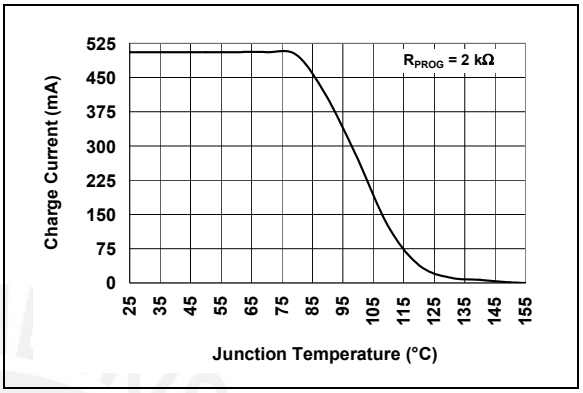


FIGURE 4-2: *Thermal Regulation.*

4.10 Thermal Shutdown

The MCP73831/2 suspend charge if the die temperature exceeds 150°C. Charging will resume when the die temperature has cooled by approximately 10°C.

5.0 DETAILED DESCRIPTION

5.1 Analog Circuitry

5.1.1 BATTERY MANAGEMENT INPUT SUPPLY (V_{DD})

The V_{DD} pin is the input supply pin for the MCP73831/2 devices. The MCP73831/2 automatically enter a Power-Down mode if the voltage on the V_{DD} input falls below the UVLO voltage (V_{STOP}). This feature prevents draining the battery pack when the V_{DD} supply is not present.

5.1.2 CURRENT REGULATION SET (PROG)

Fast charge current regulation can be scaled by placing a programming resistor (R_{PROG}) from the PROG input to V_{SS}. The program resistor and the charge current are calculated using the following equation:

$$I_{REG} = \frac{1000V}{R_{PROG}}$$

Where:

R_{PROG} = kOhms

I_{REG} = milliampere

The preconditioning trickle charge current and the charge termination current are ratiometric to the fast charge current based on the selected device options.

5.1.3 BATTERY CHARGE CONTROL OUTPUT (V_{BAT})

The battery charge control output is the drain terminal of an internal P-channel MOSFET. The MCP73831/2 provide constant current and voltage regulation to the battery pack by controlling this MOSFET in the linear region. The battery charge control output should be connected to the positive terminal of the battery pack.

5.2 Digital Circuitry

5.2.1 STATUS INDICATOR (STAT)

The charge status output of the MCP73831 has three different states: High (H), Low (L), and High-Impedance (High Z). The charge status output of the MCP73832 is open-drain. It has two different states: Low (L) and High-Impedance (High Z). The charge status output can be used to illuminate one, two or tri-color LEDs. Optionally, the charge status output can be used as an interface to a host microcontroller.

Table 5-1 summarizes the state of the status output during a charge cycle.

TABLE 5-1: STATUS OUTPUT

Charge Cycle State	STAT1	
	MCP73831	MCP73832
Shutdown	High Z	High Z
No Battery Present	High Z	High Z
Preconditioning	L	L
Constant-Current Fast Charge	L	L
Constant Voltage	L	L
Charge Complete – Standby	H	High Z

5.2.2 DEVICE DISABLE (PROG)

The current regulation set input pin (PROG) can be used to terminate a charge at any time during the charge cycle, as well as to initiate a charge cycle or initiate a recharge cycle.

Placing a programming resistor from the PROG input to V_{SS} enables the device. Allowing the PROG input to float or by applying a logic-high input signal, disables the device and terminates a charge cycle. When disabled, the device's supply current is reduced to 25 µA, typically.

NOTES:

6.0 APPLICATIONS

The MCP73831/2 are designed to operate in conjunction with a host microcontroller or in a stand-alone application. The MCP73831/2 provide the preferred charge algorithm for Lithium-Ion and Lithium-Polymer cells. The algorithm uses a constant current followed by a constant voltage charging method. Figure 6-1 depicts a typical stand-alone application circuit, while Figure 6-2 and Figure 6-3 depict the accompanying charge profile.

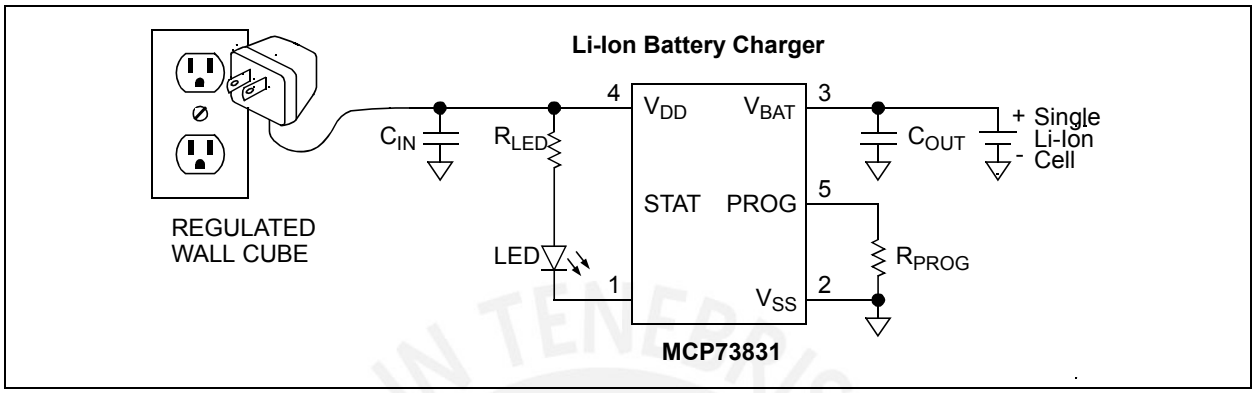


FIGURE 6-1: *Typical Application Circuit.*

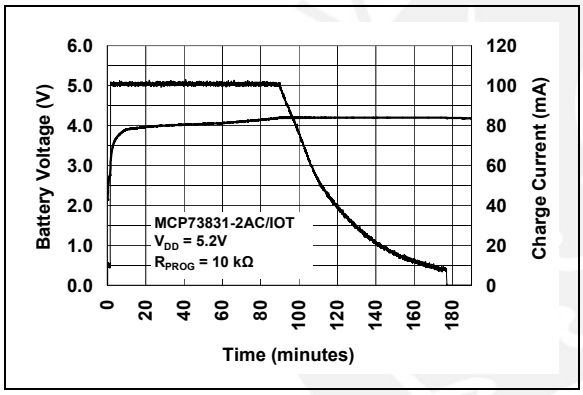


FIGURE 6-2: *Typical Charge Profile (180 mAh Battery).*

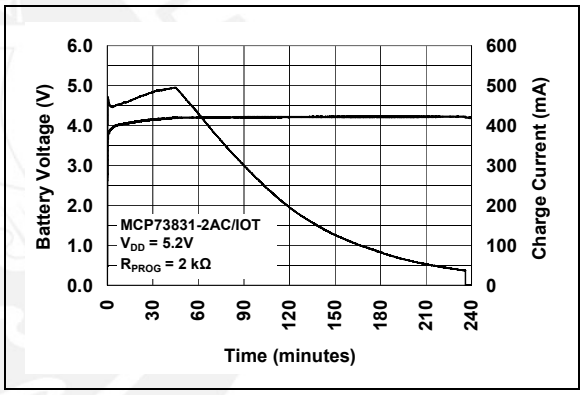


FIGURE 6-3: *Typical Charge Profile in Thermal Regulation (1000 mAh Battery).*

6.1 Application Circuit Design

Due to the low efficiency of linear charging, the most important factors are thermal design and cost, which are a direct function of the input voltage, output current and thermal impedance between the battery charger and the ambient cooling air. The worst-case situation is when the device has transitioned from the Preconditioning mode to the Constant-Current mode. In this situation, the battery charger has to dissipate the maximum power. A trade-off must be made between the charge current, cost and thermal requirements of the charger.

6.1.1 COMPONENT SELECTION

Selection of the external components in Figure 6-1 is crucial to the integrity and reliability of the charging system. The following discussion is intended as a guide for the component selection process.

6.1.1.1 Current Programming Resistor (R_{PROG})

The preferred fast charge current for Lithium-Ion cells is at the 1C rate, with an absolute maximum current at the 2C rate. For example, a 500 mAh battery pack has a preferred fast charge current of 500 mA. Charging at this rate provides the shortest charge cycle times without degradation to the battery pack performance or life.

6.1.1.2 Input Overvoltage Protection (IOVP)

Input overvoltage protection must be used when the input power source is hot-pluggable. This includes USB cables and Wall-type power supplies. The cabling of these supplies acts as an inductor. When the supplies are connected/disconnected from the system, large voltage transients are created which may damage the system circuitry. These transients should be snubbed out. A transzorb connected from the V+ input supply connector to the 0V ground reference will snub the transients.

6.1.1.3 Thermal Considerations

The worst-case power dissipation in the battery charger occurs when the input voltage is at the maximum and the device has transitioned from the Preconditioning mode to the Constant-Current mode. In this case, the power dissipation is:

$$PowerDissipation = (V_{DDMAX} - V_{PTHMIN}) \times I_{REGMAX}$$

Where:

V_{DDMAX}	=	the maximum input voltage
I_{REGMAX}	=	the maximum fast charge current
V_{PTHMIN}	=	the minimum transition threshold voltage

Power dissipation with a 5V, ±10% input voltage source is:

$$PowerDissipation = (5.5V - 2.7V) \times 550mA = 1.54W$$

This power dissipation with the battery charger in the SOT-23-5 package will cause thermal regulation to be entered as depicted in Figure 6-3. Alternatively, the 2mm x 3mm DFN package could be utilized to reduce charge cycle times.

6.1.1.4 External Capacitors

The MCP73831/2 are stable with or without a battery load. In order to maintain good AC stability in the Constant-Voltage mode, a minimum capacitance of 4.7 µF is recommended to bypass the V_{BAT} pin to V_{SS}. This capacitance provides compensation when there is no battery load. In addition, the battery and interconnections appear inductive at high frequencies. These elements are in the control feedback loop during Constant-Voltage mode. Therefore, the bypass capacitance may be necessary to compensate for the inductive nature of the battery pack.

Virtually any good quality output filter capacitor can be used, independent of the capacitor's minimum Effective Series Resistance (ESR) value. The actual value of the capacitor (and its associated ESR) depends on the output load current. A 4.7 µF ceramic, tantalum or aluminum electrolytic capacitor at the output is usually sufficient to ensure stability for output currents up to a 500 mA.

6.1.1.5 Reverse-Blocking Protection

The MCP73831/2 provide protection from a faulted or shorted input. Without the protection, a faulted or shorted input would discharge the battery pack through the body diode of the internal pass transistor.

6.1.1.6 Charge Inhibit

The current regulation set input pin (PROG) can be used to terminate a charge at any time during the charge cycle, as well as to initiate a charge cycle or initiate a recharge cycle.

Placing a programming resistor from the PROG input to V_{SS} enables the device. Allowing the PROG input to float or by applying a logic-high input signal, disables the device and terminates a charge cycle. When disabled, the device's supply current is reduced to 25 µA, typically.

6.1.1.7 Charge Status Interface

A status output provides information on the state of charge. The output can be used to illuminate external LEDs or interface to a host microcontroller. Refer to Table 5-1 for a summary of the state of the status output during a charge cycle.

6.2 PCB Layout Issues

For optimum voltage regulation, place the battery pack as close as possible to the device's V_{BAT} and V_{SS} pins. This is recommended to minimize voltage drops along the high current-carrying PCB traces.

If the PCB layout is used as a heat sink, adding many vias in the heat sink pad can help conduct more heat to the PCB backplane, thus reducing the maximum junction temperature. Figure 6-4 and Figure 6-5 depict a typical layout with PCB heatsinking.

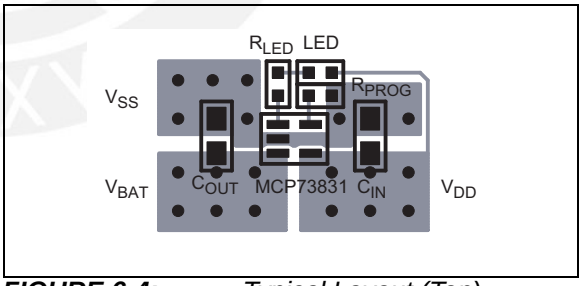


FIGURE 6-4: *Typical Layout (Top).*

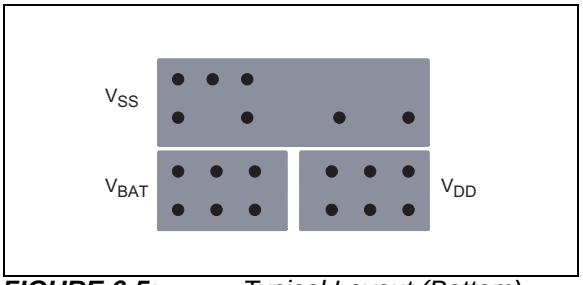
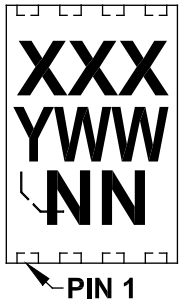


FIGURE 6-5: *Typical Layout (Bottom).*

7.0 PACKAGING INFORMATION

7.1 Package Marking Information

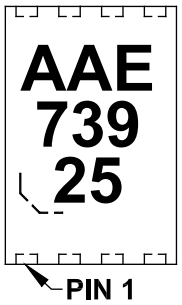
8-Lead DFN (2x3x0.9 mm)



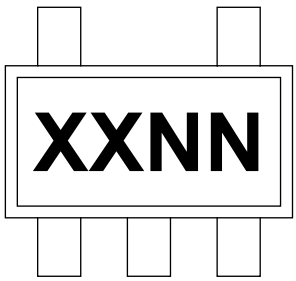
Device	Code
MCP73831T-2ACI/MC	AAE
MCP73831T-2ATI/MC	AAF
MCP73831T-2DCI/MC	AAG
MCP73831T-3ACI/MC	AAH
MCP73831T-4ADI/MC	AAJ
MCP73831T-5ACI/MC	AAK
MCP73832T-2ACI/MC	AAL
MCP73832T-2ATI/MC	AAM
MCP73832T-2DCI/MC	AAP
MCP73832T-3ACI/MC	AAQ
MCP73832T-4ADI/MC	AAR
MCP73832T-5ACI/MC	AAS

Note: Applies to 8-Lead DFN

Example



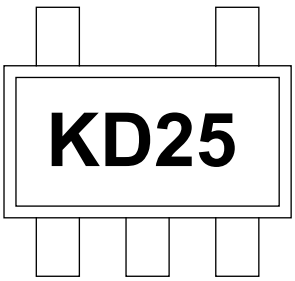
5-Lead SOT-23



Device	Code
MCP73831T-2ACI/OT	KDNN
MCP73831T-2ATI/OT	KENN
MCP73831T-2DCI/OT	KFNN
MCP73831T-3ACI/OT	KGNN
MCP73831T-4ADI/OT	KHNN
MCP73831T-5ACI/OT	KJNN
MCP73832T-2ACI/OT	KKNN
MCP73832T-2ATI/OT	KLNN
MCP73832T-2DCI/OT	KMNN
MCP73832T-3ACI/OT	KPNN
MCP73832T-4ADI/OT	KQNN
MCP73832T-5ACI/OT	KRNN
MCP73832T-2DFI/OT	LUNN

Note: Applies to 5-Lead SOT-23

Example

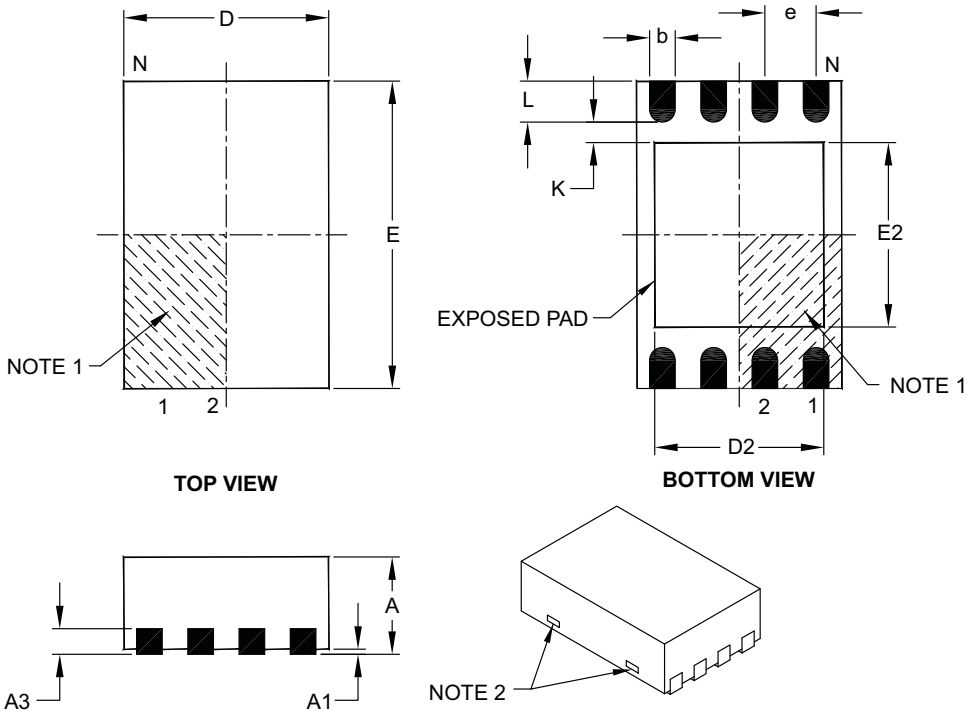


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free Compliant JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

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.

8-Lead Plastic Dual Flat, No Lead Package (MC) – 2x3x0.9 mm Body [DFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		8	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Length	D		2.00 BSC	
Overall Width	E		3.00 BSC	
Exposed Pad Length	D2	1.30	–	1.55
Exposed Pad Width	E2	1.50	–	1.75
Contact Width	b	0.20	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

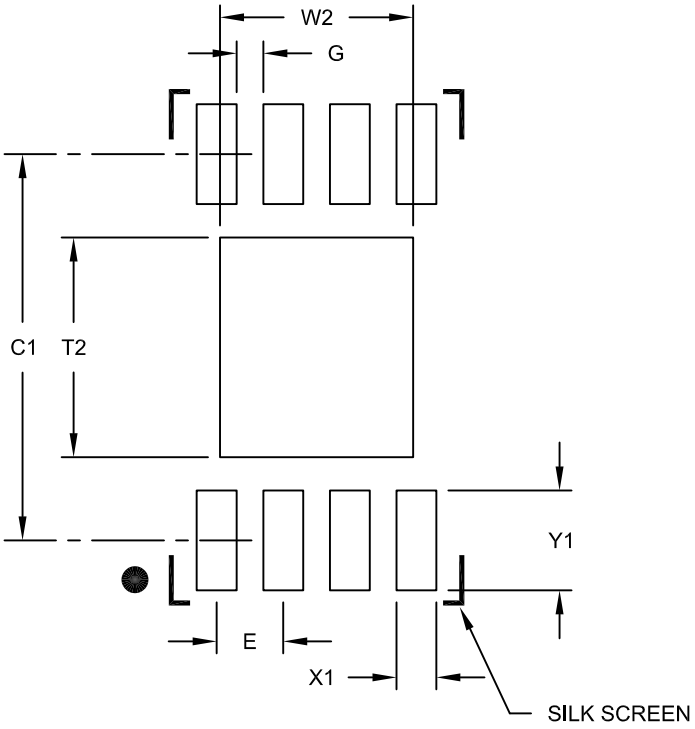
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package may have one or more exposed tie bars at ends.
3. Package is saw singulated.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-123C

8-Lead Plastic Dual Flat, No Lead Package (MC) - 2x3x0.9mm Body [DFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			1.45
Optional Center Pad Length	T2			1.75
Contact Pad Spacing	C1		2.90	
Contact Pad Width (X8)	X1			0.30
Contact Pad Length (X8)	Y1			0.75
Distance Between Pads	G	0.20		

Notes:

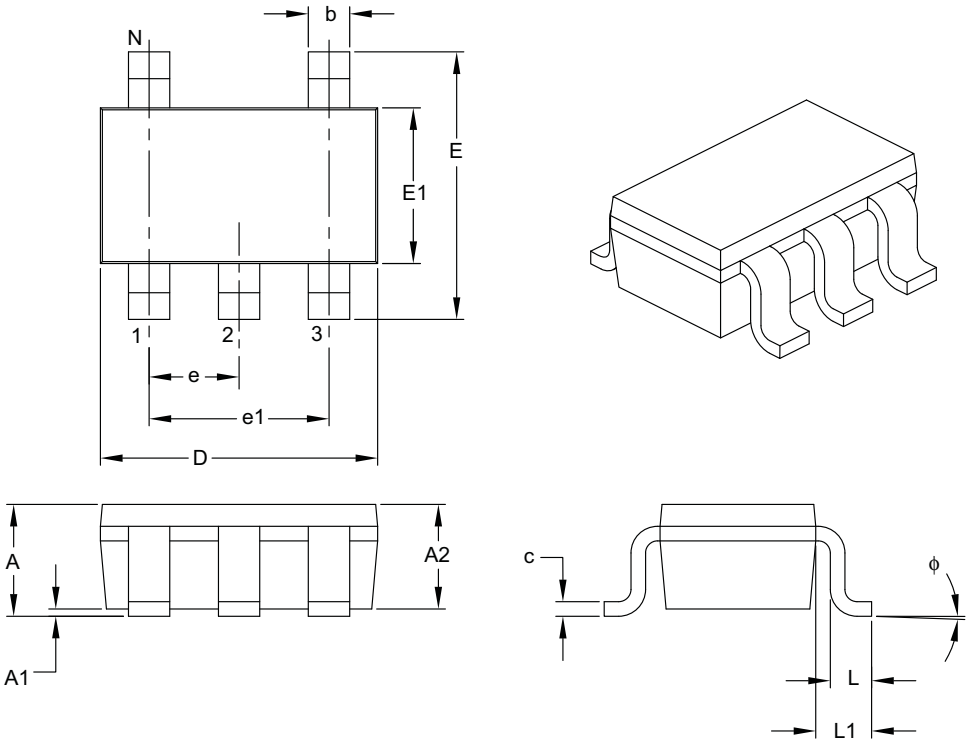
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2123B

5-Lead Plastic Small Outline Transistor (OT) [SOT-23]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		5	
Lead Pitch	e		0.95 BSC	
Outside Lead Pitch	e1		1.90 BSC	
Overall Height	A	0.90	–	1.45
Molded Package Thickness	A2	0.89	–	1.30
Standoff	A1	0.00	–	0.15
Overall Width	E	2.20	–	3.20
Molded Package Width	E1	1.30	–	1.80
Overall Length	D	2.70	–	3.10
Foot Length	L	0.10	–	0.60
Footprint	L1	0.35	–	0.80
Foot Angle	ϕ	0°	–	30°
Lead Thickness	c	0.08	–	0.26
Lead Width	b	0.20	–	0.51

Notes:

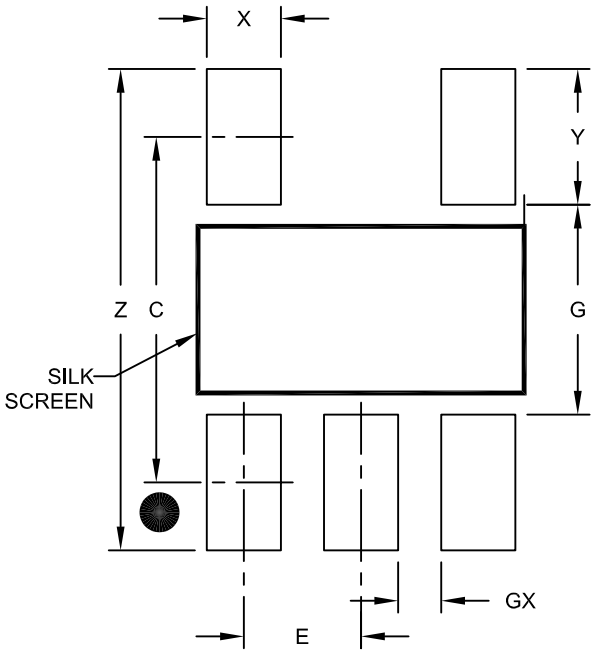
1. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.127 mm per side.
2. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-091B

5-Lead Plastic Small Outline Transistor (OT) [SOT-23]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.95 BSC	
Contact Pad Spacing	C		2.80	
Contact Pad Width (X5)	X			0.60
Contact Pad Length (X5)	Y			1.10
Distance Between Pads	G	1.70		
Distance Between Pads	GX	0.35		
Overall Width	Z			3.90

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2091A

APPENDIX A: REVISION HISTORY

Revision G (July 2014)

The following is the list of modifications:

1. Updated the **"DC Characteristics"** table.
2. Added **Section 6.1.1.2 "Input Overvoltage Protection (IOVP)"**.

Revision F (June 2013)

The following is the list of modifications:

3. Updated the Functional Block Diagram.
4. Added the Battery Detection parameter and related information in the **"DC Characteristics"** table.
5. Added new section **Section 4.2 "Battery Detection"**.
6. Minor grammatical and spelling corrections.

Revision E (September 2008)

The following is the list of modifications:

1. **Package Types:** Changed DFN pinout diagram.
2. **Section 1.0 "Electrical Characteristics"**: Changed "Charge Impedance Range from 20 kΩ to 67 kΩ.
3. **Section 1.0 "Electrical Characteristics"**: Misc. Formatting changes.
4. **Section 2.0 "Typical Performance Curves"**: Updated Figure 2-4.
5. **Section 3.0 "Pin Description"**: Added Exposed Pad pin to table and added **Section 3.6 "Exposed Thermal Pad (EP)"**.
6. Updated **Appendix A: "Revision History"**
7. Added Land Pattern Package Outline Drawing for 2x3 DFN package.

Revision D (April 2008)

The following is the list of modifications:

1. Changed Charge Termination Current Ratio to 8.5% minimum and 11.5% maximum.

Revision C (October 2007)

The following is the list of modifications:

1. Numerous edits throughout document.
2. Added note to the Temperature Specifications table.
3. Updated Figure 2-4.

Revision B (March 2006)

The following is the list of modifications:

1. Added MCP73832 through document.

Revision A (November 2005)

Original Release of this Document.

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.	X	XX	X	/XX
Device	V_{REG}	Options	Temperature Range	Package

Device:

MCP73831:	Single-Cell Charge Controller
MCP73831T:	Single-Cell Charge Controller (Tape and Reel)
MCP73832	Single-Cell Charge Controller
MCP73832T:	Single-Cell Charge Controller (Tape and Reel)

Regulation Voltage:

Code	V_{REG}
2	4.20V
3	4.35V
4	4.40V
5	4.50V

Options: *

Code	I_{PREG}/I_{REG}	V_{PTH}/V_{REG}	I_{TERM}/I_{REG}	V_{RTH}/V_{REG}
AC	10	66.5	7.5	96.5
AD	10	66.5	7.5	94
AT	10	71.5	20	94
DC	100	x	7.5	96.5

* Consult Factory for Alternative Device Options

Temperature Range: I = -40°C to +85°C (Industrial)

Package:

MC	= Dual-Flat, No-Lead (2x3 mm body), 8-Lead
OT	= Small Outline Transistor (SOT23), 5-Lead

Examples: *

a) MCP73831-2ACI/OT: 4.20V V_{REG}, Options AC, 5LD SOT23 Pkg
b) MCP73831T-2ACI/OT: Tape and Reel, 4.20V V_{REG}, Options AC, 5LD SOT23 Pkg
c) MCP73832-2ACI/MC: 4.20V V_{REG}, Options AC, 8LD DFN Package
d) MCP73832T-2ACI/MC: Tape and Reel, 4.20V V_{REG}, Options AC, 8LD DFN Package

a) MCP73831-2ATI/OT: 4.20V V_{REG}, Options AT, 5LD SOT23 Pkg
b) MCP73831T-2ATI/OT: Tape and Reel, 4.20V V_{REG}, Options AT, 5LD SOT23 Pkg
c) MCP73832-2ATI/MC: 4.20V V_{REG}, Options AT, 8LD DFN Package
d) MCP73832T-2ATI/MC: Tape and Reel, 4.20V V_{REG}, Options AT, 8LD DFN Package

a) MCP73831-2DCI/OT: 4.20V V_{REG}, Options DC, 5LD SOT23 Pkg
b) MCP73831T-2DCI/OT: Tape and Reel, 4.20V V_{REG}, Options DC, 5LD SOT23 Pkg
c) MCP73832-2DCI/MC: 4.20V V_{REG}, Options DC, 8LD DFN Package
d) MCP73832T-2DCI/MC: Tape and Reel, 4.20V V_{REG}, Options DC, 8LD DFN Package

a) MCP73831-3ACI/OT: 4.35V V_{REG}, Options AC, 5LD SOT23 Pkg
b) MCP73831T-3ACI/OT: Tape and Reel, 4.35V V_{REG}, Options AC, 5LD SOT23 Pkg
c) MCP73832-3ACI/MC: 4.35V V_{REG}, Options AC, 8LD DFN Package
d) MCP73832T-3ACI/MC: Tape and Reel, 4.35V V_{REG}, Options AC, 8LD DFN Package

a) MCP73831-4ADI/OT: 4.40V V_{REG}, Options AD, 5LD SOT23 Pkg
b) MCP73831T-4ADI/OT: Tape and Reel, 4.40V V_{REG}, Options AD, 5LD SOT23 Pkg
c) MCP73832-4ADI/MC: 4.40V V_{REG}, Options AD, 8LD DFN Package
d) MCP73832T-4ADI/MC: Tape and Reel, 4.40V V_{REG}, Options AD, 8LD DFN Package

a) MCP73831-5ACI/OT: 4.50V V_{REG}, Options AC, 5LD SOT23 Pkg
b) MCP73831T-5ACI/OT: Tape and Reel, 4.50V V_{REG}, Options AC, 5LD SOT23 Pkg
c) MCP73832-5ACI/MC: 4.50V V_{REG}, Options AC, 8LD DFN Package
d) MCP73832T-5ACI/MC: Tape and Reel, 4.50V V_{REG}, Options AC, 8LD DFN Package

* Consult Factory for Alternate Device Options

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families 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 Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is 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 products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-63276-375-4

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14

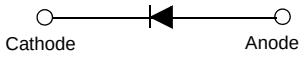
FAIRCHILD SEMICONDUCTOR®

September 2009

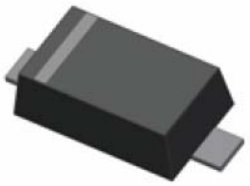
RB521S30
Schottky Barrier Diodes

Features

- Low Forward Voltage Drop
- Flat Lead, Surface Mount Device Under 0.70mm Height
- Extremely Small Outline Plastic Package SOD523F
- Moisture Level Sensitivity 1
- Pb-free Version and RoHS Compliant
- Matte Tin (Sn) Lead Finish
- Green Mold Compound



ELECTRICAL SYMBOL



SOD-523F
Band Indicates Cathode
RB521S30 Marking : 2B

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

Symbol	Parameter	Value	Units
V_{RRM}	Maximum Repetitive Reverse Voltage	30	V
$I_{F(AV)}$	Average Rectified Forward Current	200	mA
T_J	Operating Junction Temperature Range	-55 to +125	°C
T_{STG}	Storage Temperature Range	-55 to +125	°C

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

Thermal Characteristics

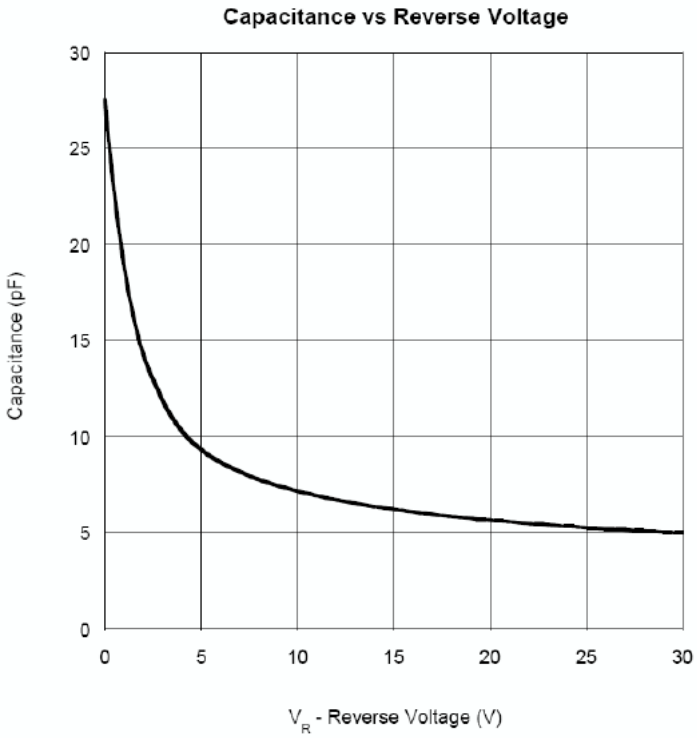
Symbol	Parameter	Value	Units
P_D	Total Device Dissipation (T_C=25°C)	200	mW
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	500	°C/W

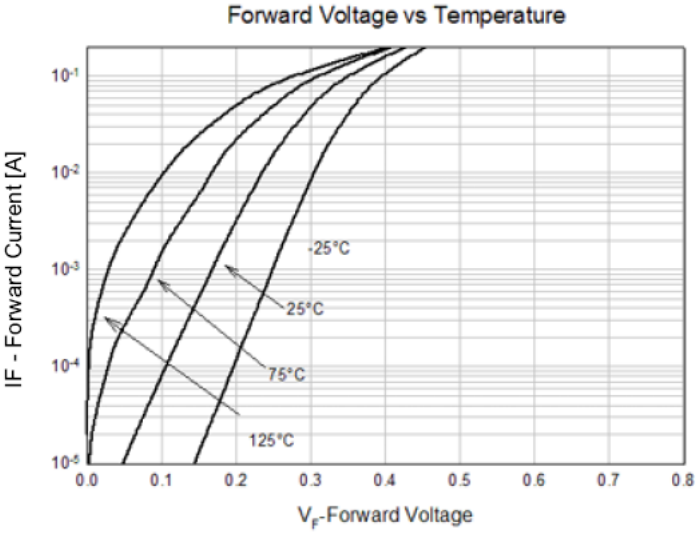
* Device mounted on FR-4 PCB minimum land pad.

Electrical Characteristics T_A=25°C unless otherwise noted

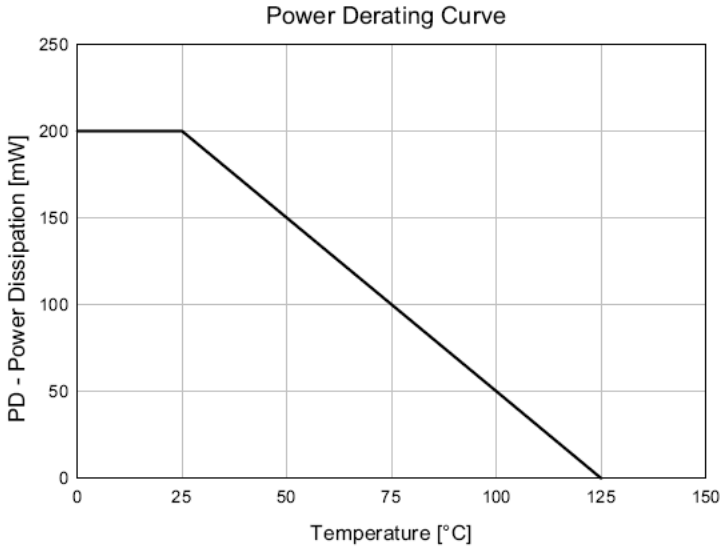
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
BV_R	Breakdown Voltage	I_R=500μA	30			V
I_R	Reverse Leakage Current	V_R=10V			30	μA
V_F	Forward Voltage	I_F=200mA			0.5	V

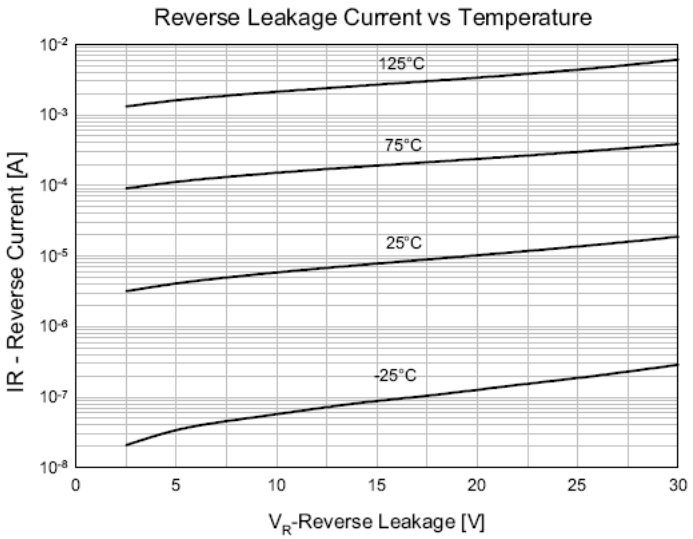
Typical Performance Characteristics





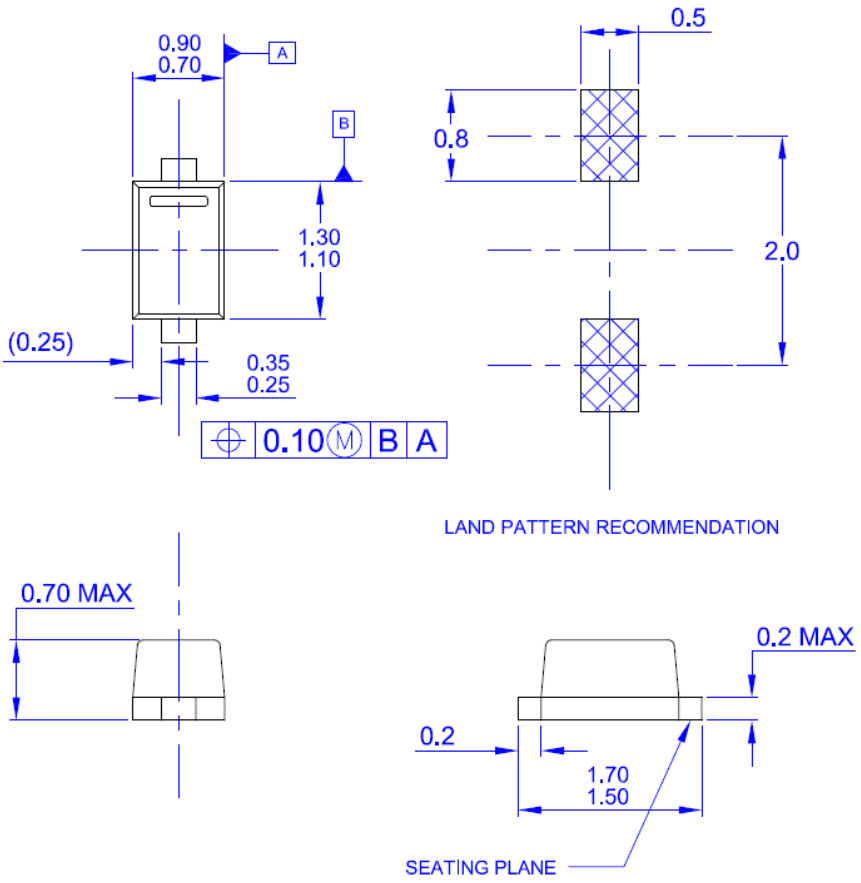
Typical Performance Characteristics (Continue)





Physical Dimension

SOD-523F



NOTES: UNLESS OTHERWISE SPECIFIED

A) PACKAGE REFERENCE: THIS PACKAGE OUTLINE CONFORMS TO JEITA SC-79.
B) ALL DIMENSIONS ARE IN MILLIMETERS.
C) DRAWING CONFORMS TO ASME Y14.5M - 1994
D) DIMENSIONS ARE EXCLUSIVE OF BURRS, MOLD FLASH, AND TIE BAR EXTRUSIONS.
E) LANDPATTERN RECOMMENDATION IS BASED ON IPC7351A STANDARD SOD1609X65M.
F) DRAWING NUMBER AND REVISION:MKT-SOD523F1rev1

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.

AccuPower™
Auto-SPM™
Build it Now™
CorePLUS™
CorePOWER™
CROSSVOLT™
CTL™
Current Transfer Logic™
EcoSPARK®
EfficentMax™
EZSWITCH™*
™*
®
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™
MegaBuck™
MICROCOUPLER™
MicroFET™
MicroPak™
MillerDrive™
MotionMax™
Motion-SPM™
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™
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™
TriFault Detect™
TRUECURRENT™*
µSerDes™
SerDes
UHC®
Ultra FRFET™
UniFET™
VCX™
VisualMax™
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. I41

500nA, I_Q 150mA, Ultra-Low Quiescent Current Low-Dropout Linear Regulator

FEATURES

- **Low I_Q: 500nA**
- **150mA, Low-Dropout Regulator**
- **Low-Dropout at +25°C, 130mV at 150mA**
- **Low-Dropout at +85°C, 175mV at 150mA**
- **3% Accuracy Over Load/Line/Temperature**
- **Available in Fixed Voltage Options (2.5V, 2.7V, and 2.8V) Using Innovative Factory EEPROM Programming**
- **Stable with a 1.0µF Ceramic Capacitor**
- **Thermal Shutdown and Overcurrent Protection**
- **CMOS Logic Level-Compatible Enable Pin**
- **Available in DDC (TSOT23-5) or DRV (2mm x 2mm SON-6) Packages**

APPLICATIONS

- **TI MSP430 Attach Applications**
- **Power Rails with Programming Mode**
- **Wireless Handsets, Smartphones, PDAs, MP3 Players, and Other Battery-Operated Handheld Products**

DESCRIPTION

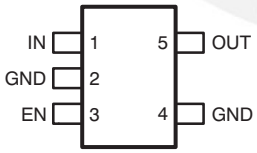
The TPS782 family of low-dropout regulators (LDOs) offers the benefits of ultra-low power (I_Q = 500nA), and miniaturized packaging (2-mm × 2-mm SON-6).

This LDO is designed specifically for battery-powered applications where ultra-low quiescent current is a critical parameter. The TPS782, with ultra-low I_Q (500nA), is ideal for microprocessors, memory cards, and smoke detectors.

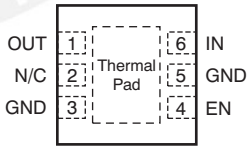
The ultra-low power and miniaturized packaging allow designers to customize power consumption for specific applications. Consult with your local factory representative for exact voltage options and ordering information; minimum order quantities may apply.

The TPS782 family is designed to be compatible with the TI MSP430 and other similar products. The enable pin (EN) is compatible with standard CMOS logic. This LDO is stable with any output capacitor greater than 1.0µF. Therefore, this device requires minimal board space because of miniaturized packaging and a potentially small output capacitor. The TPS782 series also features thermal shutdown and current limit to protect the device during fault conditions. All packages have an operating temperature range of T_J = –40°C to +125°C. For high-performance applications that require a dual-level voltage option, consider the TPS780 series, with an I_Q of 500nA and dynamic voltage scaling.

TPS782xxDDC
TSOT23-5
(TOP VIEW)



TPS782xxDRV
2mm x 2mm SON-6
(TOP VIEW)



This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ORDERING INFORMATION(1) (2)

PRODUCT	V_{OUT}
TPS782**xx** ***yyy z***	**XX** is the nominal output voltage **YYY** is the package designator. **Z** is the tape and reel quantity (R = 3000, T = 250).

(1) For the most current package and ordering information see the Package Option Addendum at the end of this document, or visit the device product folder at www.ti.com.
(2) Additional output voltage combinations are available on a quick-turn basis using innovative, factory EEPROM programming. Minimum-order quantities apply; contact your sales representative for details and availability

ABSOLUTE MAXIMUM RATINGS(1)

At T_J = –40°C to +125°C, unless otherwise noted. All voltages are with respect to GND.

PARAMETER		TPS782xx	UNIT
Input voltage range, V_{IN}		–0.3 to +6.0	V
Enable		–0.3 to V_{IN} + 0.3V	V
Output voltage range, V_{OUT}		–0.3 to V_{IN} + 0.3V	V
Maximum output current, I_{OUT}		Internally limited	
Output short-circuit duration		Indefinite	
Total continuous power dissipation, P_{DISS}		See the Dissipation Ratings table	
ESD rating	Human body model (HBM)	2	kV
	Charged device model (CDM)	500	V
Operating junction temperature range, T_J		–40 to +125	°C
Storage temperature range, T_{STG}		–55 to +150	°C

(1) Stresses above these ratings may cause permanent damage. Exposure to absolute maximum conditions for extended periods may degrade device reliability. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those specified is not implied.

DISSIPATION RATINGS

BOARD	PACKAGE	$R_{\theta JC}$	$R_{\theta JA}$	DERATING FACTOR ABOVE T_A = +25°C	T_A < +25°C	T_A = +70°C	T_A = +85°C
High-K(1)	DRV	20°C/W	65°C/W	15.4mW/°C	1540mW	845mW	615mW
High-K(1)	DDC	90°C/W	200°C/W	5.0mW/°C	500mW	275mW	200mW

(1) The JEDEC high-K (2s2p) board used to derive this data was a 3-inch × 3-inch, multilayer board with 1-ounce internal power and ground planes and 2-ounce copper traces on top and bottom of the board.

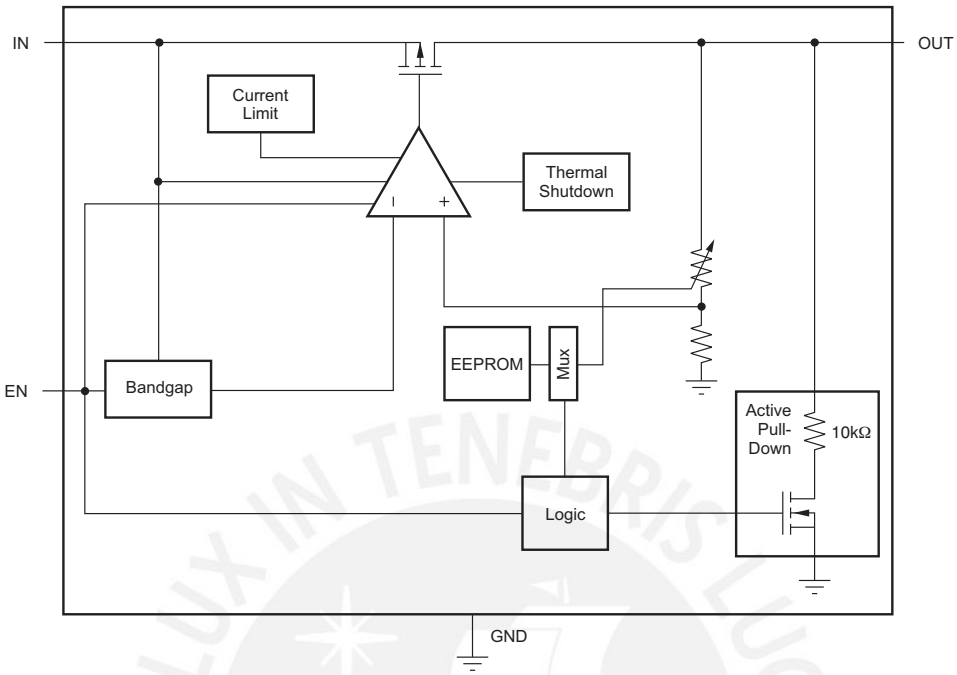
ELECTRICAL CHARACTERISTICS

Over operating temperature range (T_J = –40°C to +125°C), V_{IN} = $V_{OUT(NOM)}$ + 0.5V or 2.2V, whichever is greater; I_{OUT} = 100µA, V_{EN} = V_{IN}, C_{OUT} = 1.0µF, fixed V_{OUT} test conditions, unless otherwise noted. Typical values at T_J = +25°C.

PARAMETER			TEST CONDITIONS		TPS782xx MIN	TPS782xx TYP	TPS782xx MAX	UNIT
V_{IN}	Input voltage range				2.2		5.5	V
V_{OUT}	DC output accuracy	Nominal	T_J = +25°C		–2	±1	+2	%
		Over V_{IN}, I_{OUT}, temperature	V_{OUT} + 0.5V ≤ V_{IN} ≤ 5.5V, 0mA ≤ I_{OUT} ≤ 150mA		–3.0	±2.0	+3.0	%
$\Delta V_{OUT}/\Delta V_{IN}$	Line regulation		$V_{OUT(NOM)}$ + 0.5V ≤ V_{IN} ≤ 5.5V, I_{OUT} = 5mA			±1.0		%
$\Delta V_{OUT}/\Delta I_{OUT}$	Load regulation		0mA ≤ I_{OUT} ≤ 150mA			±2.0		%
V_{DO}	Dropout voltage(1)		V_{IN} = 95% $V_{OUT(NOM)}$, I_{OUT} = 150mA			130	250	mV
V_N	Output noise voltage		BW = 100Hz to 100kHz, V_{IN} = 2.2V, V_{OUT} = 1.2V, I_{OUT} = 1mA			86		μV_{RMS}
I_{CL}	Output current limit		V_{OUT} = 0.90 × $V_{OUT(NOM)}$		150	230	400	mA
I_{GND}	Ground pin current		I_{OUT} = 0mA			0.42	1.3	µA
			I_{OUT} = 150mA			8		µA
I_{SHDN}	Shutdown current (I_{GND})		V_{EN} ≤ 0.4V, 2.2V ≤ V_{IN} < 5.5V, T_J = –40°C to +100°C			18	130	nA
I_{EN}	EN pin current		V_{EN} = 5.5V				40	nA
PSRR	Power-supply rejection ratio		V_{IN} = 4.3V, V_{OUT} = 3.3V, I_{OUT} = 150mA	f = 10Hz		40		dB
				f = 100Hz		20		dB
				f = 1kHz		15		dB
t_{STR}	Startup time(2)		C_{OUT} = 1.0µF, V_{OUT} = 10% $V_{OUT(NOM)}$ to V_{OUT} = 90% $V_{OUT(NOM)}$			500		µs
t_{SHDN}	Shutdown time(3)		I_{OUT} = 150mA, C_{OUT} = 1.0µF, V_{OUT} = 2.8V, V_{OUT} = 90% $V_{OUT(NOM)}$ to V_{OUT} = 10% $V_{OUT(NOM)}$			500(4)		µs
T_{SD}	Thermal shutdown temperature		Shutdown, temperature increasing			+160		°C
			Reset, temperature decreasing			+140		°C
T_J	Operating junction temperature				–40		+125	°C

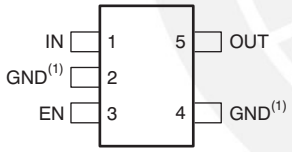
(1) V_{DO} is not measured for devices with $V_{OUT(NOM)}$ ≤ 2.3V because minimum V_{IN} = 2.2V.
(2) Time from V_{EN} = 1.2V to V_{OUT} = 90% ($V_{OUT(NOM)}$).
(3) Time from V_{EN} = 0.4V to V_{OUT} = 10% ($V_{OUT(NOM)}$).
(4) See *Shutdown* in the *Application Information* section for more details.

FUNCTIONAL BLOCK DIAGRAM

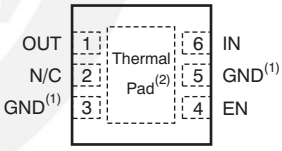


PIN CONFIGURATIONS

DDC PACKAGE
TSOT23-5
(TOP VIEW)



DRV PACKAGE
2mm x 2mm SON-6
(TOP VIEW)



(1) All ground pins must be connected to ground for proper operation.

(2) It is recommended that the thermal pad be grounded.

Table 1. PIN DESCRIPTIONS

PIN			DESCRIPTION
NAME	DRV	DDC	
OUT	1	5	Regulated output voltage pin. A small (1µF) ceramic capacitor is needed from this pin to ground to assure stability. See the *Input and Output Capacitor Requirements* in the Application Information section for more details.
N/C	2	—	Not connected.
EN	4	3	Driving the enable pin (EN) over 1.2V turns ON the regulator. Driving this pin below 0.4V puts the regulator into shutdown mode, reducing operating current to 18nA typical.
GND	3, 5	2, 4	ALL ground pins must be tied to ground for proper operation.
IN	6	1	Input pin. A small capacitor is needed from this pin to ground to assure stability. Typical input capacitor = 1.0µF. Both input and output capacitor grounds should be tied back to the IC ground with no significant impedance between them.
Thermal pad	Thermal pad	—	It is recommended that the thermal pad on the SON-6 package be connected to ground.

TYPICAL CHARACTERISTICS

Over the operating temperature range of T_J = –40°C to +125°C, V_{IN} = $V_{OUT(TYP)}$ + 0.5V or 2.2V, whichever is greater; I_{OUT} = 100µA, V_{EN} = V_{IN}, C_{OUT} = 1µF, and C_{IN} = 1µF, unless otherwise noted.

LINE REGULATION
I_{OUT} = 5mA, V_{OUT} = 2.7V (typ)
TPS78227

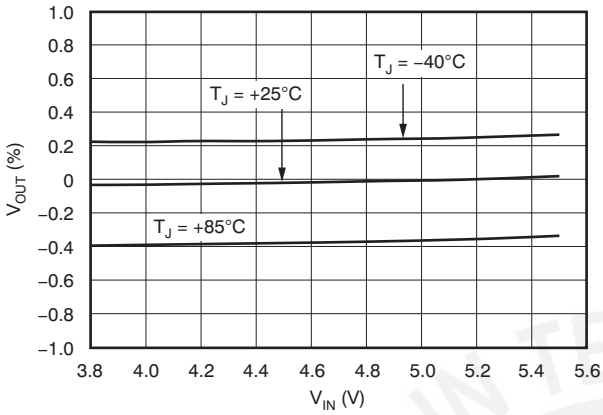


Figure 1.

LINE REGULATION
I_{OUT} = 150mA, V_{OUT} = 2.7V (typ)
TPS78227

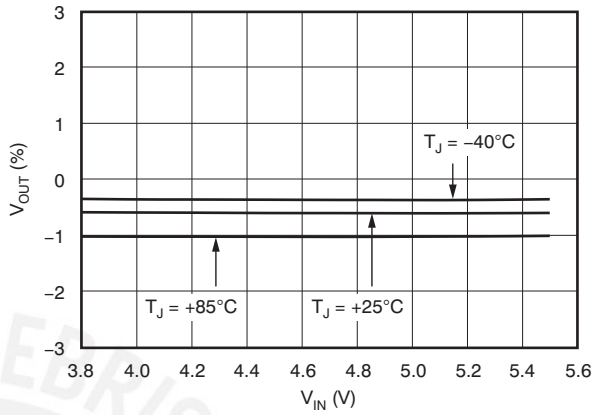


Figure 2.

LOAD REGULATION
V_{IN} = 3.8V, V_{OUT} = 2.7V
TPS78227

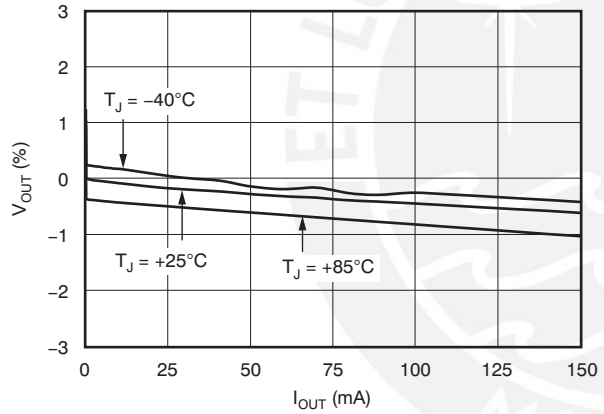


Figure 3.

DROPOUT VOLTAGE vs OUTPUT CURRENT
V_{OUT} = 2.7V (typ), V_{IN} = 0.95 × V_{OUT} (typ)
TPS78227

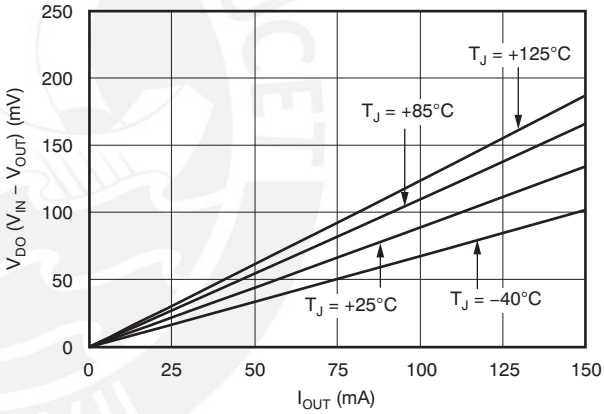


Figure 4.

DROPOUT VOLTAGE vs JUNCTION TEMPERATURE
V_{OUT} = 2.7V (typ), V_{IN} = 0.95 × V_{OUT} (typ)
TPS78227

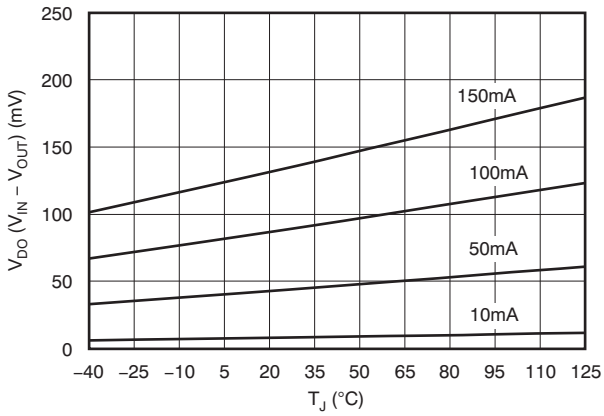


Figure 5.

GROUND PIN CURRENT vs INPUT VOLTAGE
I_{OUT} = 0mA, V_{OUT} = 3.3V
TPS78233

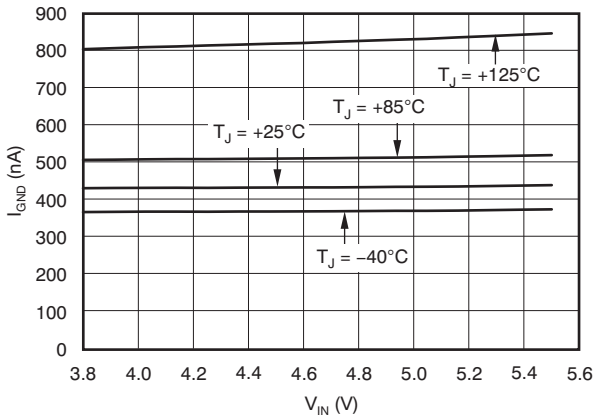


Figure 6.

TYPICAL CHARACTERISTICS (continued)

Over the operating temperature range of T_J = –40°C to +125°C, V_{IN} = $V_{OUT(TYP)}$ + 0.5V or 2.2V, whichever is greater; I_{OUT} = 100µA, V_{EN} = V_{IN}, C_{OUT} = 1µF, and C_{IN} = 1µF, unless otherwise noted.

GROUND PIN CURRENT vs INPUT VOLTAGE
I_{OUT} = 50mA, V_{OUT} = 2.7V
TPS78227

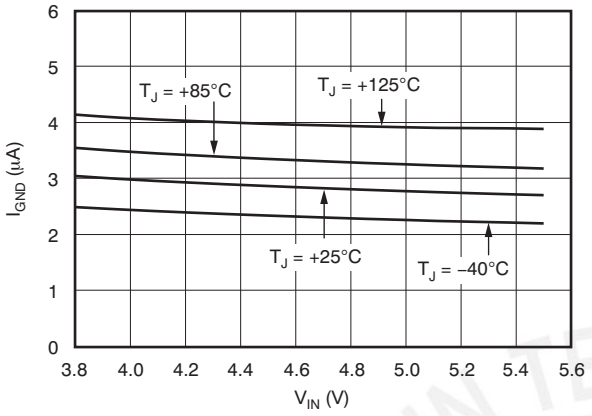


Figure 7.

GROUND PIN CURRENT vs INPUT VOLTAGE
I_{OUT} = 150mA, V_{OUT} = 2.7V
TPS78227

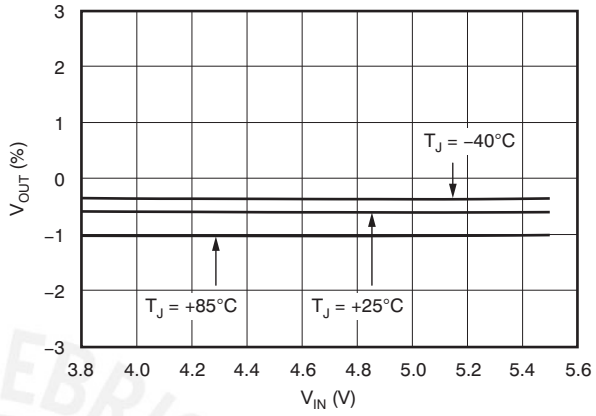


Figure 8.

CURRENT LIMIT vs INPUT VOLTAGE
V_{OUT} = 95% V_{OUT} (typ), V_{OUT} = 2.7V (typ)
TPS78227

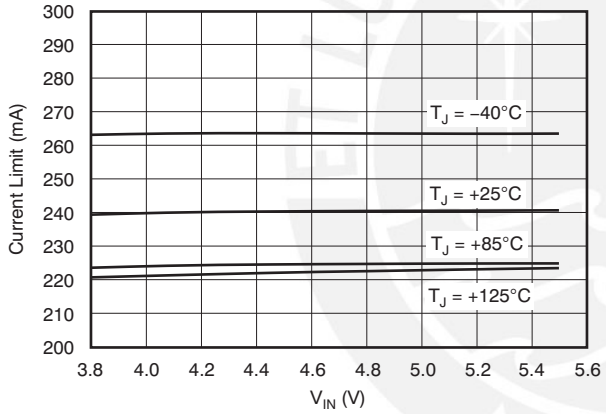


Figure 9.

ENABLE PIN CURRENT vs INPUT VOLTAGE
I_{OUT} = 100µA, V_{OUT} = 2.7V
TPS78227

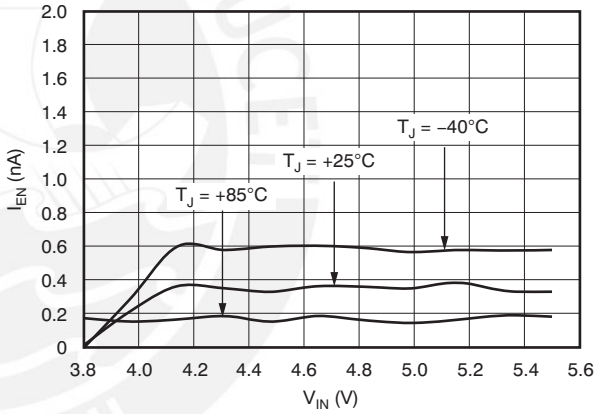


Figure 10.

ENABLE PIN HYSTERESIS vs JUNCTION TEMPERATURE
I_{OUT} = 1mA, TPS78227

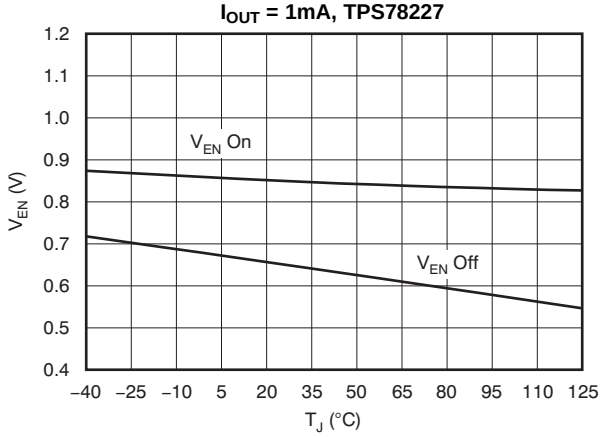


Figure 11.

%ΔV_{OUT} vs JUNCTION TEMPERATURE
V_{IN} = 3.3V, V_{OUT} = 2.7V (typ)
TPS78227

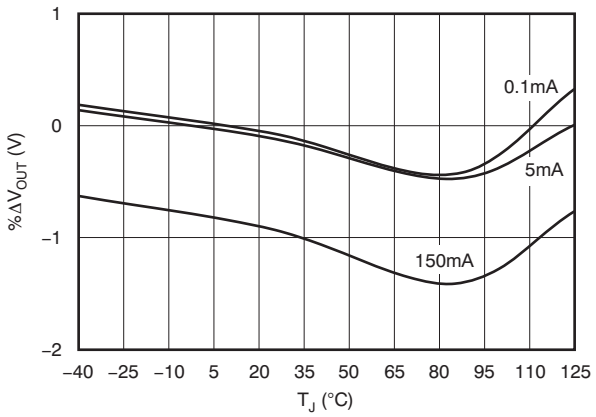


Figure 12.

TYPICAL CHARACTERISTICS (continued)

Over the operating temperature range of T_J = –40°C to +125°C, V_{IN} = $V_{OUT(TYP)}$ + 0.5V or 2.2V, whichever is greater; I_{OUT} = 100µA, V_{EN} = V_{IN}, C_{OUT} = 1µF, and C_{IN} = 1µF, unless otherwise noted.

%ΔV$_{OUT}$ vs JUNCTION TEMPERATURE
V$_{IN}$ = 3.7V, V$_{OUT}$ = 2.7V (typ)
TPS78227

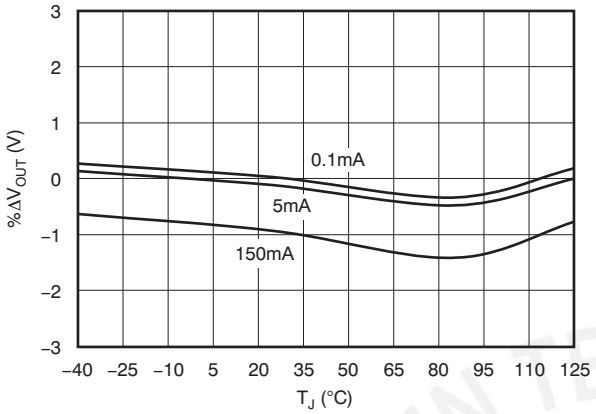


Figure 13.

OUTPUT SPECTRAL NOISE DENSITY vs FREQUENCY
C$_{IN}$ = 1µF, C$_{OUT}$ = 2.2µF, V$_{IN}$ = 3.2V
TPS78227

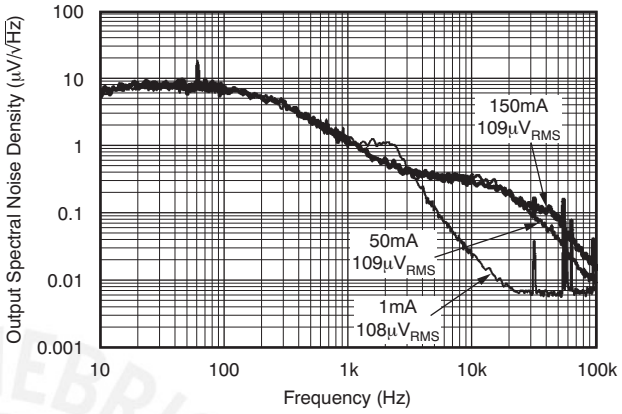


Figure 14.

RIPPLE REJECTION vs FREQUENCY
V$_{IN}$ = 4.2V, V$_{OUT}$ = 2.7V, C$_{OUT}$ = 2.2µF
TPS78227

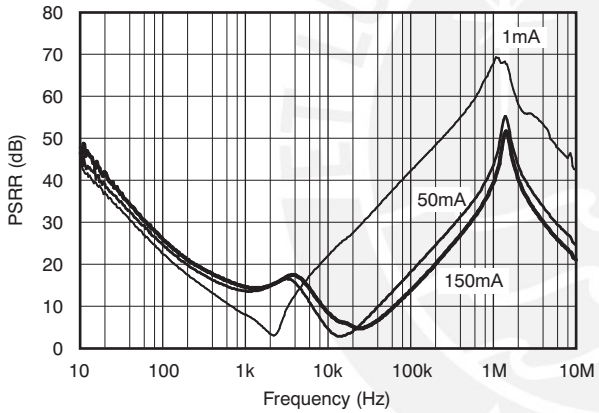


Figure 15.

INPUT VOLTAGE RAMP vs OUTPUT VOLTAGE
TPS78233

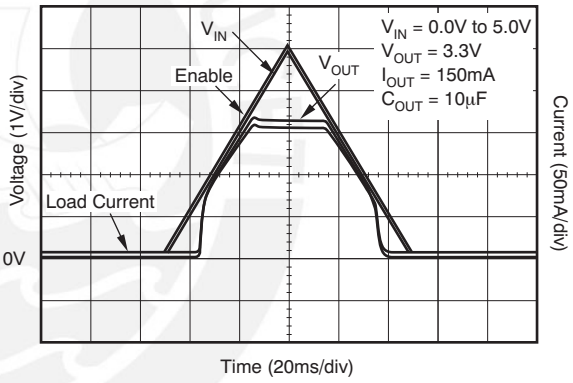


Figure 16.

OUTPUT VOLTAGE vs ENABLE (SLOW RAMP)
TPS78233

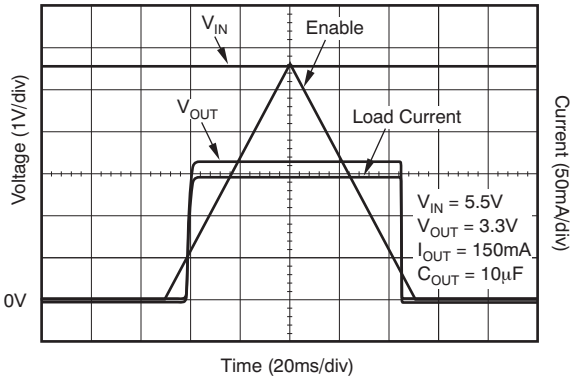


Figure 17.

INPUT VOLTAGE vs DELAY TO OUTPUT
TPS78222

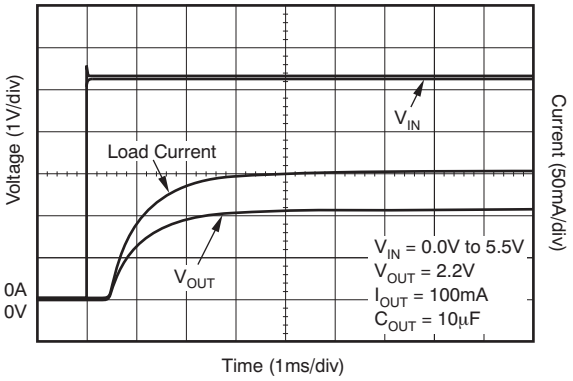


Figure 18.

TYPICAL CHARACTERISTICS (continued)

Over the operating temperature range of T_J = –40°C to +125°C, V_{IN} = $V_{OUT(TYP)}$ + 0.5V or 2.2V, whichever is greater; I_{OUT} = 100µA, V_{EN} = V_{IN}, C_{OUT} = 1µF, and C_{IN} = 1µF, unless otherwise noted.

LOAD TRANSIENT RESPONSE
TPS78233

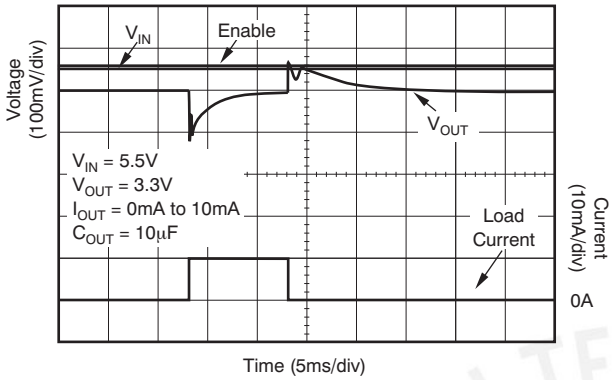


Figure 19.

ENABLE PIN vs OUTPUT VOLTAGE RESPONSE AND OUTPUT CURRENT
TPS78233

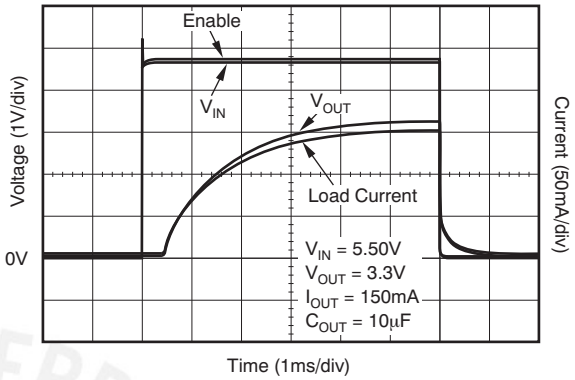


Figure 20.

ENABLE PIN vs OUTPUT VOLTAGE DELAY
TPS78233

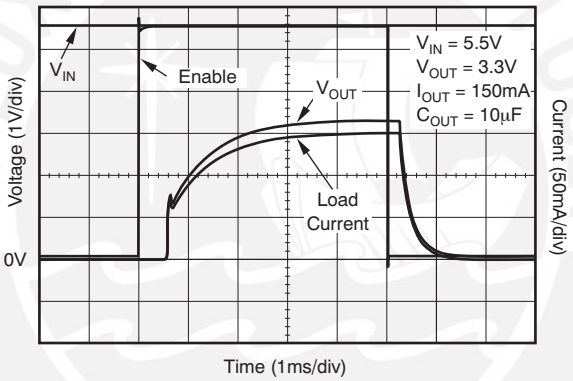


Figure 21.

APPLICATION INFORMATION

APPLICATION EXAMPLES

The TPS782 family of LDOs is factory-programmable to have a fixed output. Note that during startup or steady-state conditions, it is important that the EN pin voltage never exceed V_{IN} + 0.3V.

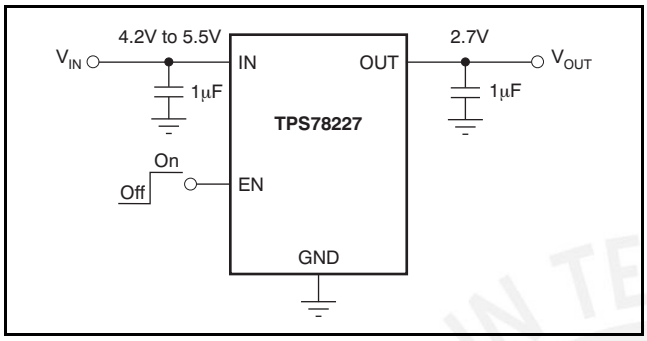


Figure 22. Typical Application Circuit

INPUT AND OUTPUT CAPACITOR REQUIREMENTS

Although an input capacitor is not required for stability, it is good analog design practice to connect a 0.1μF to 1.0μF low equivalent series resistance (ESR) capacitor across the input supply near the regulator. This capacitor counteracts reactive input sources and improves transient response, noise rejection, and ripple rejection. A higher-value capacitor may be necessary if large, fast rise-time load transients are anticipated, or if the device is not located near the power source. If source impedance is not sufficiently low, a 0.1μF input capacitor may be necessary to ensure stability.

The TPS782 series are designed to be stable with standard ceramic capacitors with values of 1.0μF or larger at the output. X5R- and X7R-type capacitors are best because they have minimal variation in value and ESR over temperature. Maximum ESR should be less than 1.0Ω. With tolerance and dc bias effects, the minimum capacitance to ensure stability is 1μF.

BOARD LAYOUT RECOMMENDATIONS TO IMPROVE PSRR AND NOISE PERFORMANCE

To improve ac performance (such as PSRR, output noise, and transient response), it is recommended that the printed circuit board (PCB) be designed with separate ground planes for V_{IN} and V_{OUT}, with each ground plane connected only at the GND pin of the device. In addition, the ground connection for the output capacitor should connect directly to the GND pin of the device. High ESR capacitors may degrade PSRR.

INTERNAL CURRENT LIMIT

The TPS782 is internally current-limited to protect the regulator during fault conditions. During current limit, the output sources a fixed amount of current that is largely independent of output voltage. For reliable operation, the device should not be operated in a current limit state for extended periods of time.

The PMOS pass element in the TPS782 series has a built-in body diode that conducts current when the voltage at OUT exceeds the voltage at IN. This current is not limited, so if extended reverse voltage operation is anticipated, external limiting to 5% of rated output current may be appropriate.

SHUTDOWN

The enable pin (EN) is active high and is compatible with standard and low-voltage CMOS levels. When shutdown capability is not required, EN should be connected to the IN pin, as shown in Figure 23. The TPS782 series, with internal active output pull-down circuitry, discharges the output to within 5% V_{OUT} with a time (*t*) shown in Equation 1:

$$t = 3 \left(\frac{10k\Omega \times R_L}{10k\Omega + R_L} \right) \times C_{OUT} \quad (1)$$

Where:

R_L= output load resistance

C_{OUT} = output capacitance

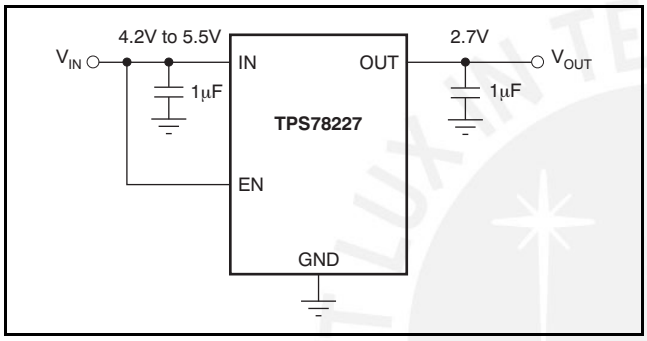


Figure 23. Circuit Showing EN Tied High when Shutdown Capability is Not Required

DROPOUT VOLTAGE

The TPS782 series use a PMOS pass transistor to achieve low dropout. When (V_{IN} – V_{OUT}) is less than the dropout voltage (V_{DO}), the PMOS pass device is the linear region of operation and the input-to-output resistance is the $R_{DS(ON)}$ of the PMOS pass element. V_{DO} approximately scales with output current because the PMOS device behaves like a resistor in dropout. As with any linear regulator, PSRR and transient response are degraded as (V_{IN} – V_{OUT}) approaches dropout. This effect is shown in the Typical Characteristics section. Refer to application report SLVA207, *Understanding LDO Dropout*, available for download from www.ti.com.

TRANSIENT RESPONSE

As with any regulator, increasing the size of the output capacitor reduces over/undershoot magnitude but increases duration of the transient response. For more information, see Figure 19.

ACTIVE V_{OUT} PULL-DOWN

In the TPS782 series, the active pull-down discharges V_{OUT} when the device is off. However, the input voltage must be greater than 2.2V for the active pull-down to work.

MINIMUM LOAD

The TPS782 series are stable with no output load. Traditional PMOS LDO regulators suffer from lower loop gain at very light output loads. The TPS782 employs an innovative, low-current circuit under very light or no-load conditions, resulting in improved output voltage regulation performance down to zero output current. See Figure 19 for the load transient response.

THERMAL INFORMATION

THERMAL PROTECTION

Thermal protection disables the device output when the junction temperature rises to approximately +160°C, allowing the device to cool. Once the junction temperature cools to approximately +140°C, the output circuitry is enabled. Depending on power dissipation, thermal resistance, and ambient temperature, the thermal protection circuit may cycle on and off again. This cycling limits the dissipation of the regulator, protecting it from damage as a result of overheating.

Any tendency to activate the thermal protection circuit indicates excessive power dissipation or an inadequate heatsink. For reliable operation, junction temperature should be limited to +125°C maximum. To estimate the margin of safety in a complete design (including heatsink), increase the ambient temperature until the thermal protection is triggered; use worst-case loads and signal conditions. For good reliability, thermal protection should trigger at least +35°C above the maximum expected ambient condition of your particular application. This configuration produces a worst-case junction temperature of +125°C at the highest expected ambient temperature and worst-case load.

The internal protection circuitry of the TPS782 series has been designed to protect against overload conditions. However, it is not intended to replace proper heatsinking. Continuously running the TPS782 series into thermal shutdown degrades device reliability.

POWER DISSIPATION

The ability to remove heat from the die is different for each package type, presenting different considerations in the PCB layout. The PCB area around the device that is free of other components moves the heat from the device to the ambient air. Performance data for JEDEC low- and high-K boards are given in the Dissipation Ratings table. Using heavier copper increases the effectiveness in removing heat from the device. The addition of plated through-holes to heat-dissipating layers also improves the heatsink effectiveness. Power dissipation depends on input voltage and load conditions. Power dissipation (P_D) is equal to the product of the output current times the voltage drop across the output pass element (V_{IN} to V_{OUT}), as shown in Equation 2:

$$P_D = (V_{IN} - V_{OUT}) \times I_{OUT} \quad (2)$$

PACKAGE MOUNTING

Solder pad footprint recommendations for the TPS782 series are available from the Texas Instruments web site at www.ti.com through the TPS782 series product folders.

REVISION HISTORY

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision B (May 2010) to Revision C — **Page**

- Changed I_Q value in *Description* section from 1µA to 500nA 1

Changes from Revision A (September 2008) to Revision B — **Page**

- Updated title of data sheet 1
- Changed first bullet of *Features* list 1
- Changed *ground pin current, I_{OUT} = 0mA* typical specification from 1.0µA to 0.42µA 3
- Added Figure 6 5

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
TPS78218DDCR	PREVIEW	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	SJY	
TPS78218DDCT	PREVIEW	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	SJY	
TPS78218DRVR	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	SAF	Samples
TPS78218DRVT	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	SAF	Samples
TPS78222DRVR	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	RAR	Samples
TPS78222DRVT	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	RAR	Samples
TPS78223DDCR	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	NXM	Samples
TPS78223DDCT	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	NXM	Samples
TPS78225DDCR	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVD	Samples
TPS78225DDCRG4	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVD	Samples
TPS78225DDCT	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVD	Samples
TPS78225DDCTG4	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVD	Samples
TPS78225DRVR	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVD	Samples
TPS78225DRVT	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVD	Samples
TPS78225DRVTG4	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVD	Samples
TPS78227DDCR	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVE	Samples
TPS78227DDCRG4	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVE	Samples

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
TPS78227DDCT	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVE	Samples
TPS78227DDCTG4	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVE	Samples
TPS78227DRVR	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVE	Samples
TPS78227DRVT	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVE	Samples
TPS78227DRVTG4	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVE	Samples
TPS78228DDCR	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVF	Samples
TPS78228DDCRG4	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVF	Samples
TPS78228DDCT	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVF	Samples
TPS78228DDCTG4	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	CVF	Samples
TPS78228DRVR	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVF	Samples
TPS78228DRVRG4	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVF	Samples
TPS78228DRVT	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	CVF	Samples
TPS78230DDCR	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	OCK	Samples
TPS78230DDCT	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	OCK	Samples
TPS78230DRVR	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	ODE	Samples
TPS78230DRVT	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	ODE	Samples
TPS78233DDCR	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	OAH	Samples
TPS78233DDCT	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	OAH	Samples

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
TPS78236DDCR	ACTIVE	SOT	DDC	5	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	SCE	Samples
TPS78236DDCT	ACTIVE	SOT	DDC	5	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-40 to 125	SCE	Samples
TPS78236DRVR	ACTIVE	SON	DRV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	SCE	Samples
TPS78236DRVT	ACTIVE	SON	DRV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	-40 to 125	SCE	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) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device 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 Device Marking for that device.

(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

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.

OTHER QUALIFIED VERSIONS OF TPS78225, TPS78227, TPS78228, TPS78230 :

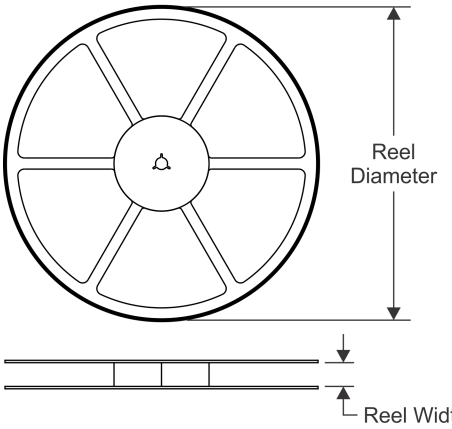
- Automotive: TPS78225-Q1, TPS78227-Q1, TPS78228-Q1, TPS78230-Q1

NOTE: Qualified Version Definitions:

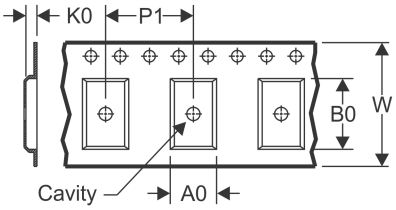
- Automotive - Q100 devices qualified for high-reliability automotive applications targeting zero defects

TAPE AND REEL INFORMATION

REEL DIMENSIONS

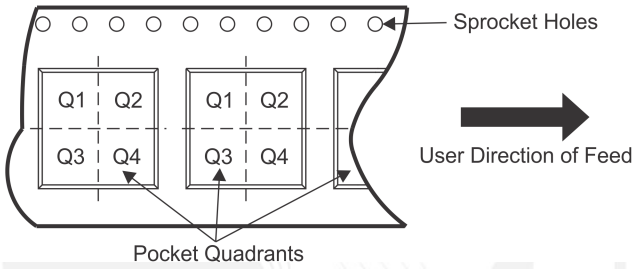


TAPE DIMENSIONS



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

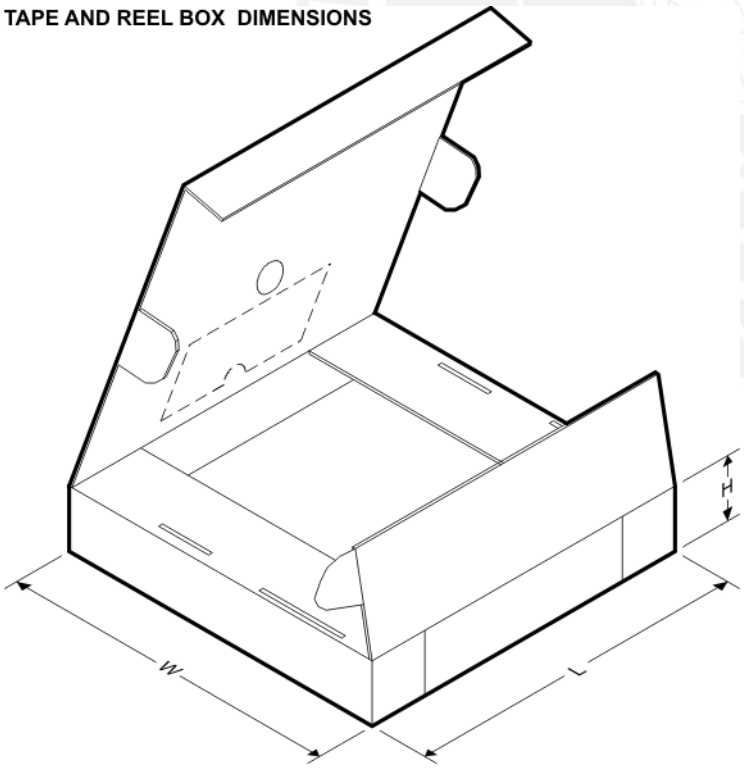


*All dimensions are nominal

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
TPS78218DRVR	SON	DRV	6	3000	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78218DRVR	SON	DRV	6	3000	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78218DRVT	SON	DRV	6	250	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78218DRVT	SON	DRV	6	250	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78222DRVR	SON	DRV	6	3000	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78222DRVR	SON	DRV	6	3000	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78222DRVT	SON	DRV	6	250	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78222DRVT	SON	DRV	6	250	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78223DDCR	SOT	DDC	5	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78223DDCT	SOT	DDC	5	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78225DDCR	SOT	DDC	5	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78225DDCT	SOT	DDC	5	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78225DRVR	SON	DRV	6	3000	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78225DRVR	SON	DRV	6	3000	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78225DRVT	SON	DRV	6	250	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78225DRVT	SON	DRV	6	250	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78227DDCR	SOT	DDC	5	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78227DDCT	SOT	DDC	5	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3

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
TPS78227DRVR	SON	DRV	6	3000	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78227DRVR	SON	DRV	6	3000	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78227DRVT	SON	DRV	6	250	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78227DRVT	SON	DRV	6	250	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78228DDCR	SOT	DDC	5	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78228DDCT	SOT	DDC	5	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78228DRVR	SON	DRV	6	3000	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78228DRVT	SON	DRV	6	250	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78230DDCR	SOT	DDC	5	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78230DDCT	SOT	DDC	5	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78230DRVR	SON	DRV	6	3000	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78230DRVR	SON	DRV	6	3000	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78230DRVT	SON	DRV	6	250	179.0	8.4	2.2	2.2	1.2	4.0	8.0	Q2
TPS78230DRVT	SON	DRV	6	250	180.0	8.4	2.25	2.25	1.0	4.0	8.0	Q2
TPS78233DDCR	SOT	DDC	5	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78233DDCT	SOT	DDC	5	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78236DDCR	SOT	DDC	5	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
TPS78236DDCT	SOT	DDC	5	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3

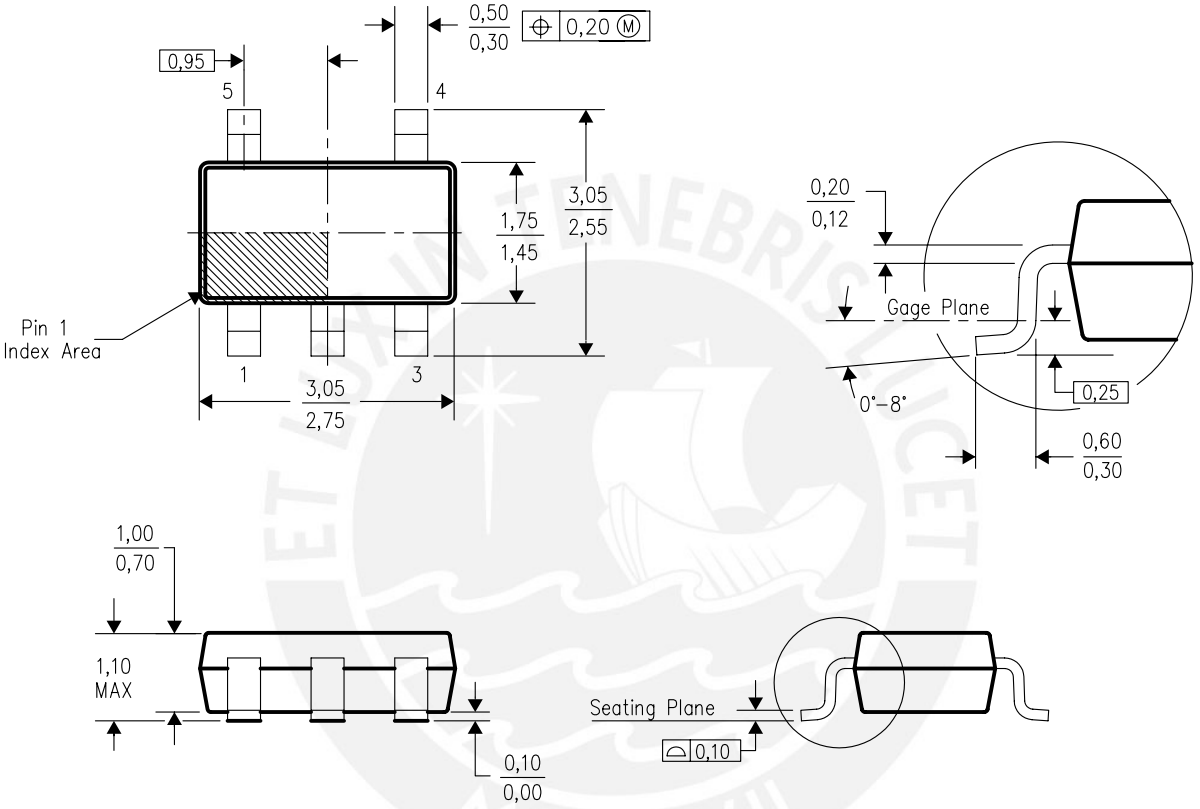
TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
TPS78218DRVR	SON	DRV	6	3000	203.0	203.0	35.0
TPS78218DRVR	SON	DRV	6	3000	205.0	200.0	33.0
TPS78218DRVT	SON	DRV	6	250	203.0	203.0	35.0
TPS78218DRVT	SON	DRV	6	250	205.0	200.0	33.0
TPS78222DRVR	SON	DRV	6	3000	205.0	200.0	33.0
TPS78222DRVR	SON	DRV	6	3000	203.0	203.0	35.0
TPS78222DRVT	SON	DRV	6	250	205.0	200.0	33.0
TPS78222DRVT	SON	DRV	6	250	203.0	203.0	35.0
TPS78223DDCR	SOT	DDC	5	3000	195.0	200.0	45.0
TPS78223DDCT	SOT	DDC	5	250	195.0	200.0	45.0
TPS78225DDCR	SOT	DDC	5	3000	195.0	200.0	45.0
TPS78225DDCT	SOT	DDC	5	250	195.0	200.0	45.0
TPS78225DRVR	SON	DRV	6	3000	203.0	203.0	35.0
TPS78225DRVR	SON	DRV	6	3000	205.0	200.0	33.0
TPS78225DRVT	SON	DRV	6	250	205.0	200.0	33.0
TPS78225DRVT	SON	DRV	6	250	203.0	203.0	35.0
TPS78227DDCR	SOT	DDC	5	3000	195.0	200.0	45.0
TPS78227DDCT	SOT	DDC	5	250	195.0	200.0	45.0
TPS78227DRVR	SON	DRV	6	3000	203.0	203.0	35.0
TPS78227DRVR	SON	DRV	6	3000	205.0	200.0	33.0
TPS78227DRVT	SON	DRV	6	250	203.0	203.0	35.0
TPS78227DRVT	SON	DRV	6	250	205.0	200.0	33.0
TPS78228DDCR	SOT	DDC	5	3000	195.0	200.0	45.0
TPS78228DDCT	SOT	DDC	5	250	195.0	200.0	45.0
TPS78228DRVR	SON	DRV	6	3000	203.0	203.0	35.0
TPS78228DRVT	SON	DRV	6	250	203.0	203.0	35.0
TPS78230DDCR	SOT	DDC	5	3000	195.0	200.0	45.0
TPS78230DDCT	SOT	DDC	5	250	195.0	200.0	45.0
TPS78230DRVR	SON	DRV	6	3000	205.0	200.0	33.0
TPS78230DRVR	SON	DRV	6	3000	203.0	203.0	35.0
TPS78230DRVT	SON	DRV	6	250	203.0	203.0	35.0
TPS78230DRVT	SON	DRV	6	250	205.0	200.0	33.0
TPS78233DDCR	SOT	DDC	5	3000	203.0	203.0	35.0
TPS78233DDCT	SOT	DDC	5	250	203.0	203.0	35.0
TPS78236DDCR	SOT	DDC	5	3000	195.0	200.0	45.0
TPS78236DDCT	SOT	DDC	5	250	195.0	200.0	45.0

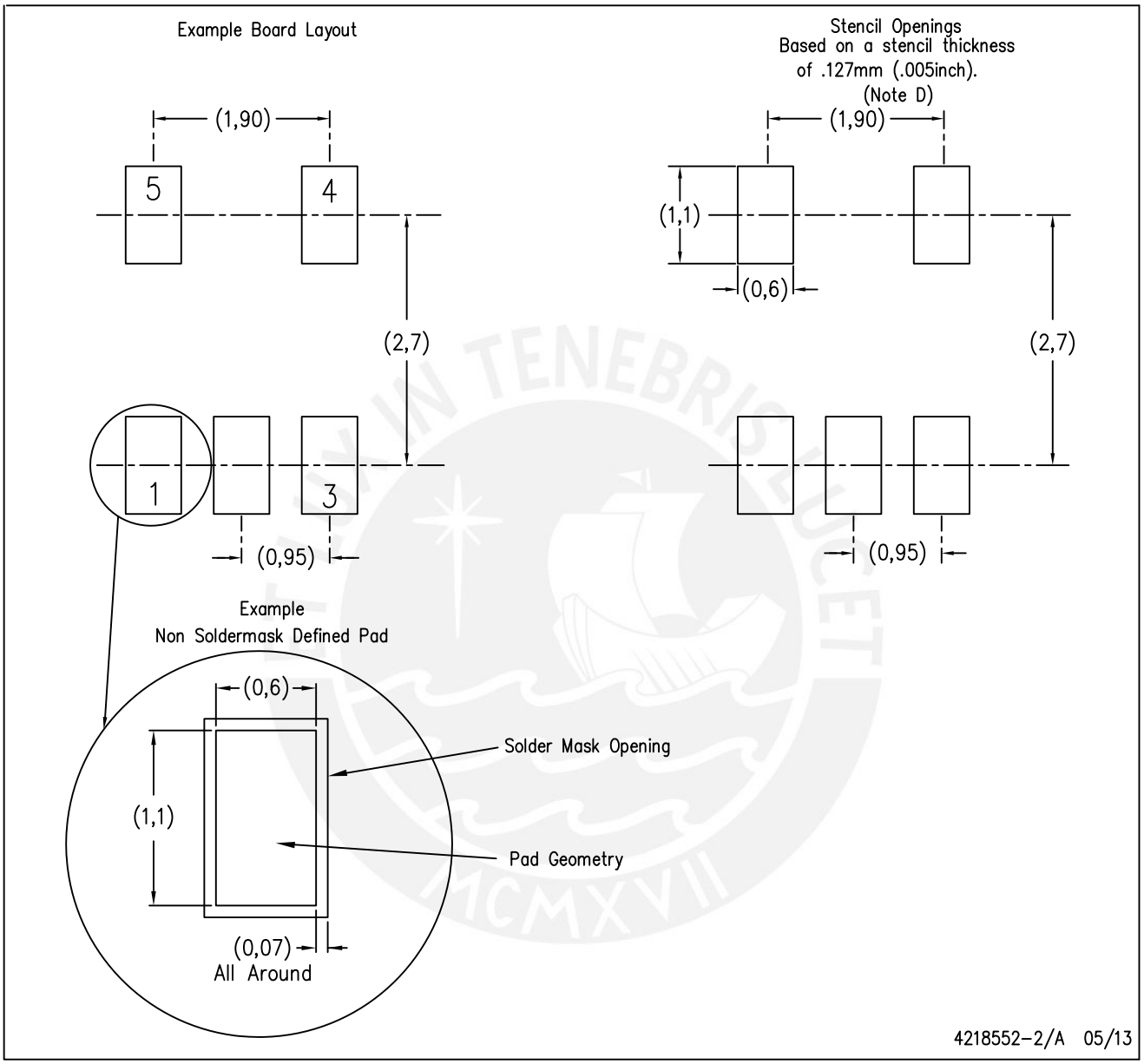
DDC (R–PDSO–G5) PLASTIC SMALL–OUTLINE



4204403–2/E 06/05

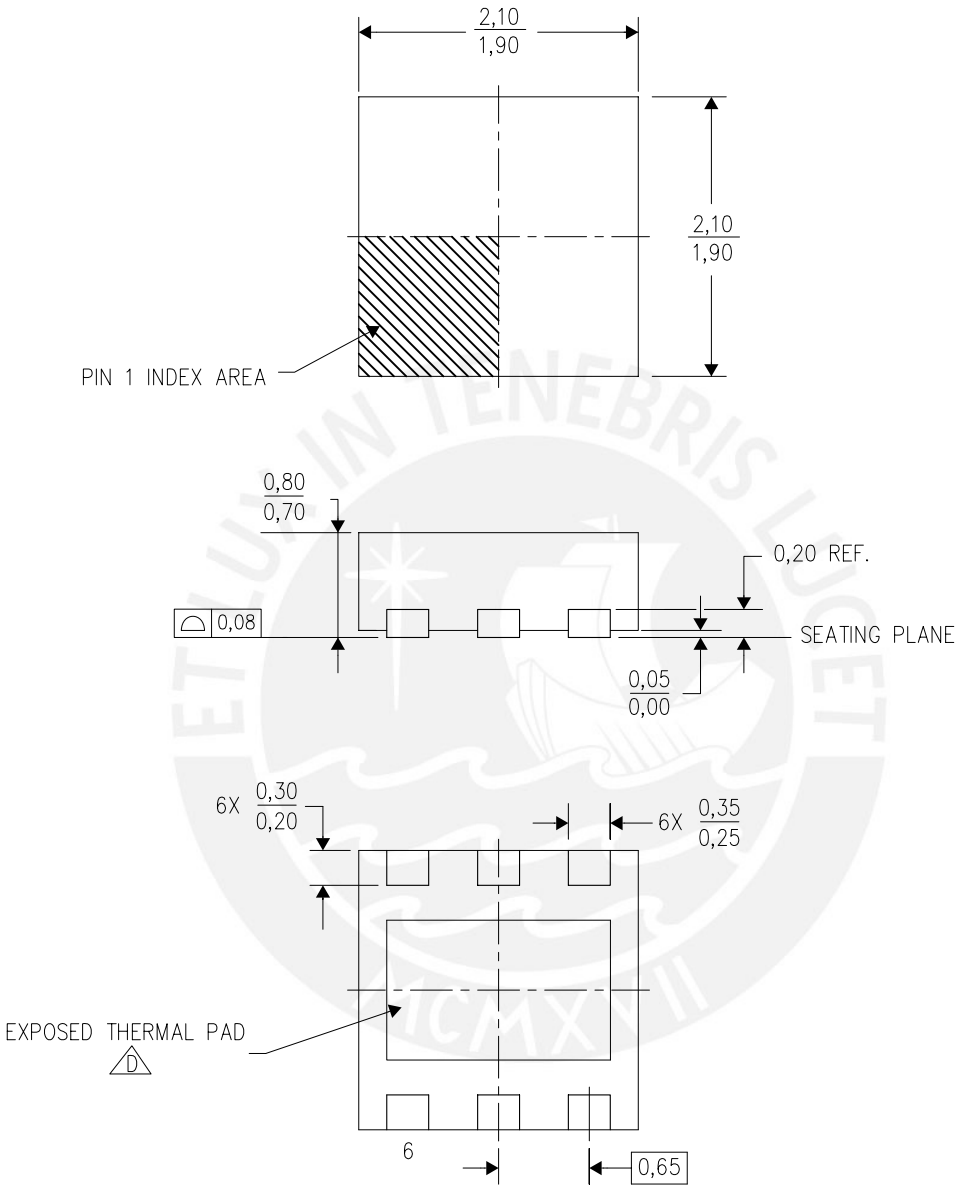
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion.
D. Falls within JEDEC MO–193 variation AB (5 pin).

DDC (R–PDSO–G5) PLASTIC SMALL OUTLINE



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Publication IPC–7351 is recommended for alternate designs.
D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC–7525 for other stencil recommendations.

DRV (S–PWSON–N6) PLASTIC SMALL OUTLINE NO–LEAD



4206925/E 10/10

NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. Small Outline No–Lead (SON) package configuration.
D. The package thermal pad must be soldered to the board for thermal and mechanical performance. See the Product Data Sheet for details regarding the exposed thermal pad dimensions.

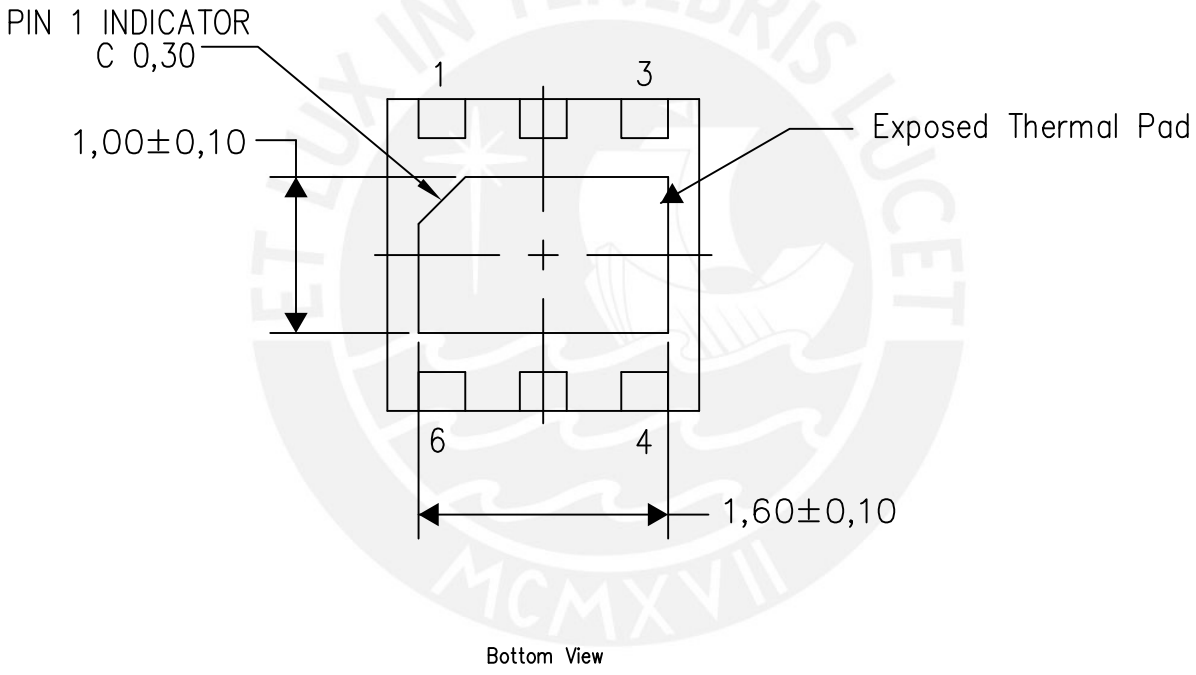
DRV (S–PWSON–N6) PLASTIC SMALL OUTLINE NO–LEAD

THERMAL INFORMATION

This package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB). After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For information on the Quad Flatpack No–Lead (QFN) package and its advantages, refer to Application Report, QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271. This document is available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.



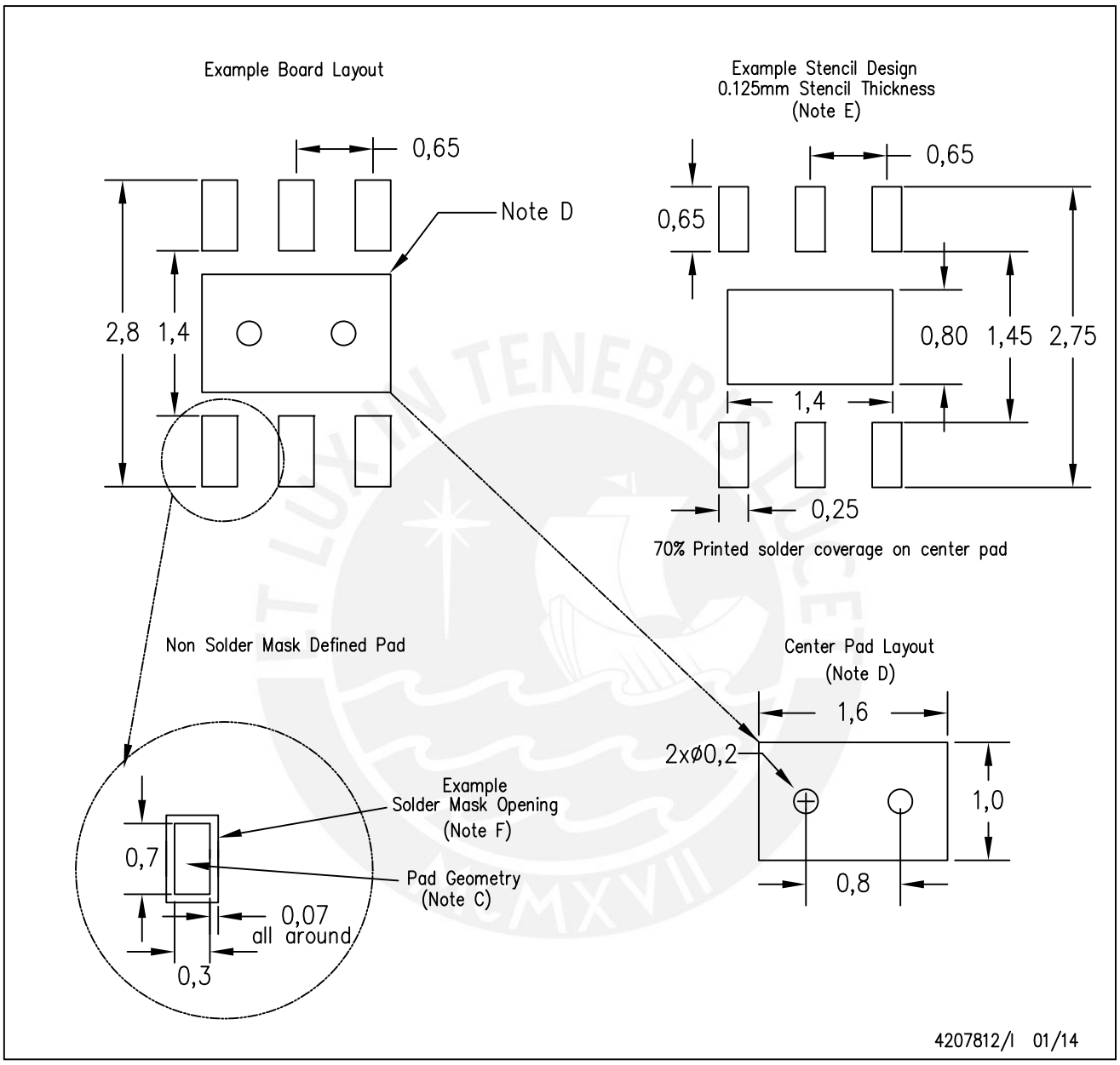
Bottom View

Exposed Thermal Pad Dimensions

4206926/0 01/14

NOTE: All linear dimensions are in millimeters

DRV (S–PWSON–N6) PLASTIC SMALL OUTLINE NO–LEAD



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Publication IPC–7351 is recommended for alternate designs.
D. This package is designed to be soldered to a thermal pad on the board. Refer to Application Note, QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271, and also the Product Data Sheets for specific thermal information, via requirements, and recommended board layout. These documents are available at www.ti.com <http://www.ti.com>.
E. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC 7525 for stencil design considerations.
F. Customers should contact their board fabrication site for solder mask tolerances.

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

Features

- High Performance, Low Power 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 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - 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
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I^2C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated 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
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (Including 32 kHz RTC)

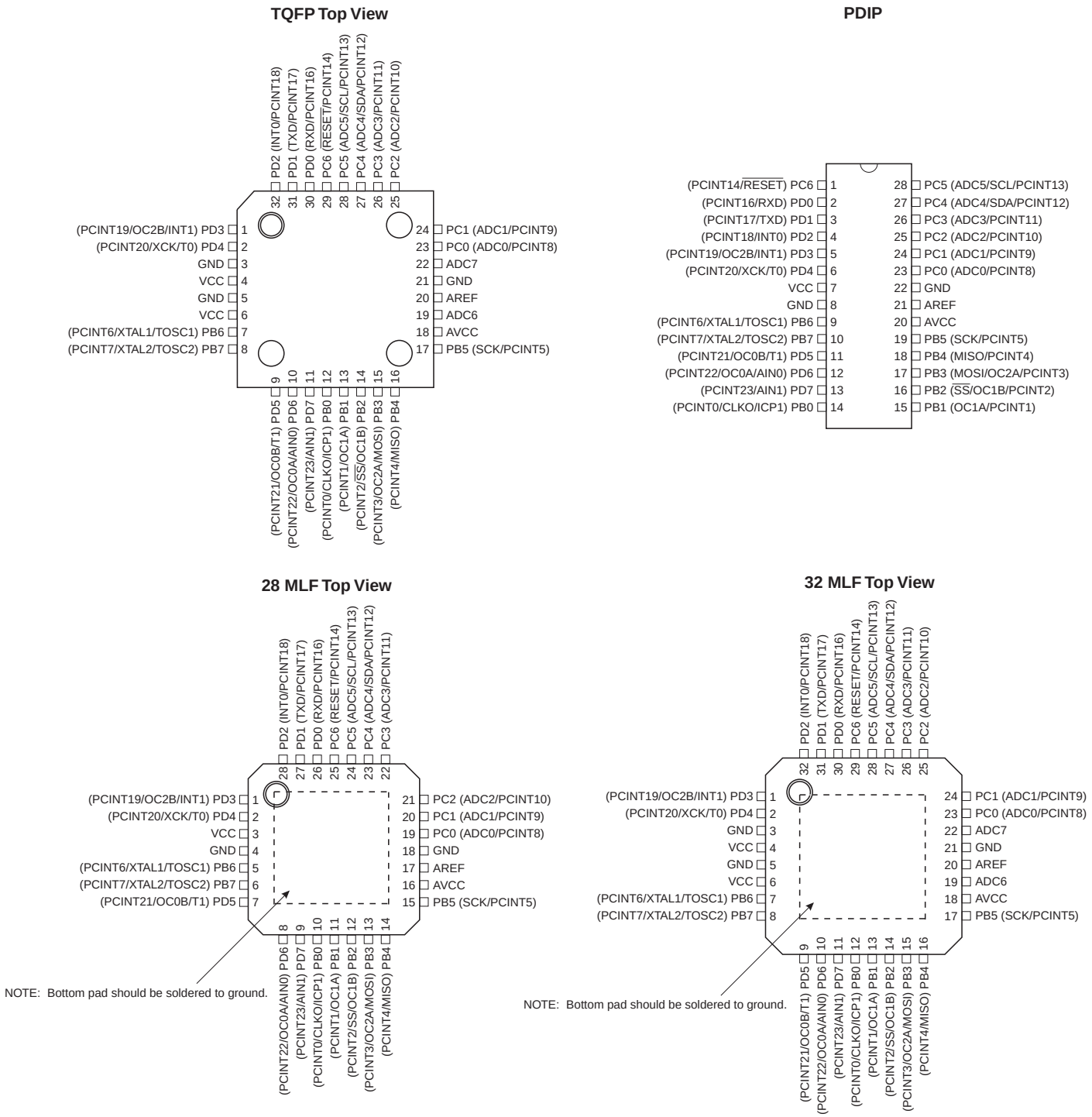
8-bit AVR® Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

Rev. 8161D–AVR–10/09

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

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.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ”Alternate Functions of Port B” on page 82 and ”System Clock and Clock Options” on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 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.

1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a 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 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ”Alternate Functions of Port C” on page 85.

1.1.6 Port D (PD7:0)

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.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 88.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. 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. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

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

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

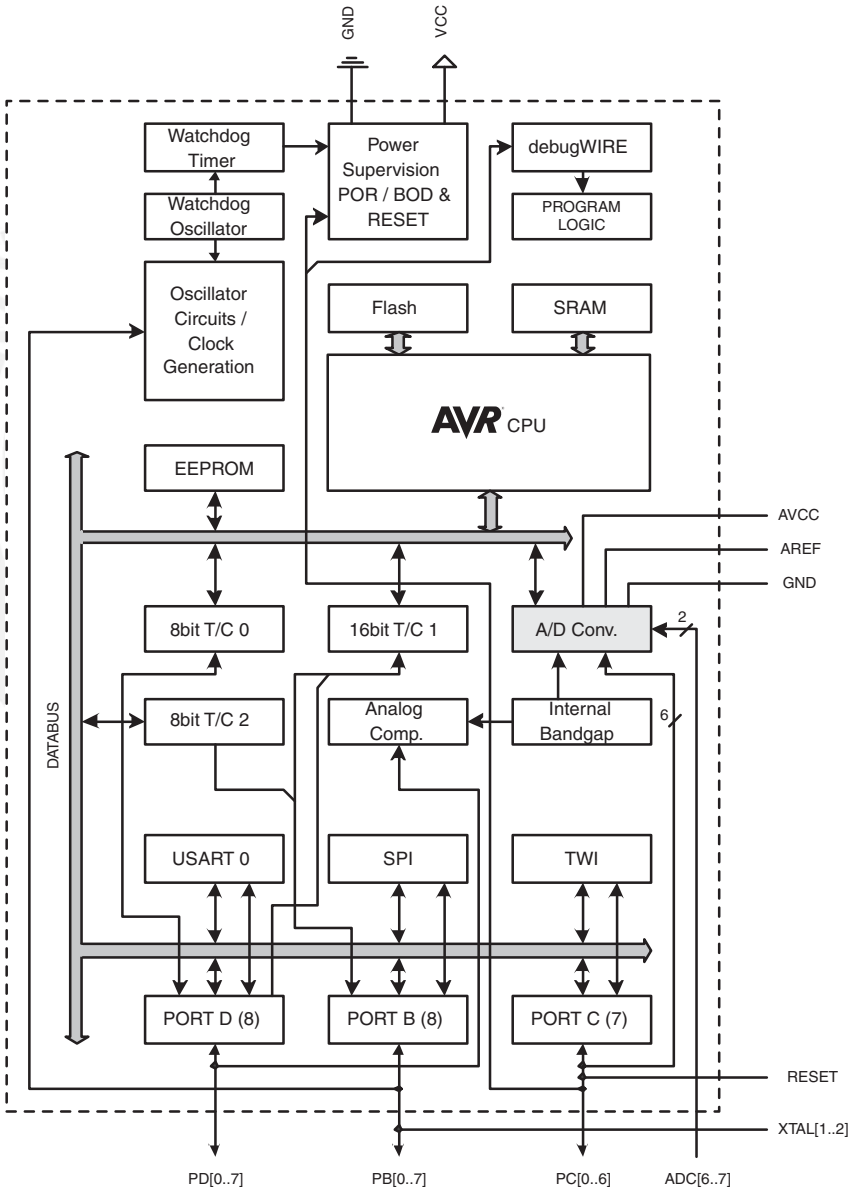
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Overview

The ATmega48PA/88PA/168PA/328P 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 ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. 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 ATmega48PA/88PA/168PA/328P provides the following features: 4K/8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, 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 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.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile 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 ATmega48PA/88PA/168PA/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48PA/88PA/168PA/328P 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.

2.2 Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P

The ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

ATmega88PA, ATmega168PA and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48PA, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

3. Resources

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

4. 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.

5. 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.

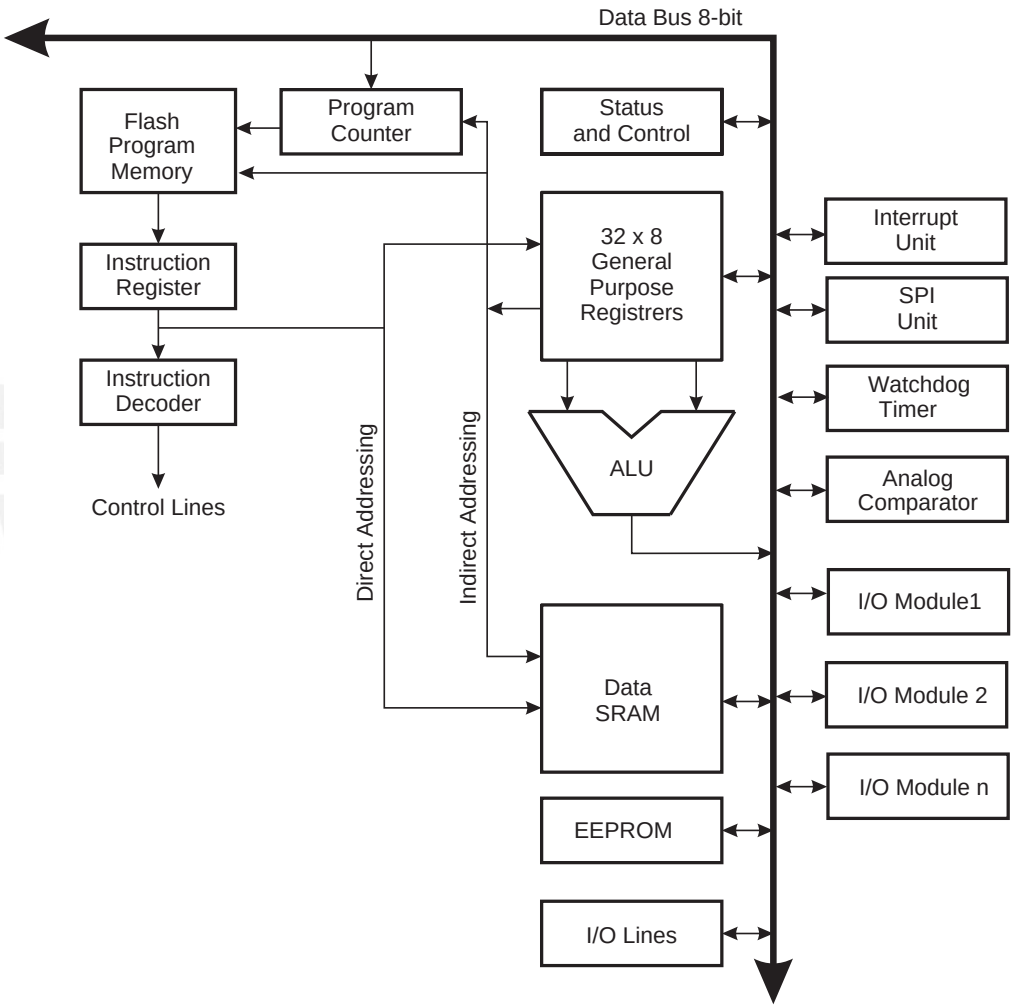
For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

6. AVR CPU Core

6.1 Overview

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.

Figure 6-1. Block Diagram of the AVR 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 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typ-

ical 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-, Y-, 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- 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, 0x20 - 0x5F. In addition, the ATmega48PA/88PA/168PA/328P has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

6.2 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.

6.3 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.

6.3.1 SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	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.

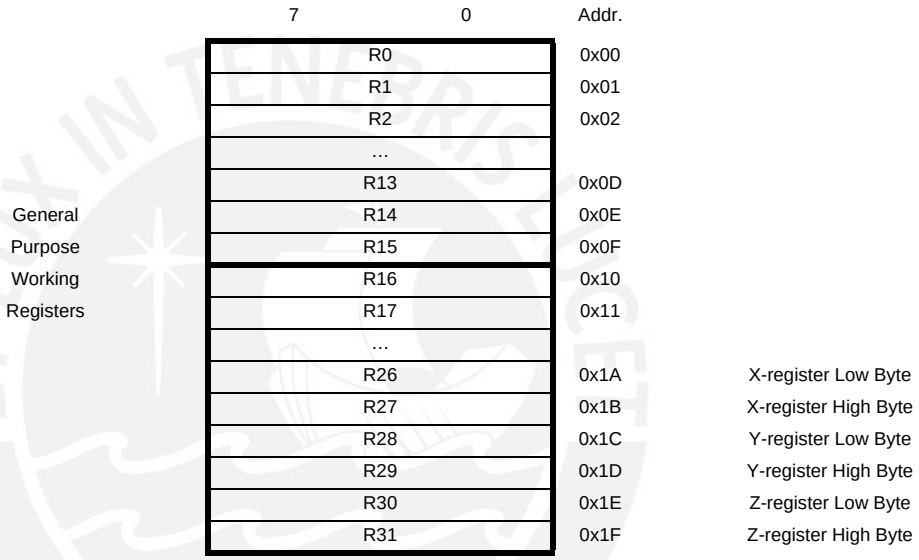
6.4 General Purpose Register File

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 6-2 shows the structure of the 32 general purpose working registers in the CPU.

Figure 6-2. AVR CPU General Purpose Working Registers



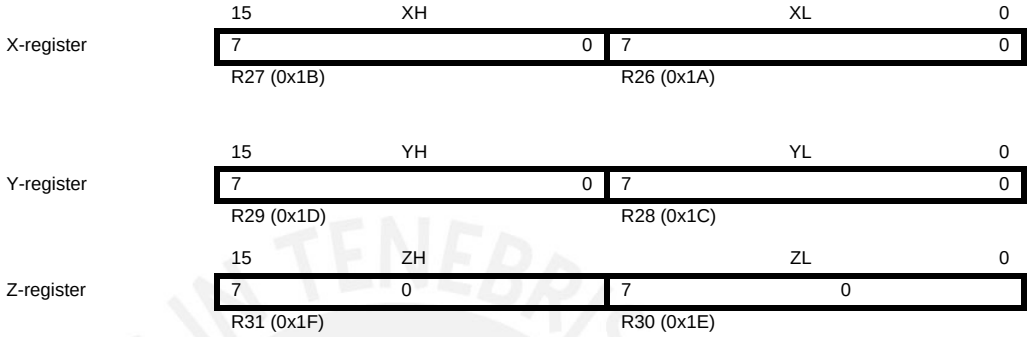
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 6-2, 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.

6.4.1 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 6-3.

Figure 6-3. The X-, Y-, and Z-registers



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).

6.5 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. Note that the Stack is implemented as growing from higher to lower memory locations. The Stack Pointer Register always points to the top of the Stack. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.

The Stack in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. Initial Stack Pointer value equals the last address of the internal SRAM and the Stack Pointer must be set to point above start of the SRAM, see Table 7-3 on page 18.

See Table 6-1 for Stack Pointer details.

Table 6-1. Stack Pointer instructions

Instruction	Stack pointer	Description
PUSH	Decremented by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decremented by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Incremented by 1	Data is popped from the stack
RET RETI	Incremented by 2	Return address is popped from the stack with return from subroutine or return from interrupt

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.

6.5.1 SPH and SPL – Stack Pointer High and Stack Pointer Low Register

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
0x3D (0x5D)	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	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	
	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	

6.6 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 6-4 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 6-4. The Parallel Instruction Fetches and Instruction Executions

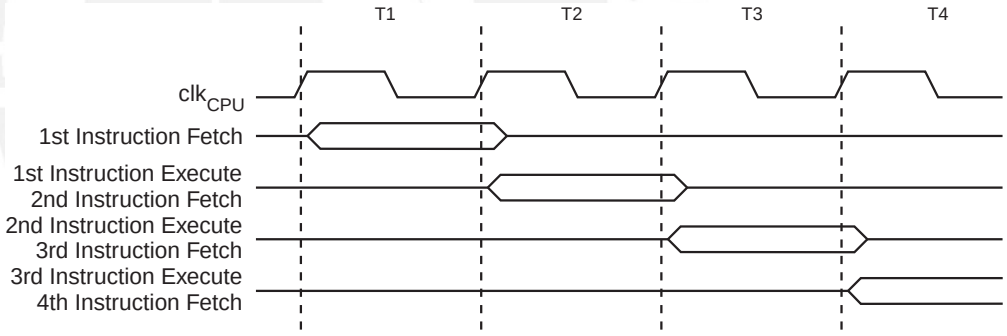
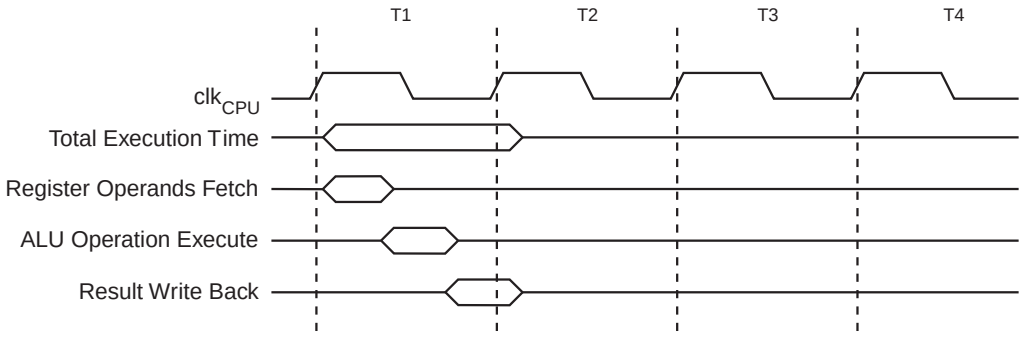


Figure 6-5 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 6-5. Single Cycle ALU Operation



6.7 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 294 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 57. 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 MCU Control Register (MCUCR). Refer to ”Interrupts” on page 57 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, ATmega88PA, ATmega168PA and ATmega328P” on page 277.

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, EEMPE   ; start EEPROM write
sbi EECR, EEPE
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<<EEMPE); /* start EEPROM write */
EECR |= (1<<EEPE);
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

```
__enable_interrupt(); /* set Global Interrupt Enable */
__sleep(); /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
```

6.7.1 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.

7. AVR Memories

7.1 Overview

This section describes the different memories in the ATmega48PA/88PA/168PA/328P. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega48PA/88PA/168PA/328P features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

7.2 In-System Reprogrammable Flash Program Memory

The ATmega48PA/88PA/168PA/328P contains 4/8/16/32K bytes 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 2/4/8/16K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Loader Section and Application Program Section in ATmega88PA and ATmega168PA. See SELFPRGEN description in section ”SPMCSR – Store Program Memory Control and Status Register” on page 292 for more details.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega48PA/88PA/168PA/328P Program Counter (PC) is 11/12/13/14 bits wide, thus addressing the 2/4/8/16K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in ”Self-Programming the Flash, ATmega48PA” on page 269 and ”Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P” on page 277. ”Memory Programming” on page 294 contains a detailed description on Flash Programming in SPI- or Parallel Programming mode.

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 7-1. Program Memory Map ATmega48PA

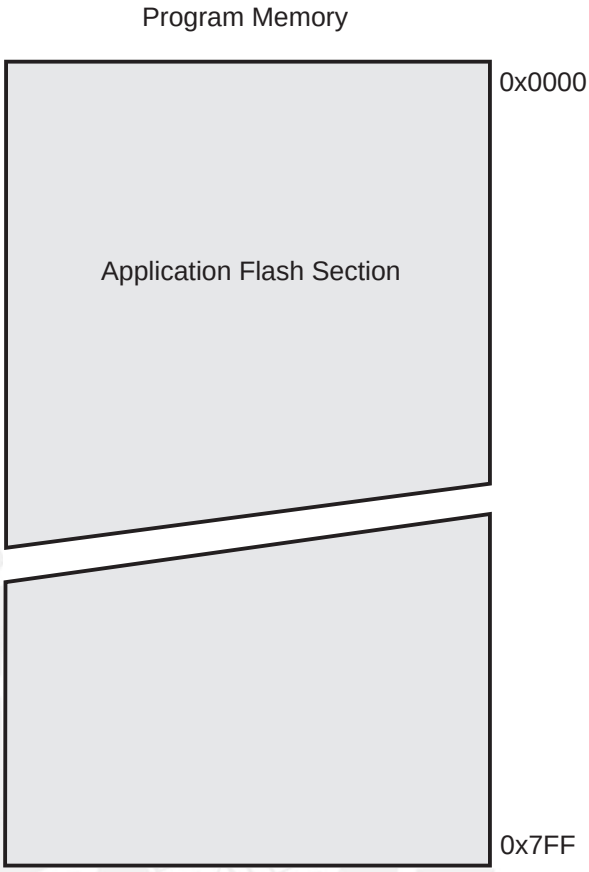
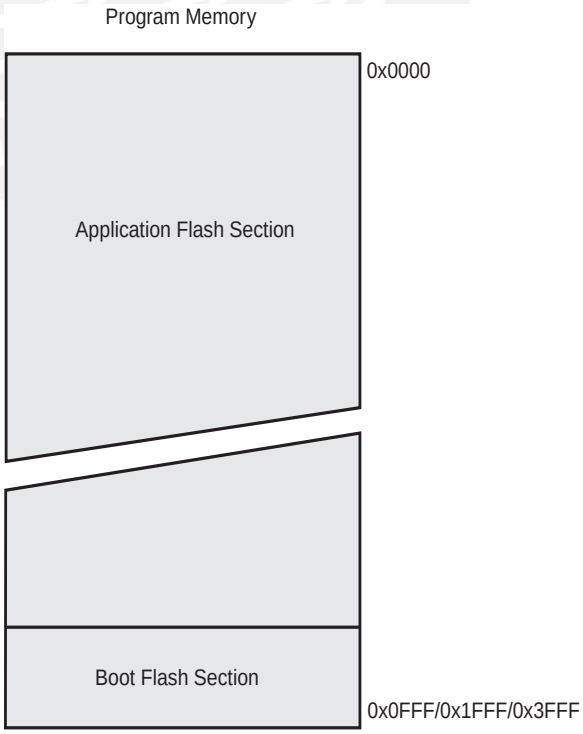


Figure 7-2. Program Memory Map ATmega88PA, ATmega168PA and ATmega328P



7.3 SRAM Data Memory

Figure 7-3 shows how the ATmega48PA/88PA/168PA/328P SRAM Memory is organized.

The ATmega48PA/88PA/168PA/328P is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in the Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

The lower 768/1280/1280/2303 data memory locations address both the Register File, the I/O memory, Extended I/O memory, and the internal data SRAM. The first 32 locations address the Register File, the next 64 location the standard I/O memory, then 160 locations of Extended I/O memory, and the next 512/1024/1024/2048 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.

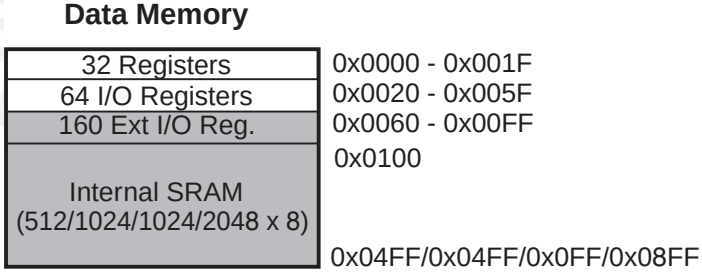
The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- 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, 160 Extended I/O Registers, and the 512/1024/1024/2048 bytes of internal data SRAM in the ATmega48PA/88PA/168PA/328P are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 11.

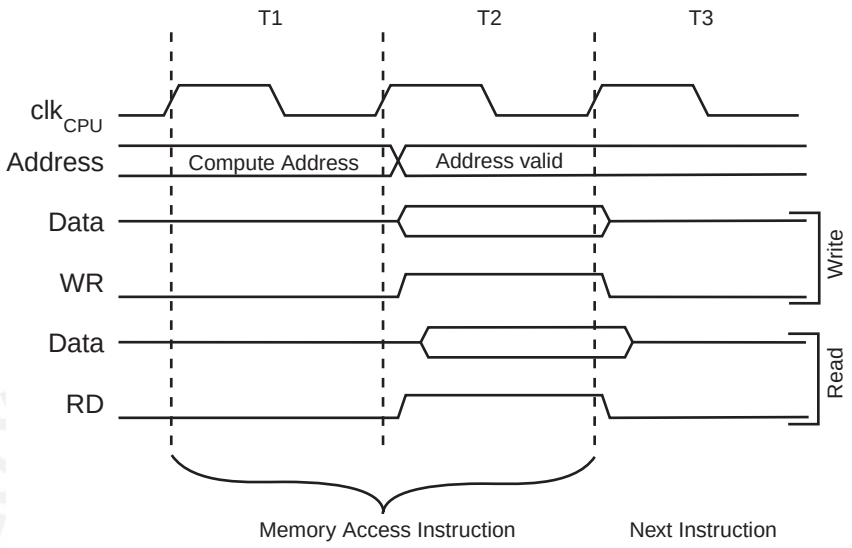
Figure 7-3. Data Memory Map



7.3.1 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 7-4.

Figure 7-4. On-chip Data SRAM Access Cycles



7.4 EEPROM Data Memory

The ATmega48PA/88PA/168PA/328P contains 256/512/512/1K 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.

”Memory Programming” on page 294 contains a detailed description on EEPROM Programming in SPI or Parallel Programming mode.

7.4.1 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 7-2. 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 20 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.

7.4.2 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.

7.5 I/O Memory

The I/O space definition of the ATmega48PA/88PA/168PA/328P is shown in ”Register Summary” on page 423.

All ATmega48PA/88PA/168PA/328P I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F 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 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48PA/88PA/168PA/328P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

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, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and peripherals control registers are explained in later sections.

7.5.1 General Purpose I/O Registers

The ATmega48PA/88PA/168PA/328P contains three General Purpose I/O Registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags. General Purpose I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

7.6 Register Description

7.6.1 EEARH and EEARL – The EEPROM Address Register

Bit	15	14	13	12	11	10	9	8	
0x22 (0x42)	–	–	–	–	–	–	–	EEAR8	EEARH
0x21 (0x41)	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 ATmega48PA/88PA/168PA/328P 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 256/512/512/1K bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 255/511/511/1023. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

EEAR8 is an unused bit in ATmega48PA and must always be written to zero.

7.6.2 EEDR – The EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
0x20 (0x40)	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.

7.6.3 EECR – The EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
0x1F (0x3F)	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	X	X	0	0	X	0	

- **Bits 7..6 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bits 5, 4 – EEPM1 and EEPM0: EEPROM Programming Mode Bits**

The EEPROM Programming mode bit setting defines which programming action that will be triggered when writing EEPE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the Erase and Write operations in two different operations. The Programming times for the different modes are shown in Table 7-1. While EEPE

is set, any write to EEPMn will be ignored. During reset, the EEPMn bits will be reset to 0b00 unless the EEPROM is busy programming.

Table 7-1. EEPROM Mode Bits

EEPM1	EEPM0	Programming Time	Operation
0	0	3.4 ms	Erase and Write in one operation (Atomic Operation)
0	1	1.8 ms	Erase Only
1	0	1.8 ms	Write Only
1	1	–	Reserved for future use

- **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 EEPE is cleared. The interrupt will not be generated during EEPROM write or SPM.

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

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

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

The EEPROM Write Enable Signal EEPE is the write strobe to the EEPROM. When address and data are correctly set up, the EEPE bit must be written to one to write the value into the EEPROM. The EEMPE bit must be written to one before a logical one is written to EEPE, 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 EEPE becomes zero.
2. Wait until SELFPRGEN in SPMCSR 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 EEMPE bit while writing a zero to EEPE in EECR.
6. Within four clock cycles after setting EEMPE, write a logical one to EEPE.

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, ATmega88PA, ATmega168PA and ATmega328P” on page 277 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 EEPE bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEPE 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 EEPE 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 7-2 lists the typical programming time for EEPROM access from the CPU.

Table 7-2. EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles	Typ Programming Time
EEPROM write (from CPU)	26,368	3.3 ms

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (e.g. 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,EEPE
  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 EEMPE
  sbi  EECR,EEMPE
  ; Start eeprom write by setting EEPE
  sbi  EECR,EEPE
  ret
```

C Code Example

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

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,EEPE
  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<<EEPE))
    ;
  /* Set up address register */
  EEAR = uiAddress;
  /* Start eeprom read by writing EERE */
  EECR |= (1<<EERE);
  /* Return data from Data Register */
  return EEDR;
}
```

7.6.4 GPIOR2 – General Purpose I/O Register 2

Bit	7	6	5	4	3	2	1	0	
0x2B (0x4B)	MSB							LSB	GPIOR2
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	

7.6.5 GPIOR1 – General Purpose I/O Register 1

Bit	7	6	5	4	3	2	1	0	
0x2A (0x4A)	MSB							LSB	GPIOR1
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	

7.6.6 GPIOR0 – General Purpose I/O Register 0

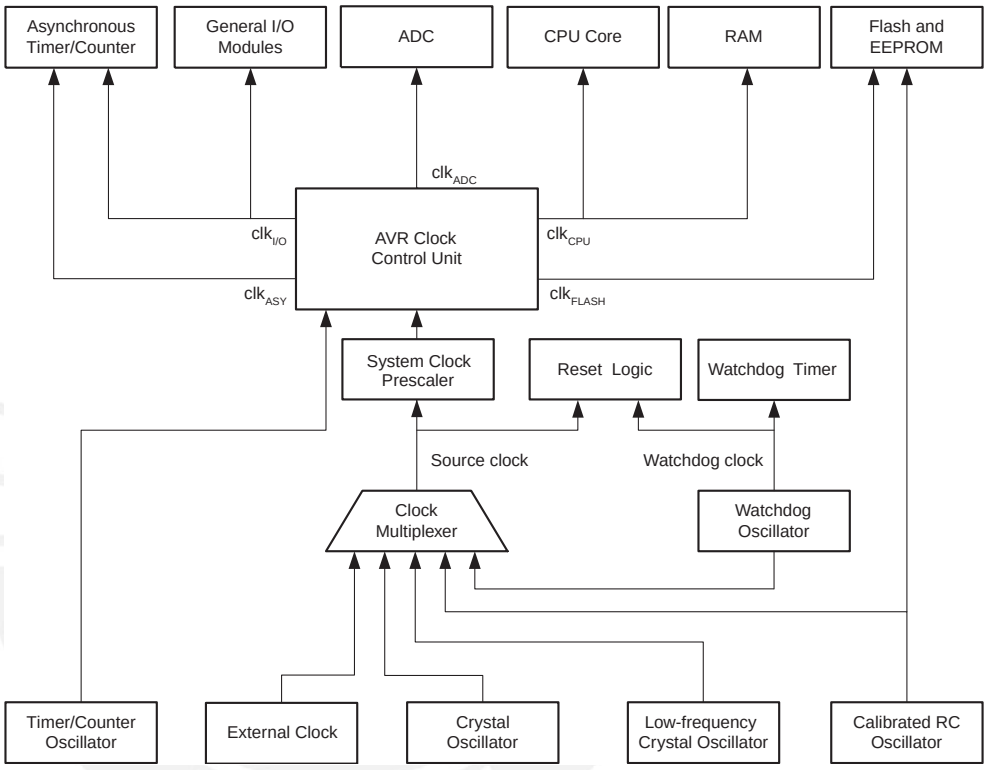
Bit	7	6	5	4	3	2	1	0	
0x1E (0x3E)	MSB							LSB	GPIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

8. System Clock and Clock Options

8.1 Clock Systems and their Distribution

Figure 8-1 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 39. The clock systems are detailed below.

Figure 8-1. Clock Distribution



8.1.1 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.

8.1.2 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 start condition detection in the USI module is carried out asynchronously when $clk_{I/O}$ is halted, TWI address recognition in all sleep modes.

8.1.3 Flash Clock – clk_{FLASH}

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

8.1.4 Asynchronous Timer Clock – clk_{ASY}

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external clock or 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.

8.1.5 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.

8.2 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 8-1. Device Clocking Options Select(1)

Device Clocking Option	CKSEL3..0
Low Power Crystal Oscillator	1111 - 1000
Full Swing Crystal Oscillator	0111 - 0110
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128 kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External Clock	0000
Reserved	0001

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

8.2.1 Default Clock Source

The device is shipped with internal RC oscillator at 8.0MHz and with the fuse CKDIV8 programmed, resulting in 1.0MHz system clock. The startup time is set to maximum and time-out period enabled. (CKSEL = "0010", SUT = "10", CKDIV8 = "0"). The default setting ensures that all users can make their desired clock source setting using any available programming interface.

8.2.2 Clock Startup Sequence

Any clock source needs a sufficient V_{CC} to start oscillating and a minimum number of oscillating cycles before it can be considered stable.

To ensure sufficient V_{CC}, the device issues an internal reset with a time-out delay (t_{TOUT}) after the device reset is released by all other reset sources. ”System Control and Reset” on page 46 describes the start conditions for the internal reset. The delay (t_{TOUT}) is timed from the Watchdog Oscillator and the number of cycles in the delay is set by the SUTx and CKSELx fuse bits. The

selectable delays are shown in Table 8-2. The frequency of the Watchdog Oscillator is voltage dependent as shown in "Typical Characteristics" on page 326.

Table 8-2. Number of Watchdog Oscillator Cycles

Typ Time-out (V_{CC} = 5.0V)	Typ Time-out (V_{CC} = 3.0V)	Number of Cycles
0 ms	0 ms	0
4.1 ms	4.3 ms	512
65 ms	69 ms	8K (8,192)

Main purpose of the delay is to keep the AVR in reset until it is supplied with minimum V_{CC}. The delay will not monitor the actual voltage and it will be required to select a delay longer than the V_{CC} rise time. If this is not possible, an internal or external Brown-Out Detection circuit should be used. A BOD circuit will ensure sufficient V_{CC} before it releases the reset, and the time-out delay can be disabled. Disabling the time-out delay without utilizing a Brown-Out Detection circuit is not recommended.

The oscillator is required to oscillate for a minimum number of cycles before the clock is considered stable. An internal ripple counter monitors the oscillator output clock, and keeps the internal reset active for a given number of clock cycles. The reset is then released and the device will start to execute. The recommended oscillator start-up time is dependent on the clock type, and varies from 6 cycles for an externally applied clock to 32K cycles for a low frequency crystal.

The start-up sequence for the clock includes both the time-out delay and the start-up time when the device starts up from reset. When starting up from Power-save or Power-down mode, V_{CC} is assumed to be at a sufficient level and only the start-up time is included.

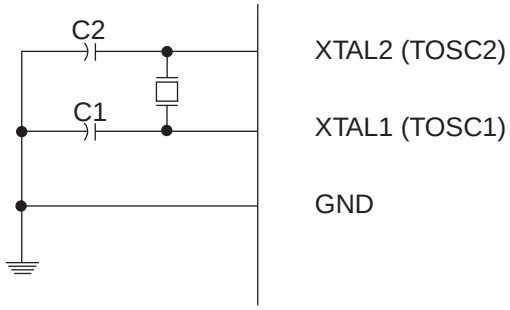
8.3 Low Power Crystal Oscillator

Pins 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 8-2 on page 29. Either a quartz crystal or a ceramic resonator may be used.

This Crystal Oscillator is a low power oscillator, with reduced voltage swing on the XTAL2 output. It gives the lowest power consumption, but is not capable of driving other clock inputs, and may be more susceptible to noise in noisy environments. In these cases, refer to the "Full Swing Crystal Oscillator" on page 30.

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 8-3 on page 29. For ceramic resonators, the capacitor values given by the manufacturer should be used.

Figure 8-2. Crystal Oscillator Connections



The Low Power 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 8-3 on page 29.

Table 8-3. Low Power Crystal Oscillator Operating Modes[3]

Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 (pF)	CKSEL3..1[1]
0.4 - 0.9	–	100[2]
0.9 - 3.0	12 - 22	101
3.0 - 8.0	12 - 22	110
8.0 - 16.0	12 - 22	111

Notes:
1. This is the recommanded CKSEL settings for the difference frenquency ranges.
2. This option should not be used with crystals, only with ceramic resonators.
3. If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8. It must be ensured that the resulting divided clock meets the frequency specification of the device.

The CKSEL0 Fuse together with the SUT1..0 Fuses select the start-up times as shown in Table 8-4.

Table 8-4. Start-up Times for the Low Power Crystal Oscillator Clock Selection

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	CKSEL0	SUT1..0
Ceramic resonator, fast rising power	258 CK	14CK + 4.1 ms[1]	0	00
Ceramic resonator, slowly rising power	258 CK	14CK + 65 ms[1]	0	01
Ceramic resonator, BOD enabled	1K CK	14CK[2]	0	10
Ceramic resonator, fast rising power	1K CK	14CK + 4.1 ms[2]	0	11
Ceramic resonator, slowly rising power	1K CK	14CK + 65 ms[2]	1	00

Table 8-4. Start-up Times for the Low Power Crystal Oscillator Clock Selection (Continued)

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	CKSEL0	SUT1..0
Crystal Oscillator, BOD enabled	16K CK	14CK	1	01
Crystal Oscillator, fast rising power	16K CK	14CK + 4.1 ms	1	10
Crystal Oscillator, slowly rising power	16K CK	14CK + 65 ms	1	11

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.

8.4 Full Swing Crystal Oscillator

Pins 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 8-2 on page 29. Either a quartz crystal or a ceramic resonator may be used.

This Crystal Oscillator is a full swing oscillator, with rail-to-rail swing on the XTAL2 output. This is useful for driving other clock inputs and in noisy environments. The current consumption is higher than the "Low Power Crystal Oscillator" on page 28. Note that the Full Swing Crystal Oscillator will only operate for V_{CC} = 2.7 - 5.5 volts.

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 8-6 on page 31. For ceramic resonators, the capacitor values given by the manufacturer should be used.

The operating mode is selected by the fuses CKSEL3..1 as shown in Table 8-5.

Table 8-5. Full Swing Crystal Oscillator operating modes

Frequency Range[1] (MHz)	Recommended Range for Capacitors C1 and C2 (pF)	CKSEL3..1
0.4 - 20	12 - 22	011

Notes:
1. If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8. It must be ensured that the resulting divided clock meets the frequency specification of the device.

Figure 8-3. Crystal Oscillator Connections

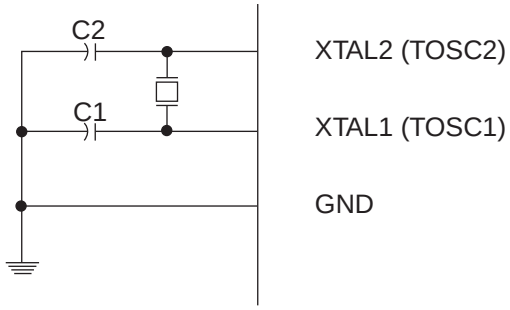


Table 8-6. Start-up Times for the Full Swing Crystal Oscillator Clock Selection

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	CKSEL0	SUT1..0
Ceramic resonator, fast rising power	258 CK	14CK + 4.1 ms[1]	0	00
Ceramic resonator, slowly rising power	258 CK	14CK + 65 ms[1]	0	01
Ceramic resonator, BOD enabled	1K CK	14CK[2]	0	10
Ceramic resonator, fast rising power	1K CK	14CK + 4.1 ms[2]	0	11
Ceramic resonator, slowly rising power	1K CK	14CK + 65 ms[2]	1	00
Crystal Oscillator, BOD enabled	16K CK	14CK	1	01
Crystal Oscillator, fast rising power	16K CK	14CK + 4.1 ms	1	10
Crystal Oscillator, slowly rising power	16K CK	14CK + 65 ms	1	11

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.

8.5 Low Frequency Crystal Oscillator

The Low-frequency Crystal Oscillator is optimized for use with a 32.768 kHz watch crystal. When selecting crystals, load capasitance and crystal's Equivalent Series Resistance, ESR must be taken into consideration. Both values are specified by the crystal vendor. ATmega48PA/88PA/168PA/328P oscillator is optimized for very low power consumption, and thus when selecting crystals, see Table 8-7 on page 32 for maximum ESR recommendations on 6.5 pF, 9.0 pF and 12.5 pF crystals

Table 8-7. Maximum ESR Recommendation for 32.768 kHz Crystal Oscillator

Crystal CL (pF)	Max ESR [kΩ](1)
6.5	75
9.0	65
12.5	30

Note: 1. Maximum ESR is typical value based on characterization

The Low-frequency Crystal Oscillator provides an internal load capacitance at each TOSC pin as specified in the Table 8-8 on page 32.

Table 8-8. Capacitance for Low-Frequency Crystal Oscillator

Device	32 kHz Osc. Type	Cap (Xtal1/Tosc1)	Cap (Xtal2/Tosc2)
ATmega48PA/88PA/168PA/328P	System Osc.	18 pF	8 pF
	Timer Osc.	18 pF	8 pF

The external capacitance (C) needed at each TOSC pin can be calculated by using:

$$C = 2 \cdot CL - C_s$$

where CL is the load capacitance for a 32.768 kHz crystal specified by the crystal vendor and C_S is the total stray capacitance for one TOSC pin.

Crystals specifying load capacitance (CL) higher than the ones given in the Table 8-8 on page 32, require external capacitors applied as described in Figure 8-2 on page 29.

The Low-frequency Crystal Oscillator must be selected by setting the CKSEL Fuses to "0110" or "0111", as shown in Table 8-10. Start-up times are determined by the SUT Fuses as shown in Table 8-9.

Table 8-9. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

SUT1..0	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	4 CK	Fast rising power or BOD enabled
01	4 CK + 4.1 ms	Slowly rising power
10	4 CK + 65 ms	Stable frequency at start-up
11	Reserved	

Table 8-10. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

CKSEL3..0	Start-up Time from Power-down and Power-save	Recommended Usage
0100[1]	1K CK	
0101	32K CK	Stable frequency at start-up

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

8.6 Calibrated Internal RC Oscillator

By default, the Internal RC Oscillator provides an approximate 8.0 MHz clock. Though voltage and temperature dependent, this clock can be very accurately calibrated by the user. See Table 28-1 on page 317 for more details. The device is shipped with the CKDIV8 Fuse programmed. See "System Clock Prescaler" on page 35 for more details.

This clock may be selected as the system clock by programming the CKSEL Fuses as shown in Table 8-11. If selected, it will operate with no external components. During reset, hardware loads the pre-programmed calibration value into the OSCCAL Register and thereby automatically calibrates the RC Oscillator. The accuracy of this calibration is shown as Factory calibration in Table 28-1 on page 317.

By changing the OSCCAL register from SW, see "OSCCAL – Oscillator Calibration Register" on page 37, it is possible to get a higher calibration accuracy than by using the factory calibration. The accuracy of this calibration is shown as User calibration in Table 28-1 on page 317.

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 298.

Table 8-11. Internal Calibrated RC Oscillator Operating Modes

Frequency Range[2] (MHz)	CKSEL3..0
7.3 - 8.1	0010[1]

Notes: 1. The device is shipped with this option selected.
2. If 8 MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8.

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 8-12 on page 33.

Table 8-12. Start-up times for the internal calibrated RC Oscillator clock selection

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	SUT1..0
BOD enabled	6 CK	14CK[1]	00
Fast rising power	6 CK	14CK + 4.1 ms	01
Slowly rising power	6 CK	14CK + 65 ms[2]	10
Reserved			11

Note: 1. If the RSTDISBL fuse is programmed, this start-up time will be increased to 14CK + 4.1 ms to ensure programming mode can be entered.
2. The device is shipped with this option selected.

8.7 128 kHz Internal Oscillator

The 128 kHz internal Oscillator is a low power Oscillator providing a clock of 128 kHz. The frequency is nominal at 3V and 25°C. This clock may be select as the system clock by programming the CKSEL Fuses to “11” as shown in Table 8-13.

Table 8-13. 128 kHz Internal Oscillator Operating Modes

Nominal Frequency(1)	CKSEL3..0
128 kHz	0011

Note: 1. Note that the 128 kHz oscillator is a very low power clock source, and is not designed for high accuracy.

When this clock source is selected, start-up times are determined by the SUT Fuses as shown in Table 8-14.

Table 8-14. Start-up Times for the 128 kHz Internal Oscillator

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset	SUT1..0
BOD enabled	6 CK	14CK(1)	00
Fast rising power	6 CK	14CK + 4 ms	01
Slowly rising power	6 CK	14CK + 64 ms	10
Reserved			11

Note: 1. If the RSTDISBL fuse is programmed, this start-up time will be increased to 14CK + 4.1 ms to ensure programming mode can be entered.

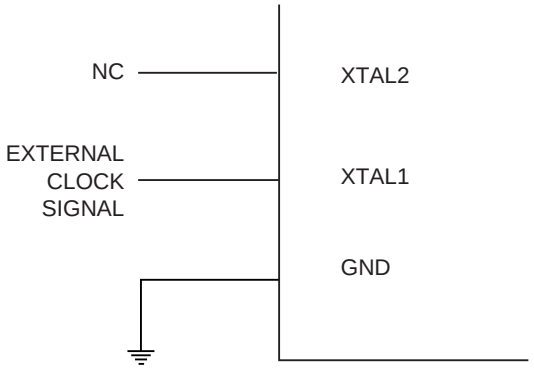
8.8 External Clock

To drive the device from an external clock source, XTAL1 should be driven as shown in Figure 8-4 on page 34. To run the device on an external clock, the CKSEL Fuses must be programmed to “0000” (see Table 8-15).

Table 8-15. Crystal Oscillator Clock Frequency

Frequency	CKSEL3..0
0 - 20 MHz	0000

Figure 8-4. External Clock Drive Configuration



When this clock source is selected, start-up times are determined by the SUT Fuses as shown in Table 8-16.

Table 8-16. Start-up Times for the External Clock Selection

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	SUT1..0
BOD enabled	6 CK	14CK	00
Fast rising power	6 CK	14CK + 4.1 ms	01
Slowly rising power	6 CK	14CK + 65 ms	10
Reserved			11

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. If changes of more than 2% is required, ensure that the MCU is kept in Reset during the changes.

Note that the System Clock Prescaler can be used to implement run-time changes of the internal clock frequency while still ensuring stable operation. Refer to "System Clock Prescaler" on page 35 for details.

8.9 Clock Output Buffer

The device can output the system clock on the CLKO pin. To enable the output, the CKOUT Fuse has to be programmed. This mode is suitable when the chip clock is used to drive other circuits on the system. The clock also will be output during reset, and the normal operation of I/O pin will be overridden when the fuse is programmed. Any clock source, including the internal RC Oscillator, can be selected when the clock is output on CLKO. If the System Clock Prescaler is used, it is the divided system clock that is output.

8.10 Timer/Counter Oscillator

ATmega48PA/88PA/168PA/328P uses the same crystal oscillator for Low-frequency Oscillator and Timer/Counter Oscillator. See "Low Frequency Crystal Oscillator" on page 32 for details on the oscillator and crystal requirements.

ATmega48PA/88PA/168PA/328P share the Timer/Counter Oscillator Pins (TOSC1 and TOSC2) with XTAL1 and XTAL2. When using the Timer/Counter Oscillator, the system clock needs to be four times the oscillator frequency. Due to this and the pin sharing, the Timer/Counter Oscillator can only be used when the Calibrated Internal RC Oscillator is selected as system clock source.

Applying an external clock source to TOSC1 can be done if EXTCLK in the ASSR Register is written to logic one. See "Asynchronous Operation of Timer/Counter2" on page 155 for further description on selecting external clock as input instead of a 32.768 kHz watch crystal.

8.11 System Clock Prescaler

The ATmega48PA/88PA/168PA/328P has a system clock prescaler, and the system clock can be divided by setting the "CLKPR – Clock Prescale Register" on page 377. This feature can be used to decrease the system clock frequency and the power consumption when the requirement for processing power is low. This can be used with all clock source options, and it will affect the clock frequency of the CPU and all synchronous peripherals. $clk_{I/O}$, clk_{ADC}, clk_{CPU}, and clk_{FLASH} are divided by a factor as shown in Table 28-3 on page 318.

When switching between prescaler settings, the System Clock Prescaler ensures that no glitches occurs in the clock system. It also ensures that no intermediate frequency is higher than neither the clock frequency corresponding to the previous setting, nor the clock frequency corresponding to the new setting. The ripple counter that implements the prescaler runs at the frequency of the undivided clock, which may be faster than the CPU's clock frequency. Hence, it is not possible to determine the state of the prescaler - even if it were readable, and the exact time it takes to switch from one clock division to the other cannot be exactly predicted. From the time the CLKPS values are written, it takes between T1 + T2 and T1 + 2 * T2 before the new clock frequency is active. In this interval, 2 active clock edges are produced. Here, T1 is the previous clock period, and T2 is the period corresponding to the new prescaler setting.

To avoid unintentional changes of clock frequency, a special write procedure must befollowed to change the CLKPS bits:

1. Write the Clock Prescaler Change Enable (CLKPCE) bit to one and all other bitsin CLKPR to zero.
2. Within four cycles, write the desired value to CLKPS while writing a zero to CLKPCE.

Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted.

8.12 Register Description

8.12.1 OSCCAL – Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0	
(0x66)	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**

The Oscillator Calibration Register is used to trim the Calibrated Internal RC Oscillator to remove process variations from the oscillator frequency. A pre-programmed calibration value is automatically written to this register during chip reset, giving the Factory calibrated frequency as specified in Table 28-1 on page 317. The application software can write this register to change the oscillator frequency. The oscillator can be calibrated to frequencies as specified in Table 28-1 on page 317. Calibration outside that range is not guaranteed.

Note that this oscillator is used to time EEPROM and Flash write accesses, and these write times will be affected accordingly. If the EEPROM or Flash are written, do not calibrate to more than 8.8 MHz. Otherwise, the EEPROM or Flash write may fail.

The CAL7 bit determines the range of operation for the oscillator. Setting this bit to 0 gives the lowest frequency range, setting this bit to 1 gives the highest frequency range. The two frequency ranges are overlapping, in other words a setting of OSCCAL = 0x7F gives a higher frequency than OSCCAL = 0x80.

The CAL6..0 bits are used to tune the frequency within the selected range. A setting of 0x00 gives the lowest frequency in that range, and a setting of 0x7F gives the highest frequency in the range.

8.12.2 CLKPR – Clock Prescale Register

Bit	7	6	5	4	3	2	1	0	
(0x61)	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	CLKPR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 7 – CLKPCE: Clock Prescaler Change Enable**

The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.

- **Bits 3..0 – CLKPS3..0: Clock Prescaler Select Bits 3 - 0**

These bits define the division factor between the selected clock source and the internal system clock. These bits can be written run-time to vary the clock frequency to suit the application requirements. As the divider divides the master clock input to the MCU, the speed of all synchronous peripherals is reduced when a division factor is used. The division factors are given in Table 8-17 on page 38.

The CKDIV8 Fuse determines the initial value of the CLKPS bits. If CKDIV8 is unprogrammed, the CLKPS bits will be reset to “0000”. If CKDIV8 is programmed, CLKPS bits are reset to “0011”, giving a division factor of 8 at start up. This feature should be used if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. Note that any value can be written to the CLKPS bits regardless of the CKDIV8 Fuse setting. The Application software must ensure that a sufficient division factor is chosen if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. The device is shipped with the CKDIV8 Fuse programmed.

Table 8-17. Clock Prescaler Select

CLKPS3	CLKPS2	CLKPS1	CLKPS0	Clock Division Factor
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

9. 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.

When enabled, the Brown-out Detector (BOD) actively monitors the power supply voltage during the sleep periods. To further save power, it is possible to disable the BOD in some sleep modes. See "BOD Disable" on page 40 for more details.

9.1 Sleep Modes

Figure 8-1 on page 26 presents the different clock systems in the ATmega48PA/88PA/168PA/328P, and their distribution. The figure is helpful in selecting an appropriate sleep mode. Table 9-1 shows the different sleep modes, their wake up sources BOD disable ability.

Table 9-1. 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 Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O	Software BOD Disable
Idle			X	X	X	X	X(2)	X	X	X	X	X	X	X	
ADC Noise Reduction				X	X	X	X(2)	X(3)	X	X(2)	X	X	X		
Power-down								X(3)	X				X		X
Power-save					X		X(2)	X(3)	X	X			X		X
Standby(1)						X		X(3)	X				X		X
Extended Standby					X(2)	X	X(2)	X(3)	X	X			X		X

Notes: 1. Only recommended with external crystal or resonator selected as clock source.
2. If Timer/Counter2 is running in asynchronous mode.
3. For INT1 and INT0, only level interrupt.

To enter any of the six sleep modes, the SE bit in SMCR must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the SMCR 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 9-2 on page 44 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, 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.

9.2 BOD Disable

When the Brown-out Detector (BOD) is enabled by BODLEVEL fuses, Table 27-7 on page 296, the BOD is actively monitoring the power supply voltage during a sleep period. To save power, it is possible to disable the BOD by software for some of the sleep modes, see Table 9-1 on page 39. The sleep mode power consumption will then be at the same level as when BOD is globally disabled by fuses. If BOD is disabled in software, the BOD function is turned off immediately after entering the sleep mode. Upon wake-up from sleep, BOD is automatically enabled again. This ensures safe operation in case the V_{CC} level has dropped during the sleep period.

When the BOD has been disabled, the wake-up time from sleep mode will be approximately 60 µs to ensure that the BOD is working correctly before the MCU continues executing code.

BOD disable is controlled by bit 6, BODS (BOD Sleep) in the control register MCUCR, see "MCUCR – MCU Control Register" on page 44. Writing this bit to one turns off the BOD in relevant sleep modes, while a zero in this bit keeps BOD active. Default setting keeps BOD active, i.e. BODS set to zero.

Writing to the BODS bit is controlled by a timed sequence and an enable bit, see "MCUCR – MCU Control Register" on page 44.

9.3 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 the SPI, USART, Analog Comparator, ADC, 2-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.

9.4 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 2-wire Serial Interface address watch, Timer/Counter2[1], 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 from the ADC Conversion Complete interrupt, only an External Reset, a Watchdog System Reset, a Watchdog Interrupt, a Brown-out Reset, a 2-wire Serial Interface address match, a Timer/Counter2 interrupt, an SPM/EEPROM ready interrupt, an external level interrupt on INT0 or INT1 or a pin change interrupt can wake up the MCU from ADC Noise Reduction mode.

Note: 1. Timer/Counter2 will only keep running in asynchronous mode, see "8-bit Timer/Counter2 with PWM and Asynchronous Operation" on page 144 for details.

9.5 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 2-wire Serial Interface address watch, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog System Reset, a Watchdog Interrupt, a Brown-out Reset, a 2-wire Serial Interface address match, an external level interrupt on INT0 or INT1, or a pin change interrupt 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 70 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 27.

9.6 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 enabled, it will keep running 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 TIMSK2, and the Global Interrupt Enable bit in SREG is set.

If Timer/Counter2 is not running, Power-down mode is recommended instead of Power-save mode.

The Timer/Counter2 can be clocked both synchronously and asynchronously in Power-save mode. If Timer/Counter2 is not using the asynchronous clock, the Timer/Counter Oscillator is stopped during sleep. If Timer/Counter2 is not using the synchronous clock, the clock source is stopped during sleep. Note that even if the synchronous clock is running in Power-save, this clock is only available for Timer/Counter2.

9.7 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.

9.8 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 with the exception that the Oscillator is kept running. From Extended Standby mode, the device wakes up in six clock cycles.

9.9 Power Reduction Register

The Power Reduction Register (PRR), see ”PRR – Power Reduction Register” on page 45, provides a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied, hence the peripheral should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before shutdown.

Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. In all other sleep modes, the clock is already stopped.

9.10 Minimizing Power Consumption

There are several possibilities 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.

9.10.1 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 250 for details on ADC operation.

9.10.2 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 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 246 for details on how to configure the Analog Comparator.

9.10.3 Brown-out Detector

If the Brown-out Detector is not needed by the application, this module should be turned off. If the Brown-out Detector is enabled by the BODLEVEL Fuses, 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 48 for details on how to configure the Brown-out Detector.

9.10.4 Internal Voltage Reference

The Internal Voltage Reference will be enabled when needed by the Brown-out Detection, 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 49 for details on the start-up time.

9.10.5 Watchdog Timer

If the Watchdog Timer is not needed in the application, the 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 50 for details on how to configure the Watchdog Timer.

9.10.6 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important is then to ensure that no pins drive resistive loads. In sleep modes where 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 79 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.

For analog input pins, the digital input buffer should be disabled at all times. An analog signal level close to $V_{CC}/2$ on an input pin can cause significant current even in active mode. Digital input buffers can be disabled by writing to the Digital Input Disable Registers (DIDR1 and DIDR0). Refer to ”DIDR1 – Digital Input Disable Register 1” on page 249 and ”DIDR0 – Digital Input Disable Register 0” on page 266 for details.

9.10.7 On-chip Debug System

If the On-chip debug system is enabled by the DWEN Fuse and the chip enters sleep mode, the main clock source is enabled and hence always consumes power. In the deeper sleep modes, this will contribute significantly to the total current consumption.

9.11 Register Description

9.11.1 SMCR – Sleep Mode Control Register

The Sleep Mode Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	–	–	–	–	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..4 Res: Reserved Bits**

These bits are unused in the ATmega48PA/88PA/168PA/328P, and will always be read as zero.

- **Bits 3..1 – SM2..0: Sleep Mode Select Bits 2, 1, and 0**

These bits select between the five available sleep modes as shown in Table 9-2.

Table 9-2. 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	External Standby(1)

Note: 1. Standby mode is only recommended for use with external crystals or resonators.

- **Bit 0 – 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 programmer's 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.

9.11.2 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – BODS: BOD Sleep**

The BODS bit must be written to logic one in order to turn off BOD during sleep, see Table 9-1 on page 39. Writing to the BODS bit is controlled by a timed sequence and an enable bit, BODSE in MCUCR. To disable BOD in relevant sleep modes, both BODS and BODSE must first

be set to one. Then, to set the BODS bit, BODS must be set to one and BODSE must be set to zero within four clock cycles.

The BODS bit is active three clock cycles after it is set. A sleep instruction must be executed while BODS is active in order to turn off the BOD for the actual sleep mode. The BODS bit is automatically cleared after three clock cycles.

- **Bit 5 – BODSE: BOD Sleep Enable**

BODSE enables setting of BODS control bit, as explained in BODS bit description. BOD disable is controlled by a timed sequence.

9.11.3 PRR – Power Reduction Register

Bit	7	6	5	4	3	2	1	0	
(0x64)	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC	PRR
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 - PRTWI: Power Reduction TWI**

Writing a logic one to this bit shuts down the TWI by stopping the clock to the module. When waking up the TWI again, the TWI should be re initialized to ensure proper operation.

- **Bit 6 - PRTIM2: Power Reduction Timer/Counter2**

Writing a logic one to this bit shuts down the Timer/Counter2 module in synchronous mode (AS2 is 0). When the Timer/Counter2 is enabled, operation will continue like before the shutdown.

- **Bit 5 - PRTIM0: Power Reduction Timer/Counter0**

Writing a logic one to this bit shuts down the Timer/Counter0 module. When the Timer/Counter0 is enabled, operation will continue like before the shutdown.

- **Bit 4 - Res: Reserved bit**

This bit is reserved in ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 3 - PRTIM1: Power Reduction Timer/Counter1**

Writing a logic one to this bit shuts down the Timer/Counter1 module. When the Timer/Counter1 is enabled, operation will continue like before the shutdown.

- **Bit 2 - PRSPI: Power Reduction Serial Peripheral Interface**

If using debugWIRE On-chip Debug System, this bit should not be written to one.

Writing a logic one to this bit shuts down the Serial Peripheral Interface by stopping the clock to the module. When waking up the SPI again, the SPI should be re initialized to ensure proper operation.

- **Bit 1 - PRUSART0: Power Reduction USART0**

Writing a logic one to this bit shuts down the USART by stopping the clock to the module. When waking up the USART again, the USART should be re initialized to ensure proper operation.

- **Bit 0 - PRADC: Power Reduction ADC**

Writing a logic one to this bit shuts down the ADC. The ADC must be disabled before shut down. The analog comparator cannot use the ADC input MUX when the ADC is shut down.

10. System Control and Reset

10.1 Resetting the AVR

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. For the ATmega168PA, the instruction placed at the Reset Vector must be a JMP – Absolute Jump – instruction to the reset handling routine. For the ATmega48PA and ATmega88PA, the instruction placed at the Reset Vector must be an RJMP – Relative 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 (ATmega88PA/168PA only). The circuit diagram in Figure 10-1 on page 47 shows the reset logic. Table 28-3 on page 318 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.

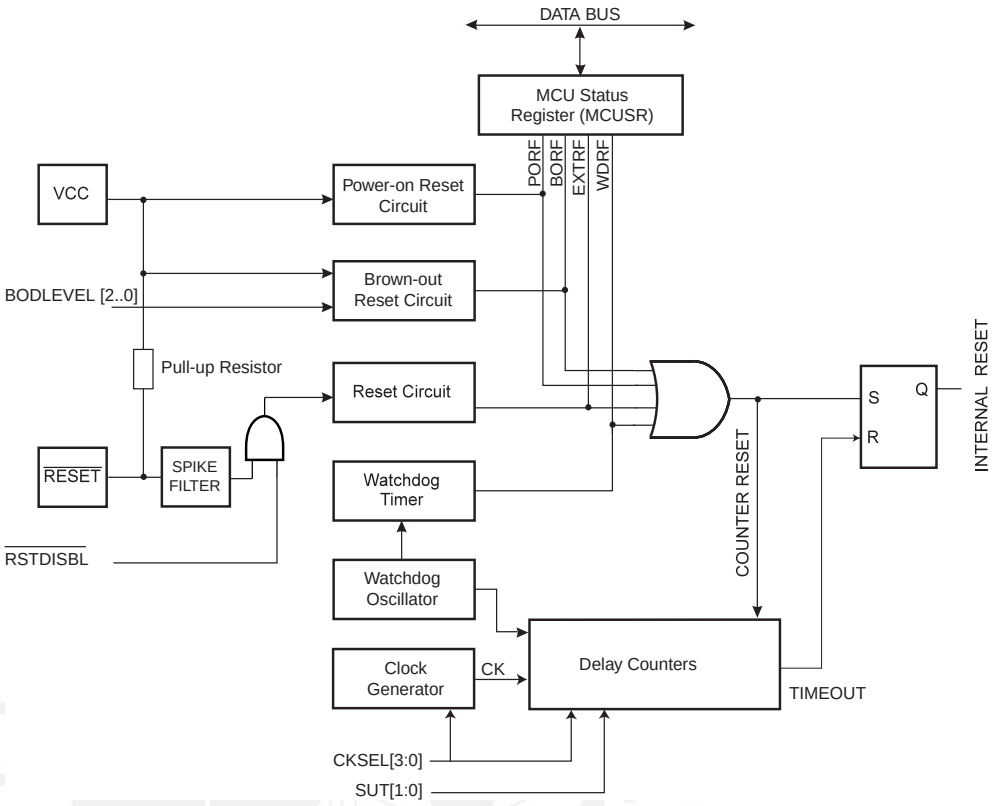
After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT and CKSEL Fuses. The different selections for the delay period are presented in ”Clock Sources” on page 27.

10.2 Reset Sources

The ATmega48PA/88PA/168PA/328P has four 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 System Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog System Reset mode 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.

Figure 10-1. Reset Logic



10.3 Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in "System and Reset Characteristics" on page 318. 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 10-2. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}

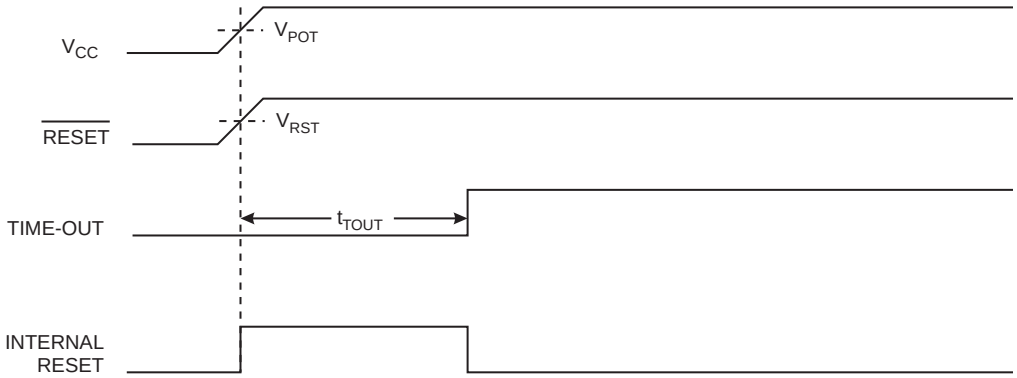
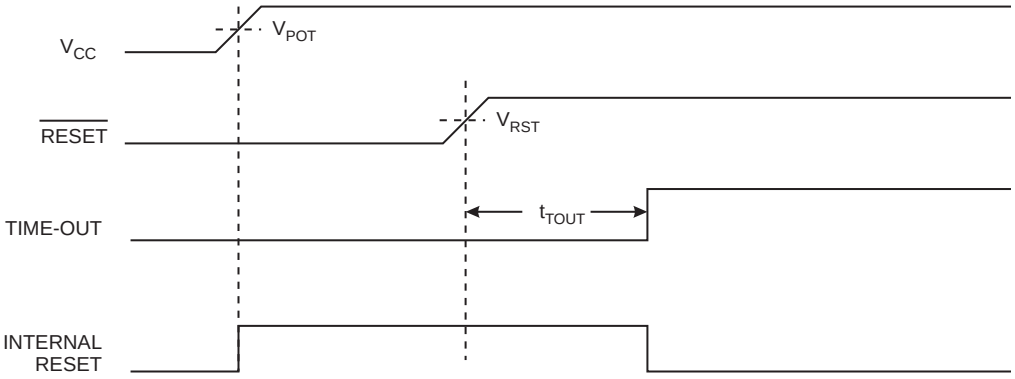


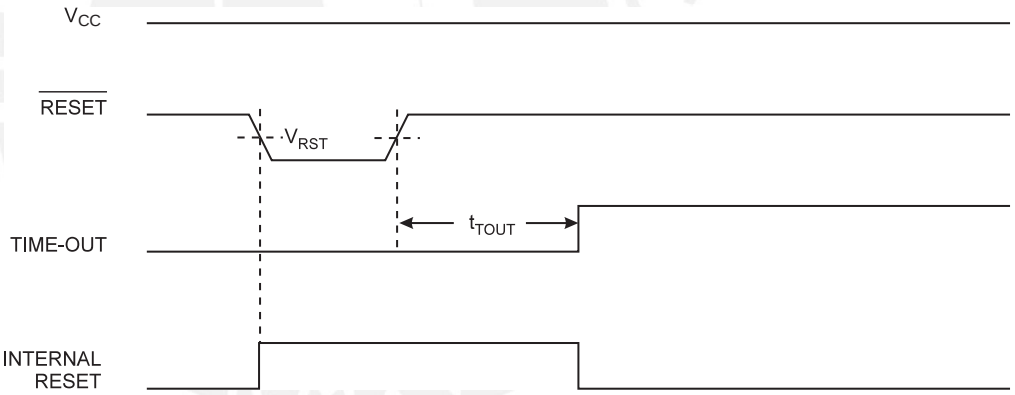
Figure 10-3. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



10.4 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 "System and Reset Characteristics" on page 318) 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. The External Reset can be disabled by the RSTDISBL fuse, see Table 27-7 on page 296.

Figure 10-4. External Reset During Operation

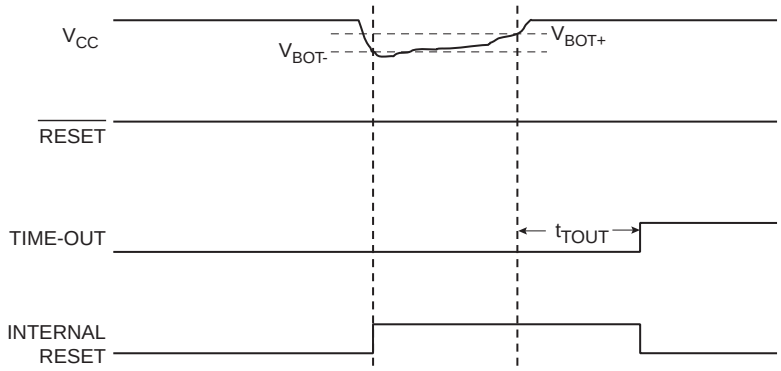


10.5 Brown-out Detection

ATmega48PA/88PA/168PA/328P 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 BODLEVEL Fuses. 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 $V_{BOT-} = V_{BOT} - V_{HYST}/2$.When the BOD is enabled, and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 10-5 on page 49), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 10-5 on page 49), 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 "System and Reset Characteristics" on page 318.

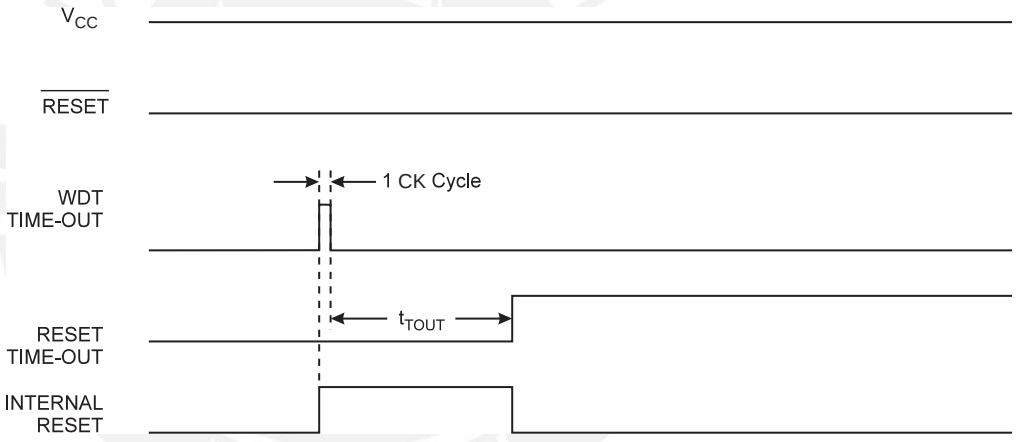
Figure 10-5. Brown-out Reset During Operation



10.6 Watchdog System 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 50 for details on operation of the Watchdog Timer.

Figure 10-6. Watchdog System Reset During Operation



10.7 Internal Voltage Reference

ATmega48PA/88PA/168PA/328P 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.

10.7.1 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 "System and Reset Characteristics" on page 318. 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 BODLEVEL [2:0] Fuses).
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.

10.8 Watchdog Timer

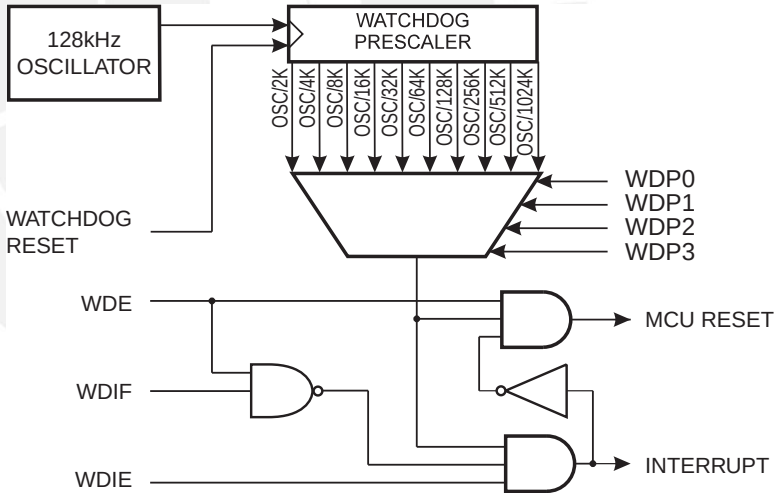
10.8.1 Features

- **Clocked from separate On-chip Oscillator**
- **3 Operating modes**
  - **Interrupt**
  - **System Reset**
  - **Interrupt and System Reset**
- **Selectable Time-out period from 16ms to 8s**
- **Possible Hardware fuse Watchdog always on (WDTON) for fail-safe mode**

10.8.2 Overview

ATmega48PA/88PA/168PA/328P has an Enhanced Watchdog Timer (WDT). The WDT is a timer counting cycles of a separate on-chip 128 kHz oscillator. The WDT gives an interrupt or a system reset when the counter reaches a given time-out value. In normal operation mode, it is required that the system uses the WDR - Watchdog Timer Reset - instruction to restart the counter before the time-out value is reached. If the system doesn't restart the counter, an interrupt or system reset will be issued.

Figure 10-7. Watchdog Timer



In Interrupt mode, the WDT gives an interrupt when the timer expires. This interrupt can be used to wake the device from sleep-modes, and also as a general system timer. One example is to limit the maximum time allowed for certain operations, giving an interrupt when the operation has run longer than expected. In System Reset mode, the WDT gives a reset when the timer expires. This is typically used to prevent system hang-up in case of runaway code. The third mode, Interrupt and System Reset mode, combines the other two modes by first giving an interrupt and then switch to System Reset mode. This mode will for instance allow a safe shutdown by saving critical parameters before a system reset.

The Watchdog always on (WDTON) fuse, if programmed, will force the Watchdog Timer to System Reset mode. With the fuse programmed the System Reset mode bit (WDE) and Interrupt

mode bit (WDIE) are locked to 1 and 0 respectively. To further ensure program security, alterations to the Watchdog set-up must follow timed sequences. The sequence for clearing WDE and changing time-out configuration is as follows:

1. In the same operation, write a logic one to the Watchdog change enable bit (WDCE) and WDE. A logic one must be written to WDE regardless of the previous value of the WDE bit.
2. Within the next four clock cycles, write the WDE and Watchdog prescaler bits (WDP) as desired, but with the WDCE bit cleared. This must be done in one operation.

The following code example shows one assembly and one C function for turning off the Watchdog Timer. The example assumes that interrupts are controlled (e.g. by disabling interrupts globally) so that no interrupts will occur during the execution of these functions.

Assembly Code Example[1]

```
WDT_off:
  ; Turn off global interrupt
  cli
  ; Reset Watchdog Timer
  wdr
  ; Clear WDRF in MCUSR
  in    r16, MCUSR
  andi  r16, (0xff & (0<<WDRF))
  out   MCUSR, r16
  ; Write logical one to WDCE and WDE
  ; Keep old prescaler setting to prevent unintentional time-out
  lds r16, WDTCSR
  ori   r16, (1<<WDCE) | (1<<WDE)
  sts WDTCSR, r16
  ; Turn off WDT
  ldi   r16, (0<<WDE)
  sts WDTCSR, r16
  ; Turn on global interrupt
  sei
  ret
```

C Code Example[1]

```
void WDT_off(void)
{
  __disable_interrupt();
  __watchdog_reset();
  /* Clear WDRF in MCUSR */
  MCUSR &= ~(1<<WDRF);
  /* Write logical one to WDCE and WDE */
  /* Keep old prescaler setting to prevent unintentional time-out */
  WDTCSR |= (1<<WDCE) | (1<<WDE);
  /* Turn off WDT */
  WDTCSR = 0x00;
  __enable_interrupt();
}
```

Note: 1. See "About Code Examples" on page 7.

Note: If the Watchdog is accidentally enabled, for example by a runaway pointer or brown-out condition, the device will be reset and the Watchdog Timer will stay enabled. If the code is not set up to handle the Watchdog, this might lead to an eternal loop of time-out resets. To avoid this situation, the application software should always clear the Watchdog System Reset Flag (WDRF) and the WDE control bit in the initialisation routine, even if the Watchdog is not in use.

The following code example shows one assembly and one C function for changing the time-out value of the Watchdog Timer.

Assembly Code Example[1]

```
WDT_Prescaler_Change:
  ; Turn off global interrupt
  cli
  ; Reset Watchdog Timer
  wdr
  ; Start timed sequence
  lds r16, WDTCSR
  ori   r16, (1<<WDCE) | (1<<WDE)
  sts WDTCSR, r16
  ; --  Got four cycles to set the new values from here -
  ; Set new prescaler(time-out) value = 64K cycles (~0.5 s)
  ldi   r16, (1<<WDE) | (1<<WDP2) | (1<<WDP0)
  sts WDTCSR, r16
  ; --  Finished setting new values, used 2 cycles -
  ; Turn on global interrupt
  sei
  ret
```

C Code Example[1]

```
void WDT_Prescaler_Change(void)
{
  __disable_interrupt();
  __watchdog_reset();
  /* Start timed  equence */
  WDTCSR |= (1<<WDCE) | (1<<WDE);
  /* Set new prescaler(time-out) value = 64K cycles (~0.5 s) */
  WDTCSR  = (1<<WDE) | (1<<WDP2) | (1<<WDP0);
  __enable_interrupt();
}
```

Note: 1. See "About Code Examples" on page 7.

Note: The Watchdog Timer should be reset before any change of the WDP bits, since a change in the WDP bits can result in a time-out when switching to a shorter time-out period.

10.9 Register Description

10.9.1 MCUSR – MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 7..4: Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 3 – WDRF: Watchdog System Reset Flag**

This bit is set if a Watchdog System 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 MCUSR 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.

10.9.2 WDTCSR – Watchdog Timer Control Register

Bit	7	6	5	4	3	2	1	0	
(0x60)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	0	0	0	

- **Bit 7 - WDIF: Watchdog Interrupt Flag**

This bit is set when a time-out occurs in the Watchdog Timer and the Watchdog Timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the Watchdog Time-out Interrupt is executed.

- **Bit 6 - WDIE: Watchdog Interrupt Enable**

When this bit is written to one and the I-bit in the Status Register is set, the Watchdog Interrupt is enabled. If WDE is cleared in combination with this setting, the Watchdog Timer is in Interrupt Mode, and the corresponding interrupt is executed if time-out in the Watchdog Timer occurs. If WDE is set, the Watchdog Timer is in Interrupt and System Reset Mode. The first time-out in the

Watchdog Timer will set WDIF. Executing the corresponding interrupt vector will clear WDIE and WDIF automatically by hardware (the Watchdog goes to System Reset Mode). This is useful for keeping the Watchdog Timer security while using the interrupt. To stay in Interrupt and System Reset Mode, WDIE must be set after each interrupt. This should however not be done within the interrupt service routine itself, as this might compromise the safety-function of the Watchdog System Reset mode. If the interrupt is not executed before the next time-out, a System Reset will be applied.

Table 10-1. Watchdog Timer Configuration

WDTON(1)	WDE	WDIE	Mode	Action on Time-out
1	0	0	Stopped	None
1	0	1	Interrupt Mode	Interrupt
1	1	0	System Reset Mode	Reset
1	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
0	x	x	System Reset Mode	Reset

Note: 1. WDTON Fuse set to “0” means programmed and “1” means unprogrammed.

- **Bit 4 - WDCE: Watchdog Change Enable**

This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE must be set.

Once written to one, hardware will clear WDCE after four clock cycles.

- **Bit 3 - WDE: Watchdog System Reset Enable**

WDE is overridden by WDRF in MCUSR. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

- **Bit 5, 2..0 - WDP3..0: Watchdog Timer Prescaler 3, 2, 1 and 0**

The WDP3..0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is running. The different prescaling values and their corresponding time-out periods are shown in Table 10-2 on page 55.

Table 10-2. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 5.0V
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32 ms
0	0	1	0	8K (8192) cycles	64 ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32768) cycles	0.25 s
0	1	0	1	64K (65536) cycles	0.5 s
0	1	1	0	128K (131072) cycles	1.0 s

Table 10-2. Watchdog Timer Prescale Select (Continued)

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 5.0V
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s
1	0	1	0	Reserved	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

11. Interrupts

This section describes the specifics of the interrupt handling as performed in ATmega48PA/88PA/168PA/328P. For a general explanation of the AVR interrupt handling, refer to ”Reset and Interrupt Handling” on page 14.

The interrupt vectors in ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P are generally the same, with the following differences:

- Each Interrupt Vector occupies two instruction words in ATmega168PA and ATmega328P, and one instruction word in ATmega48PA and ATmega88PA.
- ATmega48PA does not have a separate Boot Loader Section. In ATmega88PA, ATmega168PA and ATmega328P, the Reset Vector is affected by the BOOTRST fuse, and the Interrupt Vector start address is affected by the IVSEL bit in MCUCR.

11.1 Interrupt Vectors in ATmega48PA

Table 11-1. Reset and Interrupt Vectors in ATmega48PA

Vector No.	Program Address	Source	Interrupt Definition
1	0x000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	PCINT0	Pin Change Interrupt Request 0
5	0x004	PCINT1	Pin Change Interrupt Request 1
6	0x005	PCINT2	Pin Change Interrupt Request 2
7	0x006	WDT	Watchdog Time-out Interrupt
8	0x007	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x008	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x009	TIMER2 OVF	Timer/Counter2 Overflow
11	0x00A	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x00B	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x00C	TIMER1 COMPB	Timer/Coutner1 Compare Match B
14	0x00D	TIMER1 OVF	Timer/Counter1 Overflow
15	0x00E	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x00F	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x010	TIMER0 OVF	Timer/Counter0 Overflow
18	0x011	SPI, STC	SPI Serial Transfer Complete
19	0x012	USART, RX	USART Rx Complete
20	0x013	USART, UDRE	USART, Data Register Empty
21	0x014	USART, TX	USART, Tx Complete
22	0x015	ADC	ADC Conversion Complete
23	0x016	EE READY	EEPROM Ready

Table 11-1. Reset and Interrupt Vectors in ATmega48PA (Continued)

Vector No.	Program Address	Source	Interrupt Definition
24	0x017	ANALOG COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM READY	Store Program Memory Ready

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega48PA is:

```
Address  Labels Code                 Comments
0x000           rjmp   RESET          ; Reset Handler
0x001           rjmp   EXT_INT0       ; IRQ0 Handler
0x002           rjmp   EXT_INT1       ; IRQ1 Handler
0x003           rjmp   PCINT0         ; PCINT0 Handler
0x004           rjmp   PCINT1         ; PCINT1 Handler
0x005           rjmp   PCINT2         ; PCINT2 Handler
0x006           rjmp   WDT            ; Watchdog Timer Handler
0x007           rjmp   TIM2_COMPA     ; Timer2 Compare A Handler
0x008           rjmp   TIM2_COMPB     ; Timer2 Compare B Handler
0x009           rjmp   TIM2_OVF       ; Timer2 Overflow Handler
0x00A           rjmp   TIM1_CAPT      ; Timer1 Capture Handler
0x00B           rjmp   TIM1_COMPA     ; Timer1 Compare A Handler
0x00C           rjmp   TIM1_COMPB     ; Timer1 Compare B Handler
0x00D           rjmp   TIM1_OVF       ; Timer1 Overflow Handler
0x00E           rjmp   TIM0_COMPA     ; Timer0 Compare A Handler
0x00F           rjmp   TIM0_COMPB     ; Timer0 Compare B Handler
0x010           rjmp   TIM0_OVF       ; Timer0 Overflow Handler
0x011           rjmp   SPI_STC        ; SPI Transfer Complete Handler
0x012           rjmp   USART_RXC      ; USART, RX Complete Handler
0x013           rjmp   USART_UDRE     ; USART, UDR Empty Handler
0x014           rjmp   USART_TXC      ; USART, TX Complete Handler
0x015           rjmp   ADC            ; ADC Conversion Complete Handler
0x016           rjmp   EE_RDY         ; EEPROM Ready Handler
0x017           rjmp   ANA_COMP       ; Analog Comparator Handler
0x018           rjmp   TWI            ; 2-wire Serial Interface Handler
0x019           rjmp   SPM_RDY        ; Store Program Memory Ready Handler
;
0x01ARESET:     ldi    r16, high(RAMEND); Main program start
0x01B           out    SPH,r16        ; Set Stack Pointer to top of RAM
0x01C           ldi    r16, low(RAMEND)
0x01D           out    SPL,r16
0x01E           sei                   ; Enable interrupts
0x01F           <instr>  xxx
  ...     ...     ...   ...
```

11.2 Interrupt Vectors in ATmega88PA

Table 11-2. Reset and Interrupt Vectors in ATmega88PA

Vector No.	Program Address(2)	Source	Interrupt Definition
1	0x000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	PCINT0	Pin Change Interrupt Request 0
5	0x004	PCINT1	Pin Change Interrupt Request 1
6	0x005	PCINT2	Pin Change Interrupt Request 2
7	0x006	WDT	Watchdog Time-out Interrupt
8	0x007	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x008	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x009	TIMER2 OVF	Timer/Counter2 Overflow
11	0x00A	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x00B	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x00C	TIMER1 COMPB	Timer/Coutner1 Compare Match B
14	0x00D	TIMER1 OVF	Timer/Counter1 Overflow
15	0x00E	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x00F	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x010	TIMER0 OVF	Timer/Counter0 Overflow
18	0x011	SPI, STC	SPI Serial Transfer Complete
19	0x012	USART, RX	USART Rx Complete
20	0x013	USART, UDRE	USART, Data Register Empty
21	0x014	USART, TX	USART, Tx Complete
22	0x015	ADC	ADC Conversion Complete
23	0x016	EE READY	EEPROM Ready
24	0x017	ANALOG COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM READY	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, ATmega88PA, ATmega168PA and ATmega328P" on page 277.
2. When the IVSEL bit in MCUCR 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 11-3 on page 60 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 11-3. Reset and Interrupt Vectors Placement in ATmega88PA[1]

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x001
1	1	0x000	Boot Reset Address + 0x001
0	0	Boot Reset Address	0x001
0	1	Boot Reset Address	Boot Reset Address + 0x001

Note: 1. The Boot Reset Address is shown in Table 26-7 on page 289. 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 ATmega88PA is:

```
Address  Labels Code                 Comments
0x000           rjmp   RESET          ; Reset Handler
0x001           rjmp   EXT_INT0       ; IRQ0 Handler
0x002           rjmp   EXT_INT1       ; IRQ1 Handler
0x003           rjmp   PCINT0         ; PCINT0 Handler
0x004           rjmp   PCINT1         ; PCINT1 Handler
0x005           rjmp   PCINT2         ; PCINT2 Handler
0x006           rjmp   WDT            ; Watchdog Timer Handler
0x007           rjmp   TIM2_COMPA     ; Timer2 Compare A Handler
0X008           rjmp   TIM2_COMPB     ; Timer2 Compare B Handler
0x009           rjmp   TIM2_OVF       ; Timer2 Overflow Handler
0x00A           rjmp   TIM1_CAPT      ; Timer1 Capture Handler
0x00B           rjmp   TIM1_COMPA     ; Timer1 Compare A Handler
0x00C           rjmp   TIM1_COMPB     ; Timer1 Compare B Handler
0x00D           rjmp   TIM1_OVF       ; Timer1 Overflow Handler
0x00E           rjmp   TIM0_COMPA     ; Timer0 Compare A Handler
0x00F           rjmp   TIM0_COMPB     ; Timer0 Compare B Handler
0x010           rjmp   TIM0_OVF       ; Timer0 Overflow Handler
0x011           rjmp   SPI_STC        ; SPI Transfer Complete Handler
0x012           rjmp   USART_RXC      ; USART, RX Complete Handler
0x013           rjmp   USART_UDRE     ; USART, UDR Empty Handler
0x014           rjmp   USART_TXC      ; USART, TX Complete Handler
0x015           rjmp   ADC            ; ADC Conversion Complete Handler
0x016           rjmp   EE_RDY         ; EEPROM Ready Handler
0x017           rjmp   ANA_COMP       ; Analog Comparator Handler
0x018           rjmp   TWI            ; 2-wire Serial Interface Handler
0x019           rjmp   SPM_RDY        ; Store Program Memory Ready Handler
;
0x01ARESET:     ldi    r16, high(RAMEND); Main program start
0x01B           out    SPH,r16        ; Set Stack Pointer to top of RAM
0x01C           ldi    r16, low(RAMEND)
0x01D           out    SPL,r16
0x01E           sei                   ; Enable interrupts
0x01F           <instr>  xxx
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88PA is:

```
Address  Labels Code                 Comments
0x000    RESET: ldi    r16,high(RAMEND); Main program start
0x001           out    SPH,r16        ; Set Stack Pointer to top of RAM
0x002           ldi    r16,low(RAMEND)
0x003           out    SPL,r16
0x004           sei                   ; Enable interrupts
0x005           <instr>  xxx
;
.org 0xC01
0xC01           rjmp   EXT_INT0       ; IRQ0 Handler
0xC02           rjmp   EXT_INT1       ; IRQ1 Handler
...             ...    ...            ;
0xC19           rjmp   SPM_RDY        ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88PA is:

```
Address  Labels Code              Comments
.org 0x001
0x001           rjmp   EXT_INT0       ; IRQ0 Handler
0x002           rjmp   EXT_INT1       ; IRQ1 Handler
...             ...    ...            ;
0x019           rjmp   SPM_RDY        ; Store Program Memory Ready Handler
;
.org 0xC00
0xC00    RESET: ldi    r16,high(RAMEND); Main program start
0xC01           out    SPH,r16        ; Set Stack Pointer to top of RAM
0xC02           ldi    r16,low(RAMEND)
0xC03           out    SPL,r16
0xC04           sei                   ; Enable interrupts
0xC05           <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega88PA is:

```
Address  Labels Code                 Comments
;
.org 0xC00
0xC00           rjmp   RESET          ; Reset handler
0xC01           rjmp   EXT_INT0       ; IRQ0 Handler
0xC02           rjmp   EXT_INT1       ; IRQ1 Handler
...             ...    ...            ;
0xC19           rjmp   SPM_RDY        ; Store Program Memory Ready Handler
;
0xC1A    RESET: ldi    r16,high(RAMEND); Main program start
```

```
0xC1B          out    SPH,r16         ; Set Stack Pointer to top of RAM
0xC1C          ldi    r16,low(RAMEND)
0xC1D          out    SPL,r16
0xC1E          sei                    ; Enable interrupts
0xC1F          <instr>  xxx
```

11.3 Interrupt Vectors in ATmega168PA

Table 11-4. Reset and Interrupt Vectors in ATmega168PA

VectorNo.	Program Address(2)	Source	Interrupt Definition
1	0x0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Coutner1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	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, ATmega88PA, ATmega168PA and ATmega328P” on page 277.
2. When the IVSEL bit in MCUCR 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 11-5 on page 63 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 11-5. Reset and Interrupt Vectors Placement in ATmega168PA(1)

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

Note: 1. The Boot Reset Address is shown in Table 26-7 on page 289. 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 ATmega168PA is:

```
Address  Labels Code                 Comments
0x0000          jmp    RESET         ; Reset Handler
0x0002          jmp    EXT_INT0      ; IRQ0 Handler
0x0004          jmp    EXT_INT1      ; IRQ1 Handler
0x0006          jmp    PCINT0        ; PCINT0 Handler
0x0008          jmp    PCINT1        ; PCINT1 Handler
0x000A          jmp    PCINT2        ; PCINT2 Handler
0x000C          jmp    WDT           ; Watchdog Timer Handler
0x000E          jmp    TIM2_COMPA    ; Timer2 Compare A Handler
0x0010          jmp    TIM2_COMPB    ; Timer2 Compare B Handler
0x0012          jmp    TIM2_OVF      ; Timer2 Overflow Handler
0x0014          jmp    TIM1_CAPT     ; Timer1 Capture Handler
0x0016          jmp    TIM1_COMPA    ; Timer1 Compare A Handler
0x0018          jmp    TIM1_COMPB    ; Timer1 Compare B Handler
0x001A          jmp    TIM1_OVF      ; Timer1 Overflow Handler
0x001C          jmp    TIM0_COMPA    ; Timer0 Compare A Handler
0x001E          jmp    TIM0_COMPB    ; Timer0 Compare B Handler
0x0020          jmp    TIM0_OVF      ; Timer0 Overflow Handler
0x0022          jmp    SPI_STC       ; SPI Transfer Complete Handler
0x0024          jmp    USART_RXC     ; USART, RX Complete Handler
0x0026          jmp    USART_UDRE    ; USART, UDR Empty Handler
0x0028          jmp    USART_TXC     ; USART, TX Complete Handler
0x002A          jmp    ADC           ; ADC Conversion Complete Handler
0x002C          jmp    EE_RDY        ; EEPROM Ready Handler
0x002E          jmp    ANA_COMP      ; Analog Comparator Handler
0x0030          jmp    TWI           ; 2-wire Serial Interface Handler
0x0032          jmp    SPM_RDY       ; Store Program Memory Ready Handler
;
0x0033RESET:    ldi    r16, high(RAMEND); Main program start
```

```
0x0034          out    SPH,r16         ; Set Stack Pointer to top of RAM
0x0035          ldi    r16, low(RAMEND)
0x0036          out    SPL,r16
0x0037          sei                    ; Enable interrupts
0x0038          <instr>  xxx
  ...     ...     ...  ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PA is:

```
Address  Labels Code                  Comments
0x0000   RESET: ldi    r16,high(RAMEND); Main program start
0x0001          out    SPH,r16         ; Set Stack Pointer to top of RAM
0x0002          ldi    r16,low(RAMEND)
0x0003          out    SPL,r16
0x0004          sei                    ; Enable interrupts
0x0005          <instr>  xxx
;
.org 0x1C02
0x1C02          jmp    EXT_INT0        ; IRQ0 Handler
0x1C04          jmp    EXT_INT1        ; IRQ1 Handler
...             ...    ...             ;
0x1C32          jmp    SPM_RDY         ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PA is:

```
Address  Labels Code              Comments
.org 0x0002
0x0002          jmp    EXT_INT0        ; IRQ0 Handler
0x0004          jmp    EXT_INT1        ; IRQ1 Handler
...             ...    ...             ;
0x0032          jmp    SPM_RDY         ; Store Program Memory Ready Handler
;
.org 0x1C00
0x1C00   RESET: ldi    r16,high(RAMEND); Main program start
0x1C01          out    SPH,r16         ; Set Stack Pointer to top of RAM
0x1C02          ldi    r16,low(RAMEND)
0x1C03          out    SPL,r16
0x1C04          sei                    ; Enable interrupts
0x1C05          <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168PA is:

```
Address  Labels Code                  Comments
;
```

```
.org 0x1C00
0x1C00          jmp    RESET           ; Reset handler
0x1C02          jmp    EXT_INT0        ; IRQ0 Handler
0x1C04          jmp    EXT_INT1        ; IRQ1 Handler
...             ...    ...             ;
0x1C32          jmp    SPM_RDY         ; Store Program Memory Ready Handler
;
0x1C33   RESET: ldi    r16,high(RAMEND); Main program start
0x1C34          out    SPH,r16         ; Set Stack Pointer to top of RAM
0x1C35          ldi    r16,low(RAMEND)
0x1C36          out    SPL,r16
0x1C37          sei                    ; Enable interrupts
0x1C38          <instr>  xxx
```

11.4 Interrupt Vectors in ATmega328P

Table 11-6. Reset and Interrupt Vectors in ATmega328P

VectorNo.	Program Address(2)	Source	Interrupt Definition
1	0x0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Coutner1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete

Table 11-6. Reset and Interrupt Vectors in ATmega328P (Continued)

VectorNo.	Program Address(2)	Source	Interrupt Definition
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	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, ATmega88PA, ATmega168PA and ATmega328P" on page 277.
2. When the IVSEL bit in MCUCR 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 11-7 on page 66 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 11-7. Reset and Interrupt Vectors Placement in ATmega328P(1)

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

Note: 1. The Boot Reset Address is shown in Table 26-7 on page 289. 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 ATmega328P is:

```
Address  Labels Code                 Comments
0x0000         jmp    RESET          ; Reset Handler
0x0002         jmp    EXT_INT0       ; IRQ0 Handler
0x0004         jmp    EXT_INT1       ; IRQ1 Handler
0x0006         jmp    PCINT0         ; PCINT0 Handler
0x0008         jmp    PCINT1         ; PCINT1 Handler
0x000A         jmp    PCINT2         ; PCINT2 Handler
0x000C         jmp    WDT            ; Watchdog Timer Handler
0x000E         jmp    TIM2_COMPA     ; Timer2 Compare A Handler
0x0010         jmp    TIM2_COMPB     ; Timer2 Compare B Handler
0x0012         jmp    TIM2_OVF       ; Timer2 Overflow Handler
0x0014         jmp    TIM1_CAPT      ; Timer1 Capture Handler
0x0016         jmp    TIM1_COMPA     ; Timer1 Compare A Handler
0x0018         jmp    TIM1_COMPB     ; Timer1 Compare B Handler
0x001A         jmp    TIM1_OVF       ; Timer1 Overflow Handler
0x001C         jmp    TIM0_COMPA     ; Timer0 Compare A Handler
0x001E         jmp    TIM0_COMPB     ; Timer0 Compare B Handler
```

```
0x0020          jmp     TIM0_OVF        ; Timer0 Overflow Handler
0x0022          jmp     SPI_STC         ; SPI Transfer Complete Handler
0x0024          jmp     USART_RXC       ; USART, RX Complete Handler
0x0026          jmp     USART_UDRE      ; USART, UDR Empty Handler
0x0028          jmp     USART_TXC       ; USART, TX Complete Handler
0x002A          jmp     ADC             ; ADC Conversion Complete Handler
0x002C          jmp     EE_RDY          ; EEPROM Ready Handler
0x002E          jmp     ANA_COMP        ; Analog Comparator Handler
0x0030          jmp     TWI             ; 2-wire Serial Interface Handler
0x0032          jmp     SPM_RDY         ; Store Program Memory Ready Handler
;
0x0033RESET:    ldi     r16, high(RAMEND); Main program start
0x0034          out     SPH,r16         ; Set Stack Pointer to top of RAM
0x0035          ldi     r16, low(RAMEND)
0x0036          out     SPL,r16
0x0037          sei                     ; Enable interrupts
0x0038          <instr>  xxx
  ...     ...     ...  ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328P is:

```
Address  Labels Code                    Comments
0x0000   RESET: ldi     r16,high(RAMEND); Main program start
0x0001          out     SPH,r16         ; Set Stack Pointer to top of RAM
0x0002          ldi     r16,low(RAMEND)
0x0003          out     SPL,r16
0x0004          sei                     ; Enable interrupts
0x0005          <instr>  xxx
;
.org 0x3C02
0x3C02          jmp     EXT_INT0        ; IRQ0 Handler
0x3C04          jmp     EXT_INT1        ; IRQ1 Handler
...             ...     ...             ;
0x3C32          jmp     SPM_RDY         ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328P is:

```
Address  Labels Code                Comments
.org 0x0002
0x0002          jmp     EXT_INT0        ; IRQ0 Handler
0x0004          jmp     EXT_INT1        ; IRQ1 Handler
...             ...     ...             ;
0x0032          jmp     SPM_RDY         ; Store Program Memory Ready Handler
;
.org 0x3C00
0x3C00   RESET: ldi     r16,high(RAMEND); Main program start
```

```
0x3C01          out    SPH,r16         ; Set Stack Pointer to top of RAM
0x3C02          ldi    r16,low(RAMEND)
0x3C03          out    SPL,r16
0x3C04          sei                    ; Enable interrupts
0x3C05          <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega328P is:

```
Address  Labels Code                 Comments
;
.org 0x3C00
0x3C00          jmp    RESET           ; Reset handler
0x3C02          jmp    EXT_INT0        ; IRQ0 Handler
0x3C04          jmp    EXT_INT1        ; IRQ1 Handler
...             ...    ...             ;
0x3C32          jmp    SPM_RDY         ; Store Program Memory Ready Handler
;
0x3C33  RESET: ldi     r16,high(RAMEND); Main program start
0x3C34          out    SPH,r16         ; Set Stack Pointer to top of RAM
0x3C35          ldi    r16,low(RAMEND)
0x3C36          out    SPL,r16
0x3C37          sei                    ; Enable interrupts
0x3C38          <instr>  xxx
```

11.5 Register Description

11.5.1 Moving Interrupts Between Application and Boot Space, ATmega88PA, ATmega168PA and ATmega328P

The MCU Control Register controls the placement of the Interrupt Vector table.

11.5.2 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	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, ATmega88PA, ATmega168PA and ATmega328P” on page 277 for details. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit:

a. Write the Interrupt Vector Change Enable (IVCE) bit to one.
b. 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, ATmega88PA, ATmega168PA and ATmega328P" on page 277 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  MCUCR, r16
  ; Move interrupts to Boot Flash section
  ldi  r16, (1<<IVSEL)
  out  MCUCR, r16
  ret
```

C Code Example

```
void Move_interrupts(void)
{
  /* Enable change of Interrupt Vectors */
  MCUCR = (1<<IVCE);
  /* Move interrupts to Boot Flash section */
  MCUCR = (1<<IVSEL);
}
```

12. External Interrupts

The External Interrupts are triggered by the INT0 and INT1 pins or any of the PCINT23..0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0 and INT1 or PCINT23..0 pins are configured as outputs. This feature provides a way of generating a software interrupt. The pin change interrupt PCI2 will trigger if any enabled PCINT23..16 pin toggles. The pin change interrupt PCI1 will trigger if any enabled PCINT14..8 pin toggles. The pin change interrupt PCI0 will trigger if any enabled PCINT7..0 pin toggles. The PCMSK2, PCMSK1 and PCMSK0 Registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT23..0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

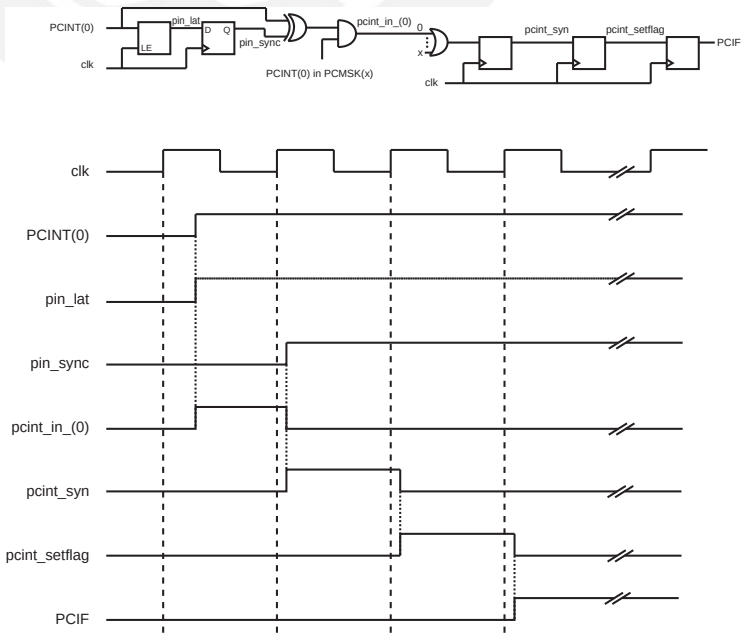
The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the External Interrupt Control Register A – EICRA. When the INT0 or INT1 interrupts are enabled and are configured as level triggered, the interrupts will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock, described in "Clock Systems and their Distribution" on page 26. Low level interrupt on INT0 and INT1 is detected asynchronously. This implies that this interrupt 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, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL Fuses as described in "System Clock and Clock Options" on page 26.

12.1 Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in Figure 12-1.

Figure 12-1. Timing of pin change interrupts



12.2 Register Description

12.2.1 EICRA – External Interrupt Control Register A

The External Interrupt Control Register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
(0x69)	–	–	–	–	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..4 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **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-flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 12-1. 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 12-1. 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 12-2. 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 12-2. 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.

12.2.2 EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	–	–	–	–	–	–	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 1 – 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 External Interrupt Control Register A (EICRA) 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 0 – 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 External Interrupt Control Register A (EICRA) 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.

12.2.3 EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	–	–	–	–	–	–	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 1 – 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 EIMSK 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 0 – 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 EIMSK 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.

12.2.4 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 - Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 - PCIE2: Pin Change Interrupt Enable 2**

When the PCIE2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT23..16 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI2 Interrupt Vector. PCINT23..16 pins are enabled individually by the PCMSK2 Register.

- **Bit 1 - PCIE1: Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT14..8 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI1 Interrupt Vector. PCINT14..8 pins are enabled individually by the PCMSK1 Register.

- **Bit 0 - PCIE0: Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7..0 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI0 Interrupt Vector. PCINT7..0 pins are enabled individually by the PCMSK0 Register.

12.2.5 PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 - Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 - PCIF2: Pin Change Interrupt Flag 2**

When a logic change on any PCINT23..16 pin triggers an interrupt request, PCIF2 becomes set (one). If the I-bit in SREG and the PCIE2 bit in PCICR 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.

- **Bit 1 - PCIF1: Pin Change Interrupt Flag 1**

When a logic change on any PCINT14..8 pin triggers an interrupt request, PCIF1 becomes set (one). If the I-bit in SREG and the PCIE1 bit in PCICR 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.

- **Bit 0 - PCIF0: Pin Change Interrupt Flag 0**

When a logic change on any PCINT7..0 pin triggers an interrupt request, PCIF0 becomes set (one). If the I-bit in SREG and the PCIE0 bit in PCICR 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.

12.2.6 PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
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..0 – PCINT23..16: Pin Change Enable Mask 23..16**

Each PCINT23..16-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is set and the PCIE2 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT23..16 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

12.2.7 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	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 – Res: Reserved Bit**

This bit is an unused bit in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 6..0 – PCINT14..8: Pin Change Enable Mask 14..8**

Each PCINT14..8-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT14..8 is set and the PCIE1 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT14..8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

12.2.8 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
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..0 – PCINT7..0: Pin Change Enable Mask 7..0**

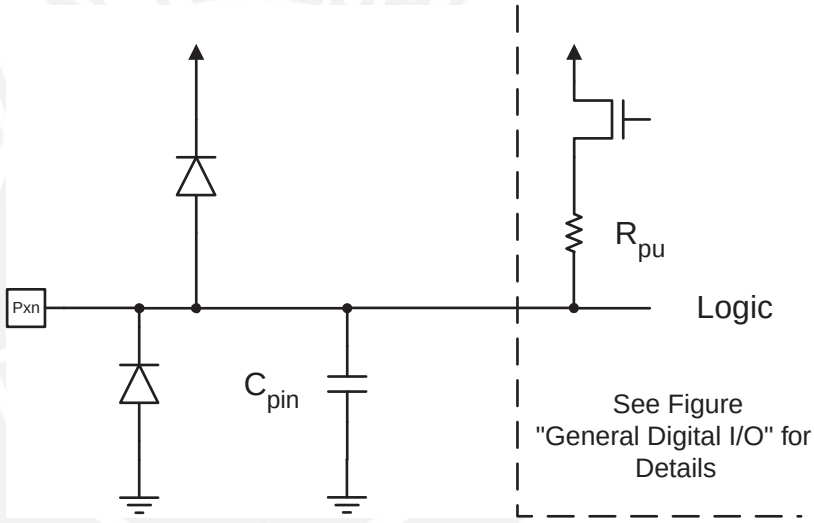
Each PCINT7..0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7..0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

13. I/O-Ports

13.1 Overview

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 13-1. Refer to ”Electrical Characteristics” on page 313 for a complete list of parameters.

Figure 13-1. 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. For example, 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” on page 92.

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. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

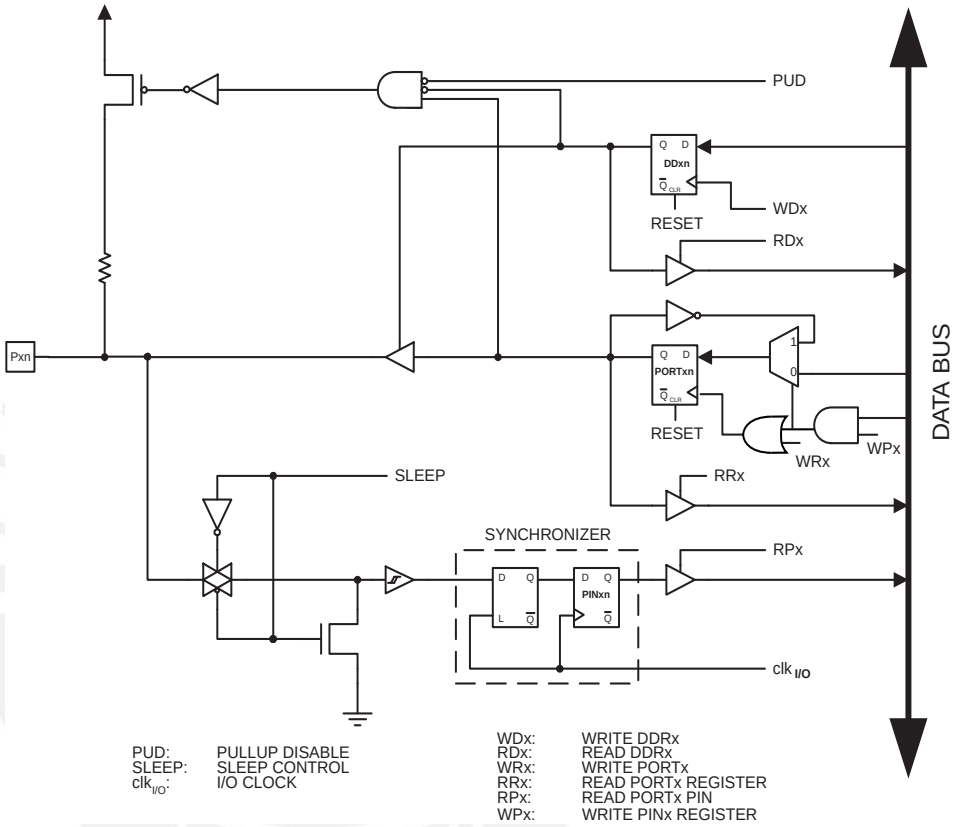
Using the I/O port as General Digital I/O is described in ”Ports as General Digital I/O” on page 76. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in ”Alternate Port Functions” on page 80. 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.

13.2 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 13-2 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 13-2. General Digital I/O(1)



Note: 1. WRx, 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.

13.2.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in ”Register Description” on page 92, 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 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).

13.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

13.2.3 Switching Between Input and Output

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-impedance 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 MCUCR 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 13-1 summarizes the control signals for the pin value.

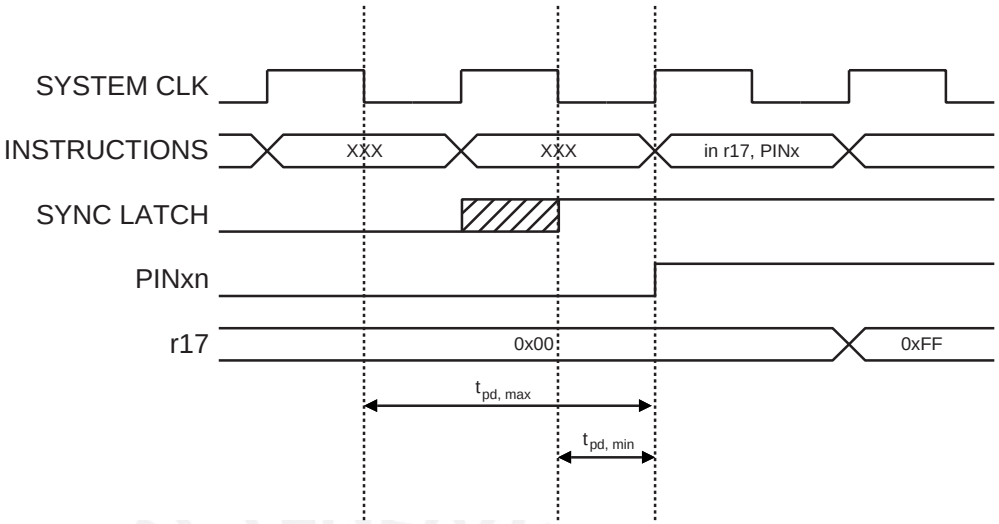
Table 13-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	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)

13.2.4 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 13-2, 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 13-3 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.

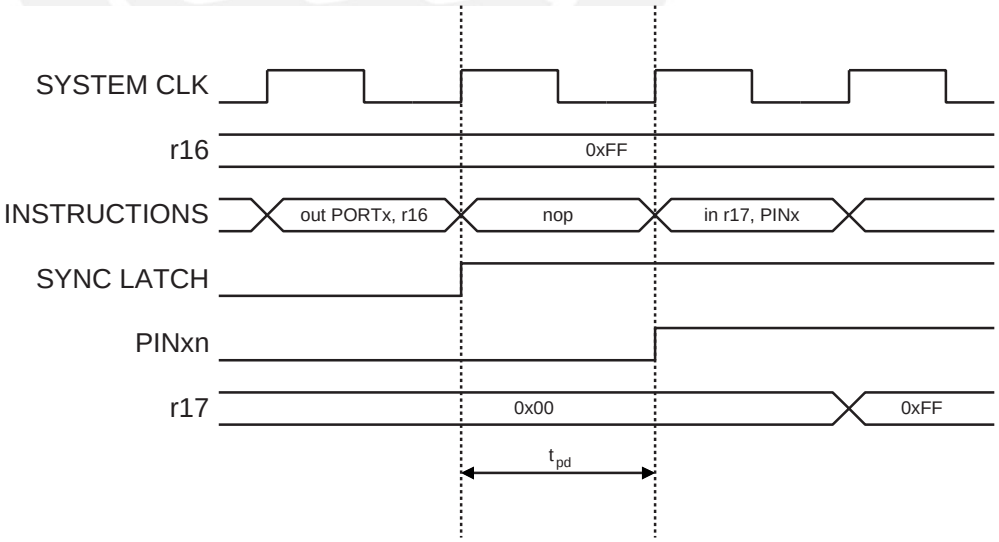
Figure 13-3. Synchronization when Reading an Externally Applied Pin value



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 tpd,max and tpd,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 13-4. The out instruction sets the “SYNC LATCH” signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is 1 system clock period.

Figure 13-4. 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

```
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*/
__no_operation();
/* 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.

13.2.5 Digital Input Enable and Sleep Modes

As shown in Figure 13-2, 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, and 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 80.

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 mode, as the clamping in these sleep mode produces the requested logic change.

13.2.6 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, float-

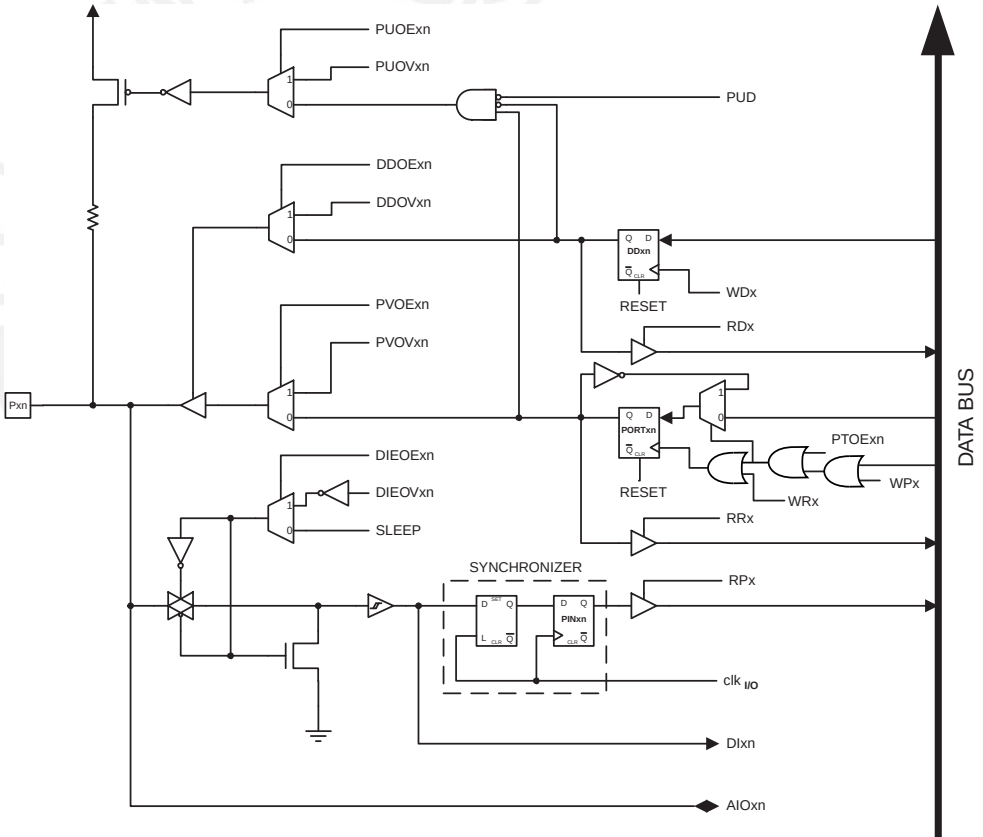
ing inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

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.

13.3 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. Figure 13-5 shows how the port pin control signals from the simplified Figure 13-2 on page 76 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 13-5. 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	WRx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE	WPx:	WRITE PINx
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE	$clk_{I/O}$:	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	DIGITAL INPUT PIN n ON PORTx
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Note: 1. WRx, 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 13-2 summarizes the function of the overriding signals. The pin and port indexes from Figure 13-5 on page 80 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 13-2. 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.
PTOE	Port Toggle Override Enable	If PTOE is set, the PORTxn Register bit is inverted.
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 mode).
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 mode).
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.

13.3.1 Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 13-3.

Table 13-3. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	$\overline{SS}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

The alternate pin configuration is as follows:

- **XTAL2/TOSC2/PCINT7 – Port B, Bit 7**

XTAL2: Chip clock Oscillator pin 2. Used as clock pin for crystal Oscillator or Low-frequency crystal Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC2: Timer Oscillator pin 2. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the AS2 bit in ASSR is set (one) and the EXCLK bit is cleared (zero) to enable asynchronous clocking of Timer/Counter2 using the Crystal Oscillator, pin PB7 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 cannot be used as an I/O pin.

PCINT7: Pin Change Interrupt source 7. The PB7 pin can serve as an external interrupt source.

If PB7 is used as a clock pin, DDB7, PORTB7 and PINB7 will all read 0.

- **XTAL1/TOSC1/PCINT6 – Port B, Bit 6**

XTAL1: Chip clock Oscillator pin 1. Used for all chip clock sources except internal calibrated RC Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

TOSC1: Timer Oscillator pin 1. Used only if internal calibrated RC Oscillator is selected as chip clock source, and the asynchronous timer is enabled by the correct setting in ASSR. When the

AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pin PB6 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.

PCINT6: Pin Change Interrupt source 6. The PB6 pin can serve as an external interrupt source.

If PB6 is used as a clock pin, DDB6, PORTB6 and PINB6 will all read 0.

- **SCK/PCINT5 – Port B, Bit 5**

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 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.

PCINT5: Pin Change Interrupt source 5. The PB5 pin can serve as an external interrupt source.

- **MISO/PCINT4 – Port B, Bit 4**

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 DDB4. When the SPI is enabled as a Slave, 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.

PCINT4: Pin Change Interrupt source 4. The PB4 pin can serve as an external interrupt source.

- **MOSI/OC2/PCINT3 – Port B, Bit 3**

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 DDB3. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB3. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB3 bit.

OC2, Output Compare Match Output: The PB3 pin can serve as an external output for the Timer/Counter2 Compare Match. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC2 pin is also the output pin for the PWM mode timer function.

PCINT3: Pin Change Interrupt source 3. The PB3 pin can serve as an external interrupt source.

- **$\overline{SS}$/OC1B/PCINT2 – Port B, Bit 2**

$\overline{SS}$: Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB2. 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 DDB2. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB2 bit.

OC1B, Output Compare Match output: The PB2 pin can serve as an external output for the Timer/Counter1 Compare Match B. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

PCINT2: Pin Change Interrupt source 2. The PB2 pin can serve as an external interrupt source.

- **OC1A/PCINT1 – Port B, Bit 1**

OC1A, Output Compare Match output: The PB1 pin can serve as an external output for the Timer/Counter1 Compare Match A. The PB1 pin has to be configured as an output (DDB1 set

(one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

PCINT1: Pin Change Interrupt source 1. The PB1 pin can serve as an external interrupt source.

- **ICP1/CLKO/PCINT0 – Port B, Bit 0**

ICP1, Input Capture Pin: The PB0 pin can act as an Input Capture Pin for Timer/Counter1.

CLKO, Divided System Clock: The divided system clock can be output on the PB0 pin. The divided system clock will be output if the CKOUT Fuse is programmed, regardless of the PORTB0 and DDB0 settings. It will also be output during reset.

PCINT0: Pin Change Interrupt source 0. The PB0 pin can serve as an external interrupt source.

Table 13-4 and Table 13-5 on page 85 relate the alternate functions of Port B to the overriding signals shown in Figure 13-5 on page 80. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 13-4. Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/XTAL2/ TOSC2/PCINT7(1)	PB6/XTAL1/ TOSC1/PCINT6(1)	PB5/SCK/ PCINT5	PB4/MISO/ PCINT4
PUOE	$\overline{\text{INTRC}}$ • $\overline{\text{EXTCK}}$+ AS2	$\overline{\text{INTRC}}$ + AS2	SPE • $\overline{\text{MSTR}}$	SPE • MSTR
PUOV	0	0	PORTB5 • $\overline{\text{PUD}}$	PORTB4 • $\overline{\text{PUD}}$
DDOE	$\overline{\text{INTRC}}$ • $\overline{\text{EXTCK}}$+ AS2	$\overline{\text{INTRC}}$ + AS2	SPE • $\overline{\text{MSTR}}$	SPE • MSTR
DDOV	0	0	0	0
PVOE	0	0	SPE • MSTR	SPE • $\overline{\text{MSTR}}$
PVOV	0	0	SCK OUTPUT	SPI SLAVE OUTPUT
DIEOE	$\overline{\text{INTRC}}$ • $\overline{\text{EXTCK}}$ + AS2 + PCINT7 • PCIE0	$\overline{\text{INTRC}}$ + AS2 + PCINT6 • PCIE0	PCINT5 • PCIE0	PCINT4 • PCIE0
DIEOV	(INTRC + EXTCK) • $\overline{\text{AS2}}$	INTRC • $\overline{\text{AS2}}$	1	1
DI	PCINT7 INPUT	PCINT6 INPUT	PCINT5 INPUT SCK INPUT	PCINT4 INPUT SPI MSTR INPUT
AIO	Oscillator Output	Oscillator/Clock Input	–	–

Notes: 1. INTRC means that one of the internal RC Oscillators are selected (by the CKSEL fuses), EXTCK means that external clock is selected (by the CKSEL fuses)

Table 13-5. Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/MOSI/ OC2/PCINT3	PB2/$\overline{SS}$/ OC1B/PCINT2	PB1/OC1A/ PCINT1	PB0/ICP1/ PCINT0
PUOE	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	0	0
PUOV	PORTB3 • $\overline{PUD}$	PORTB2 • $\overline{PUD}$	0	0
DDOE	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	0	0
DDOV	0	0	0	0
PVOE	SPE • MSTR + OC2A ENABLE	OC1B ENABLE	OC1A ENABLE	0
PVOV	SPI MSTR OUTPUT + OC2A	OC1B	OC1A	0
DIEOE	PCINT3 • PCIE0	PCINT2 • PCIE0	PCINT1 • PCIE0	PCINT0 • PCIE0
DIEOV	1	1	1	1
DI	PCINT3 INPUT SPI SLAVE INPUT	PCINT2 INPUT SPI $\overline{SS}$	PCINT1 INPUT	PCINT0 INPUT ICP1 INPUT
AIO	–	–	–	–

13.3.2 Alternate Functions of Port C

The Port C pins with alternate functions are shown in Table 13-6.

Table 13-6. Port C Pins Alternate Functions

Port Pin	Alternate Function
PC6	$\overline{RESET}$ (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

The alternate pin configuration is as follows:

- **$\overline{\text{RESET}}$/PCINT14 – Port C, Bit 6**

$\overline{\text{RESET}}$, Reset pin: When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on Power-on Reset and Brown-out Reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin.

If PC6 is used as a reset pin, DDC6, PORTC6 and PINC6 will all read 0.

PCINT14: Pin Change Interrupt source 14. The PC6 pin can serve as an external interrupt source.

- **SCL/ADC5/PCINT13 – Port C, Bit 5**

SCL, 2-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PC5 is disconnected from the port and becomes the Serial Clock I/O pin for the 2-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.

PC5 can also be used as ADC input Channel 5. Note that ADC input channel 5 uses digital power.

PCINT13: Pin Change Interrupt source 13. The PC5 pin can serve as an external interrupt source.

- **SDA/ADC4/PCINT12 – Port C, Bit 4**

SDA, 2-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PC4 is disconnected from the port and becomes the Serial Data I/O pin for the 2-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.

PC4 can also be used as ADC input Channel 4. Note that ADC input channel 4 uses digital power.

PCINT12: Pin Change Interrupt source 12. The PC4 pin can serve as an external interrupt source.

- **ADC3/PCINT11 – Port C, Bit 3**

PC3 can also be used as ADC input Channel 3. Note that ADC input channel 3 uses analog power.

PCINT11: Pin Change Interrupt source 11. The PC3 pin can serve as an external interrupt source.

- **ADC2/PCINT10 – Port C, Bit 2**

PC2 can also be used as ADC input Channel 2. Note that ADC input channel 2 uses analog power.

PCINT10: Pin Change Interrupt source 10. The PC2 pin can serve as an external interrupt source.

- **ADC1/PCINT9 – Port C, Bit 1**

PC1 can also be used as ADC input Channel 1. Note that ADC input channel 1 uses analog power.

PCINT9: Pin Change Interrupt source 9. The PC1 pin can serve as an external interrupt source.

- **ADC0/PCINT8 – Port C, Bit 0**

PC0 can also be used as ADC input Channel 0. Note that ADC input channel 0 uses analog power.

PCINT8: Pin Change Interrupt source 8. The PC0 pin can serve as an external interrupt source.

Table 13-7 and Table 13-8 relate the alternate functions of Port C to the overriding signals shown in Figure 13-5 on page 80.

Table 13-7. Overriding Signals for Alternate Functions in PC6..PC4[1]

Signal Name	PC6/$\overline{\text{RESET}}$/PCINT14	PC5/SCL/ADC5/PCINT13	PC4/SDA/ADC4/PCINT12
PUOE	RSTDISBL	TWEN	TWEN
PUOV	1	PORTC5 • $\overline{\text{PUD}}$	PORTC4 • $\overline{\text{PUD}}$
DDOE	RSTDISBL	TWEN	TWEN
DDOV	0	SCL_OUT	SDA_OUT
PVOE	0	TWEN	TWEN
PVOV	0	0	0
DIEOE	RSTDISBL + PCINT14 • PCIE1	PCINT13 • PCIE1 + ADC5D	PCINT12 • PCIE1 + ADC4D
DIEOV	RSTDISBL	PCINT13 • PCIE1	PCINT12 • PCIE1
DI	PCINT14 INPUT	PCINT13 INPUT	PCINT12 INPUT
AIO	RESET INPUT	ADC5 INPUT / SCL INPUT	ADC4 INPUT / SDA INPUT

Note: 1. When enabled, the 2-wire Serial Interface enables slew-rate controls on the output pins PC4 and PC5. 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.

Table 13-8. Overriding Signals for Alternate Functions in PC3..PC0

Signal Name	PC3/ADC3/ PCINT11	PC2/ADC2/ PCINT10	PC1/ADC1/ PCINT9	PC0/ADC0/ PCINT8
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	PCINT11 • PCIE1 + ADC3D	PCINT10 • PCIE1 + ADC2D	PCINT9 • PCIE1 + ADC1D	PCINT8 • PCIE1 + ADC0D
DIEOV	PCINT11 • PCIE1	PCINT10 • PCIE1	PCINT9 • PCIE1	PCINT8 • PCIE1
DI	PCINT11 INPUT	PCINT10 INPUT	PCINT9 INPUT	PCINT8 INPUT
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

13.3.3 Alternate Functions of Port D

The Port D pins with alternate functions are shown in Table 13-9.

Table 13-9. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

The alternate pin configuration is as follows:

- **AIN1/OC2B/PCINT23 – Port D, Bit 7**

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.

PCINT23: Pin Change Interrupt source 23. The PD7 pin can serve as an external interrupt source.

- **AIN0/OC0A/PCINT22 – Port D, Bit 6**

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.

OC0A, Output Compare Match output: The PD6 pin can serve as an external output for the Timer/Counter0 Compare Match A. The PD6 pin has to be configured as an output (DDD6 set (one)) to serve this function. The OC0A pin is also the output pin for the PWM mode timer function.

PCINT22: Pin Change Interrupt source 22. The PD6 pin can serve as an external interrupt source.

- **T1/OC0B/PCINT21 – Port D, Bit 5**

T1, Timer/Counter1 counter source.

OC0B, Output Compare Match output: The PD5 pin can serve as an external output for the Timer/Counter0 Compare Match B. The PD5 pin has to be configured as an output (DDD5 set (one)) to serve this function. The OC0B pin is also the output pin for the PWM mode timer function.

PCINT21: Pin Change Interrupt source 21. The PD5 pin can serve as an external interrupt source.

- **XCK/T0/PCINT20 – Port D, Bit 4**

XCK, USART external clock.

T0, Timer/Counter0 counter source.

PCINT20: Pin Change Interrupt source 20. The PD4 pin can serve as an external interrupt source.

- **INT1/OC2B/PCINT19 – Port D, Bit 3**

INT1, External Interrupt source 1: The PD3 pin can serve as an external interrupt source.

OC2B, Output Compare Match output: The PD3 pin can serve as an external output for the Timer/Counter0 Compare Match B. The PD3 pin has to be configured as an output (DDD3 set (one)) to serve this function. The OC2B pin is also the output pin for the PWM mode timer function.

PCINT19: Pin Change Interrupt source 19. The PD3 pin can serve as an external interrupt source.

- **INT0/PCINT18 – Port D, Bit 2**

INT0, External Interrupt source 0: The PD2 pin can serve as an external interrupt source.

PCINT18: Pin Change Interrupt source 18. The PD2 pin can serve as an external interrupt source.

- **TXD/PCINT17 – 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.

PCINT17: Pin Change Interrupt source 17. The PD1 pin can serve as an external interrupt source.

- **RXD/PCINT16 – 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.

PCINT16: Pin Change Interrupt source 16. The PD0 pin can serve as an external interrupt source.

Table 13-10 and Table 13-11 relate the alternate functions of Port D to the overriding signals shown in Figure 13-5 on page 80.

Table 13-10. Overriding Signals for Alternate Functions PD7..PD4

Signal Name	PD7/AIN1 /PCINT23	PD6/AIN0/ OC0A/PCINT22	PD5/T1/OC0B/ PCINT21	PD4/XCK/ T0/PCINT20
PUOE	0	0	0	0
PUO	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	OC0A ENABLE	OC0B ENABLE	UMSEL
PVOV	0	OC0A	OC0B	XCK OUTPUT
DIEOE	PCINT23 • PCIE2	PCINT22 • PCIE2	PCINT21 • PCIE2	PCINT20 • PCIE2
DIEOV	1	1	1	1
DI	PCINT23 INPUT	PCINT22 INPUT	PCINT21 INPUT T1 INPUT	PCINT20 INPUT XCK INPUT T0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	–	–

Table 13-11. Overriding Signals for Alternate Functions in PD3..PD0

Signal Name	PD3/OC2B/INT1/ PCINT19	PD2/INT0/ PCINT18	PD1/TXD/ PCINT17	PD0/RXD/ PCINT16
PUOE	0	0	TXEN	RXEN
PUO	0	0	0	PORTD0 • $\overline{PUD}$
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	OC2B ENABLE	0	TXEN	0
PVOV	OC2B	0	TXD	0
DIEOE	INT1 ENABLE + PCINT19 • PCIE2	INT0 ENABLE + PCINT18 • PCIE1	PCINT17 • PCIE2	PCINT16 • PCIE2
DIEOV	1	1	1	1
DI	PCINT19 INPUT INT1 INPUT	PCINT18 INPUT INT0 INPUT	PCINT17 INPUT	PCINT16 INPUT RXD
AIO	–	–	–	–

13.4 Register Description

13.4.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 4 – 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 76 for more details about this feature.

13.4.2 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	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	

13.4.3 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	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	

13.4.4 PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	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	

13.4.5 PORTC – The Port C Data Register

Bit	7	6	5	4	3	2	1	0	
0x08 (0x28)	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.6 DDRC – The Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x07 (0x27)	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.7 PINC – The Port C Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x06 (0x26)	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

13.4.8 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	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	

13.4.9 DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	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	

13.4.10 PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	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	

14. 8-bit Timer/Counter0 with PWM

14.1 Features

- **Two Independent Output Compare Units**
- **Double Buffered Output Compare Registers**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch Free, Phase Correct Pulse Width Modulator (PWM)**
- **Variable PWM Period**
- **Frequency Generator**
- **Three Independent Interrupt Sources (TOV0, OCF0A, and OCF0B)**

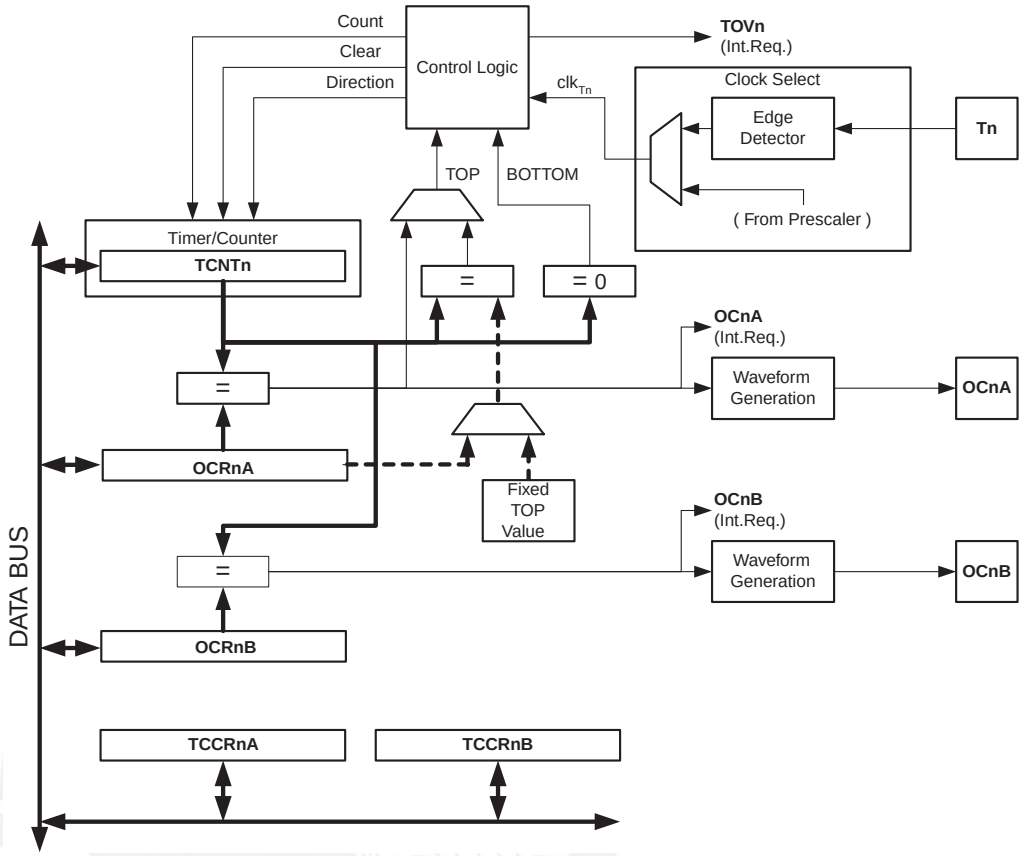
14.2 Overview

Timer/Counter0 is a general purpose 8-bit Timer/Counter module, with two independent Output Compare Units, and with PWM support. It allows accurate program execution timing (event management) and wave generation.

A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 14-1. For the actual placement of I/O pins, refer to ”Pinout ATmega48PA/88PA/168PA/328P” 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 ”Register Description” on page 106.

The PRTIM0 bit in ”Minimizing Power Consumption” on page 42 must be written to zero to enable Timer/Counter0 module.

Figure 14-1. 8-bit Timer/Counter Block Diagram



14.2.1 Definitions

Many register and bit references in this section are written in general form. A lower case “n” replaces the Timer/Counter number, in this case 0. A lower case “x” replaces the Output Compare Unit, in this case Compare Unit A or Compare Unit B. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions in Table 14-1 are also used extensively throughout the document.

Table 14-1. 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 OCR0A Register. The assignment is dependent on the mode of operation.

14.2.2 Registers

The Timer/Counter (TCNT0) and Output Compare Registers (OCR0A and OCR0B) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in the figure) signals are all visible in the Timer Interrupt Flag Register (TIFR0). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK0). TIFR0 and TIMSK0 are not shown in the figure.

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 Registers (OCR0A and OCR0B) are 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 pins (OC0A and OC0B). See Section “15.7.3” on page 123. for details. The compare match event will also set the Compare Flag (OCF0A or OCF0B) which can be used to generate an Output Compare interrupt request.

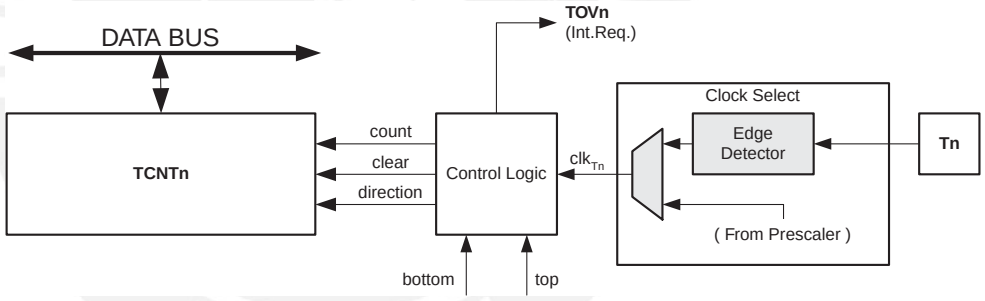
14.3 Timer/Counter Clock Sources

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 (TCCR0B). For details on clock sources and prescaler, see ”Timer/Counter0 and Timer/Counter1 Prescalers” on page 141.

14.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 14-2 shows a block diagram of the counter and its surroundings.

Figure 14-2. 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 (TCCR0A) and the WGM02 bit located in the Timer/Counter Control Register B (TCCR0B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC0A and OC0B. For more details about advanced counting sequences and waveform generation, see ”Modes of Operation” on page 99.

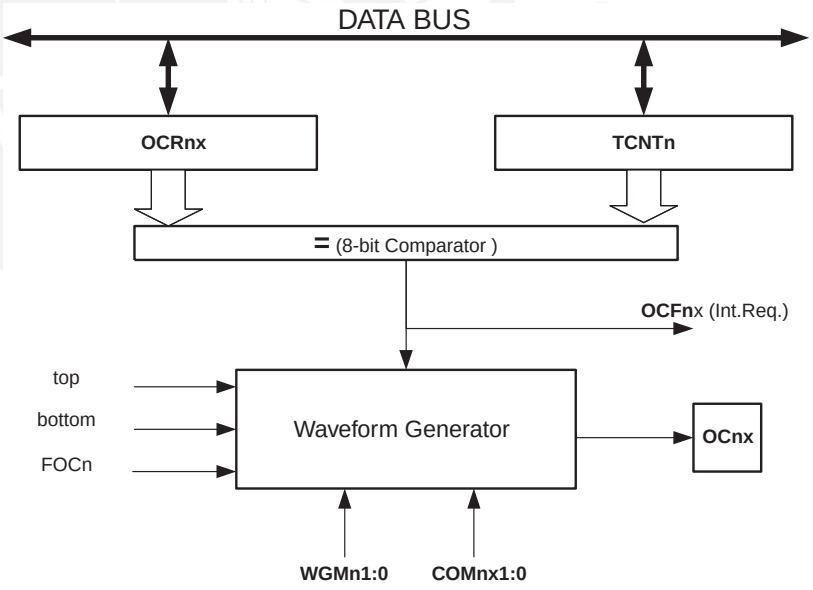
The Timer/Counter Overflow Flag (TOV0) is set according to the mode of operation selected by the WGM02:0 bits. TOV0 can be used for generating a CPU interrupt.

14.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Registers (OCR0A and OCR0B). Whenever TCNT0 equals OCR0A or OCR0B, the comparator signals a match. A match will set the Output Compare Flag (OCF0A or OCF0B) at the next timer clock cycle. If the corresponding interrupt is enabled, the Output Compare Flag generates an Output Compare interrupt. The Output Compare Flag is automatically cleared when the interrupt is executed. Alternatively, the 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 WGM02:0 bits and Compare Output mode (COM0x1:0) 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 99).

Figure 14-3 shows a block diagram of the Output Compare unit.

Figure 14-3. Output Compare Unit, Block Diagram



The OCR0x Registers are 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 OCR0x Compare Registers 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 OCR0x Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR0x Buffer Register, and if double buffering is disabled the CPU will access the OCR0x directly.

14.5.1 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 (FOC0x) bit. Forcing compare match will not set the OCF0x Flag or reload/clear the timer, but the OC0x pin will be updated as if a real compare match had occurred (the COM0x1:0 bits settings define whether the OC0x pin is set, cleared or toggled).

14.5.2 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 OCR0x to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

14.5.3 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 OCR0x 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.

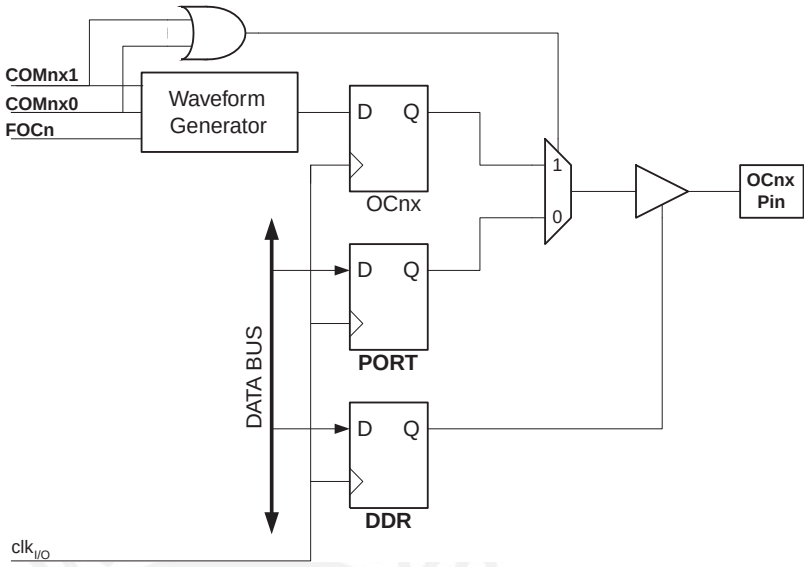
The setup of the OC0x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC0x value is to use the Force Output Compare (FOC0x) strobe bits in Normal mode. The OC0x Registers keep their values even when changing between Waveform Generation modes.

Be aware that the COM0x1:0 bits are not double buffered together with the compare value. Changing the COM0x1:0 bits will take effect immediately.

14.6 Compare Match Output Unit

The Compare Output mode (COM0x1:0) bits have two functions. The Waveform Generator uses the COM0x1:0 bits for defining the Output Compare (OC0x) state at the next compare match. Also, the COM0x1:0 bits control the OC0x pin output source. Figure 14-4 shows a simplified schematic of the logic affected by the COM0x1:0 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 COM0x1:0 bits are shown. When referring to the OC0x state, the reference is for the internal OC0x Register, not the OC0x pin. If a system reset occur, the OC0x Register is reset to “0”.

Figure 14-4. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC0x) from the Waveform Generator if either of the COM0x1:0 bits are set. However, the OC0x 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 OC0x pin (DDR_OC0x) must be set as output before the OC0x 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 OC0x state before the output is enabled. Note that some COM0x1:0 bit settings are reserved for certain modes of operation. See Section “14.9” on page 106.

14.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM0x1:0 bits differently in Normal, CTC, and PWM modes. For all modes, setting the COM0x1:0 = 0 tells the Waveform Generator that no action on the OC0x Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 14-2 on page 106. For fast PWM mode, refer to Table 14-3 on page 106, and for phase correct PWM refer to Table 14-4 on page 107.

A change of the COM0x1:0 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 FOC0x strobe bits.

14.7 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM02:0) and Compare Output mode (COM0x1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM0x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM0x1:0 bits control whether the output should be set, cleared, or toggled at a compare match (See Section “14.6” on page 98.).

For detailed timing information refer to ”Timer/Counter Timing Diagrams” on page 104.

14.7.1 Normal Mode

The simplest mode of operation is the Normal mode (WGM02:0 = 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.

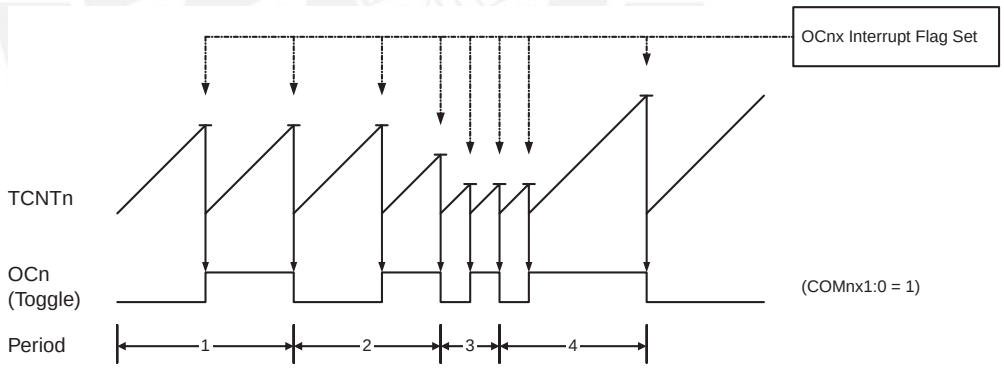
The Output Compare unit can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

14.7.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM02:0 = 2), the OCR0A Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT0) matches the OCR0A. The OCR0A 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 14-5. The counter value (TCNT0) increases until a compare match occurs between TCNT0 and OCR0A, and then counter (TCNT0) is cleared.

Figure 14-5. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0A 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 OCR0A 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 OC0A output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM0A1:0 = 1). The OC0A 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 OCR0A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

The *N* variable represents the prescale factor (1, 8, 64, 256, or 1024).

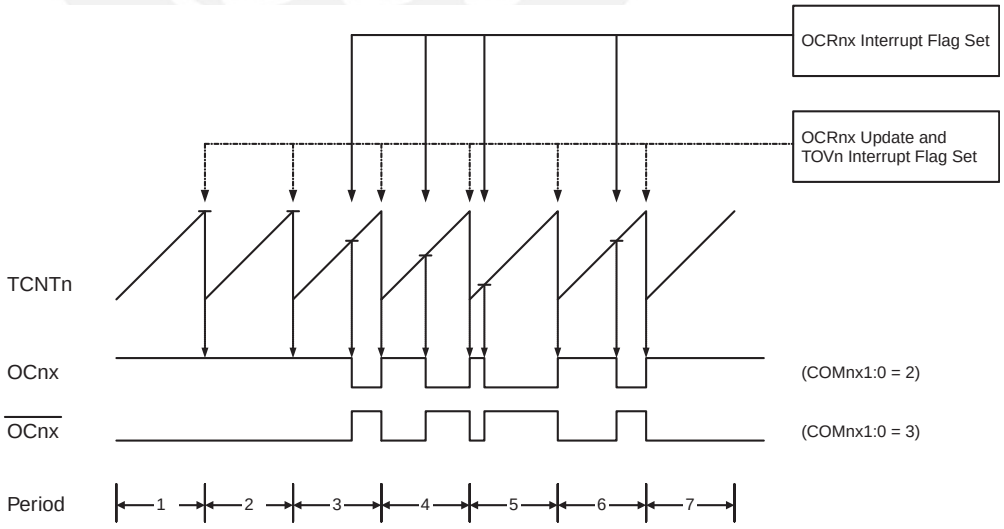
As for the Normal mode of operation, the TOV0 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

14.7.3 Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM02:0 = 3 or 7) 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 TOP then restarts from BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR0A when WGM2:0 = 7. In non-inverting Compare Output mode, the Output Compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x, 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 TOP value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 14-6. 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 OCR0x and TCNT0.

Figure 14-6. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches TOP. 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 OC0x pins. Setting the COM0x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM0x1:0 to three: Setting the COM0A1:0 bits to one allows the OC0A pin to toggle on Compare Matches if the WGM02 bit is set. This option is not available for the OC0B pin (see Table 14-6 on page 107). The actual OC0x 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 OC0x Register at the compare match between OCR0x and TCNT0, and clearing (or setting) the OC0x 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:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

The *N* variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0A Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR0A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM0A1:0 bits.)

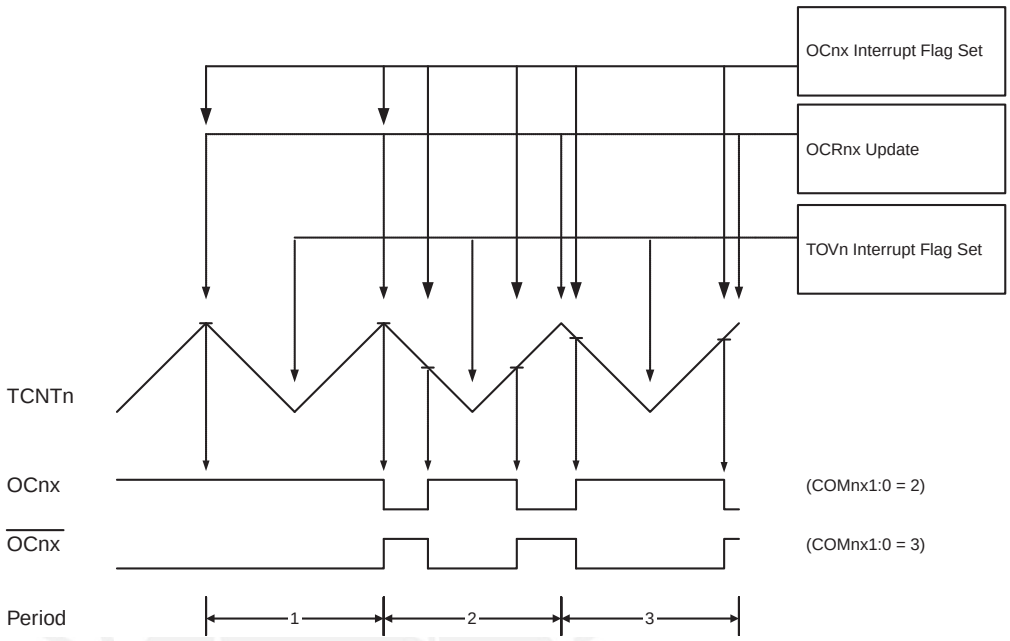
A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC0x to toggle its logical level on each compare match (COM0x1:0 = 1). The waveform generated will have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0A is set to zero. This feature is similar to the OC0A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

14.7.4 Phase Correct PWM Mode

The phase correct PWM mode (WGM02:0 = 1 or 5) 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 TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 1, and OCR0A when WGM2:0 = 5. In non-inverting Compare Output mode, the Output Compare (OC0x) is cleared on the compare match between TCNT0 and OCR0x 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.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT0 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 14-7. 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 OCR0x and TCNT0.

Figure 14-7. 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 OC0x pins. Setting the COM0x1:0 bits to two will produce a non-inverted PWM. An inverted PWM output can be generated by setting the COM0x1:0 to three: Setting the COM0A0 bits to one allows the OC0A pin to toggle on Compare Matches if the WGM02 bit is set. This option is not available for the OC0B pin (see Table 14-7 on page 108). The actual OC0x 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 OC0x Register at the compare match between OCR0x and TCNT0 when the counter increments, and setting (or clearing) the OC0x Register at compare match between OCR0x and TCNT0 when the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$

The N variable represents the prescale factor (1, 8, 64, 256, or 1024).

The extreme values for the OCR0A Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR0A 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 14-7 OCnx 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.

- OCRnx changes its value from MAX, like in Figure 14-7. 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 OCnx value at MAX must correspond to the result of an up-counting Compare Match.

- The timer starts counting from a value higher than the one in OCRnx, and for that reason misses the Compare Match and hence the OCnx change that would have happened on the way up.

14.8 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 14-8 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 14-8. Timer/Counter Timing Diagram, no Prescaling

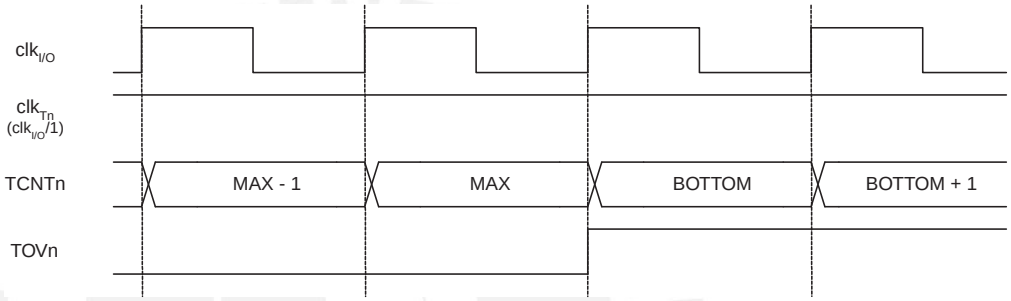


Figure 14-9 shows the same timing data, but with the prescaler enabled.

Figure 14-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

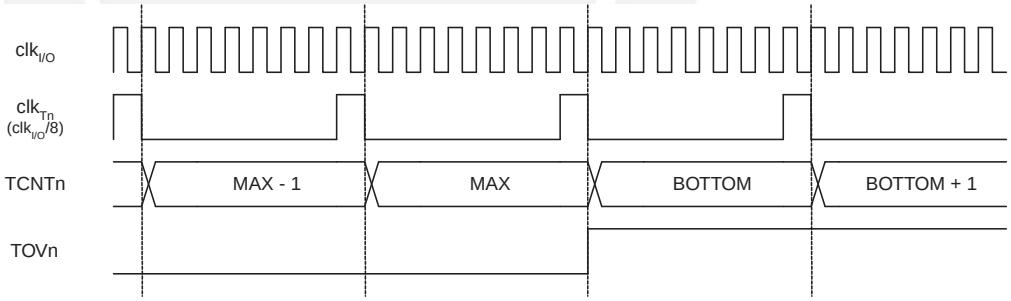


Figure 14-10 shows the setting of OCF0B in all modes and OCF0A in all modes except CTC mode and PWM mode, where OCR0A is TOP.

Figure 14-10. Timer/Counter Timing Diagram, Setting of OCF0x, with Prescaler ($f_{clk\_I/O}/8$)

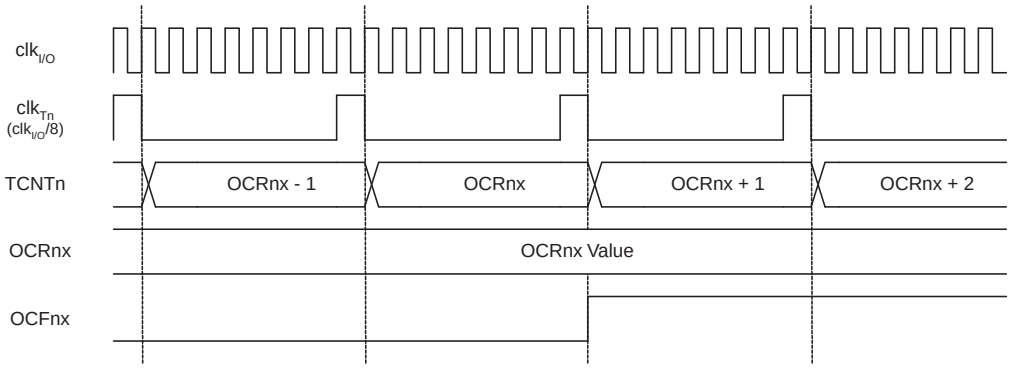
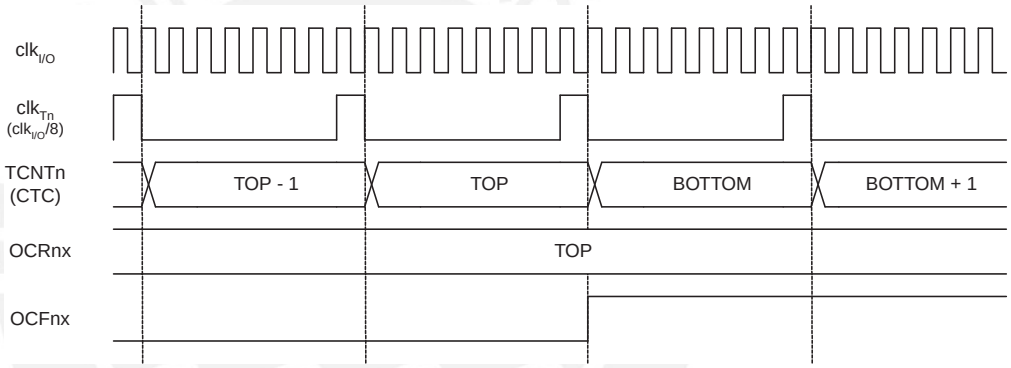


Figure 14-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

Figure 14-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk\_I/O}/8$)



14.9 Register Description

14.9.1 TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM0A1:0: Compare Match Output A Mode**

These bits control the Output Compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A 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 OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 14-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-2. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

Table 14-3 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 14-3. Compare Output Mode, Fast PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 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 101 for more details.

Table 14-4 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 14-4. Compare Output Mode, Phase Correct PWM Mode(1)

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 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 128 for more details.

- **Bits 5:4 – COM0B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B 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 OC0B pin must be set in order to enable the output driver.

When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. Table 14-5 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-5. Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match
1	0	Clear OC0B on Compare Match
1	1	Set OC0B on Compare Match

Table 14-6 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to fast PWM mode.

Table 14-6. Compare Output Mode, Fast PWM Mode(1)

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match, set OC0B at BOTTOM, (non-inverting mode)
1	1	Set OC0B on Compare Match, clear OC0B at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 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 101 for more details.

Table 14-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 14-7. Compare Output Mode, Phase Correct PWM Mode(1)

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on Compare Match when up-counting. Set OC0B on Compare Match when down-counting.
1	1	Set OC0B on Compare Match when up-counting. Clear OC0B on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 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 102 for more details.

- **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bits 1:0 – WGM01:0: Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B 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 14-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see "Modes of Operation" on page 99).

Table 14-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

14.9.2 TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0A: Force Output Compare A**

The FOC0A 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 TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare.

A FOC0A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0A as TOP.

The FOC0A bit is always read as zero.

- **Bit 6 – FOC0B: Force Output Compare B**

The FOC0B 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 TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0B bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0B output is changed according to its COM0B1:0 bits setting. Note that the FOC0B bit is implemented as a strobe. Therefore it is the value present in the COM0B1:0 bits that determines the effect of the forced compare.

A FOC0B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0B as TOP.

The FOC0B bit is always read as zero.

- **Bits 5:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 3 – WGM02: Waveform Generation Mode**

See the description in the ”TCCR0A – Timer/Counter Control Register A” on page 106.

- **Bits 2:0 – CS02:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter.

Table 14-9. 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.

14.9.3 TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	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 OCR0x Registers.

14.9.4 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
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 A 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 OC0A pin.

14.9.5 OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x28 (0x48)	OCR0B[7:0]								OCR0B
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 B 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 OC0B pin.

14.9.6 TIMSK0 – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 2 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter Compare Match B interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter Interrupt Flag Register – TIFR0.

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

- **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, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

14.9.7 TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 2 – OCF0B: Timer/Counter 0 Output Compare B Match Flag**

The OCF0B bit is set when a Compare Match occurs between the Timer/Counter and the data in OCR0B – Output Compare Register0 B. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter Compare B Match Interrupt Enable), and OCF0B are set, the Timer/Counter Compare Match Interrupt is executed.

- **Bit 1 – OCF0A: Timer/Counter 0 Output Compare A Match Flag**

The OCF0A bit is set when a Compare Match occurs between the Timer/Counter0 and the data in OCR0A – Output Compare Register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to

the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 Compare Match Interrupt Enable), and OCF0A are set, the Timer/Counter0 Compare Match Interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set 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, the Timer/Counter0 Overflow interrupt is executed.

The setting of this flag is dependent of the WGM02:0 bit setting. Refer to Table 14-8, ”Waveform Generation Mode Bit Description” on page 108.

15. 16-bit Timer/Counter1 with PWM

15.1 Features

- **True 16-bit Design (i.e., 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)**

15.2 Overview

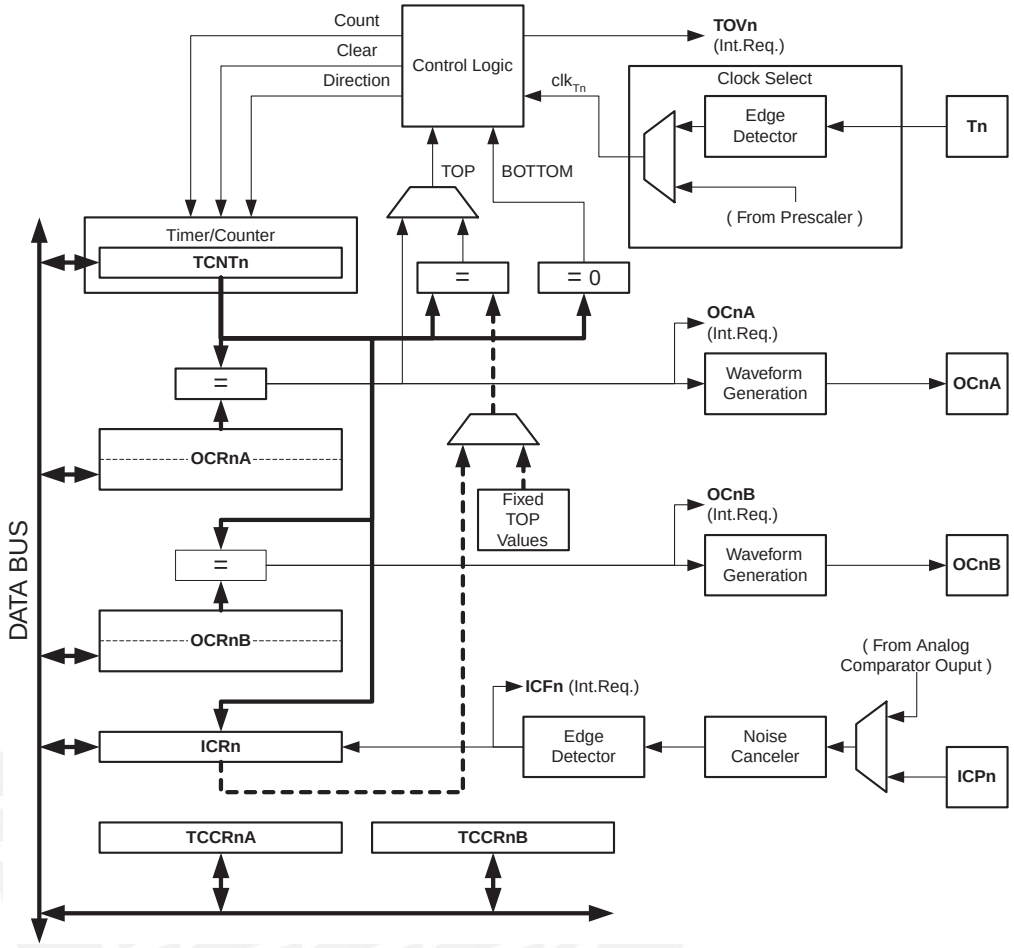
The 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement.

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 channel. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT1 for accessing Timer/Counter1 counter value and so on.

A simplified block diagram of the 16-bit Timer/Counter is shown in Figure 15-1. For the actual placement of I/O pins, refer to ”Pinout ATmega48PA/88PA/168PA/328P” 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 ”Register Description” on page 134.

The PRTIM1 bit in ”PRR – Power Reduction Register” on page 45 must be written to zero to enable Timer/Counter1 module.

Figure 15-1. 16-bit Timer/Counter Block Diagram[1]



Note: 1. Refer to Figure 1-1 on page 2, Table 13-3 on page 82 and Table 13-9 on page 88 for Timer/Counter1 pin placement and description.

15.2.1 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 115. 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* (TIFR1). All interrupts are individually masked with the *Timer Interrupt Mask Register* (TIMSK1). TIFR1 and TIMSK1 are not shown in the figure.

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 "Out-

put Compare Units" on page 122. 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 246) 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.

15.2.2 Definitions

The following definitions are used extensively throughout the section:

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.

15.3 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 Examples[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 Examples[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.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

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.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

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.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

The assembly code example requires that the r17:r16 register pair contains the value to be written to TCNT1.

15.3.1 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.

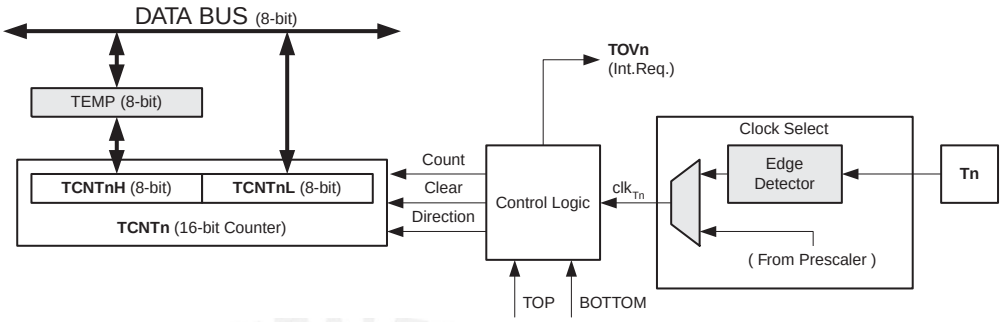
15.4 Timer/Counter Clock Sources

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 141.

15.5 Counter Unit

The main part of the 16-bit Timer/Counter is the programmable 16-bit bi-directional counter unit. Figure 15-2 shows a block diagram of the counter and its surroundings.

Figure 15-2. 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 eight 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 125.

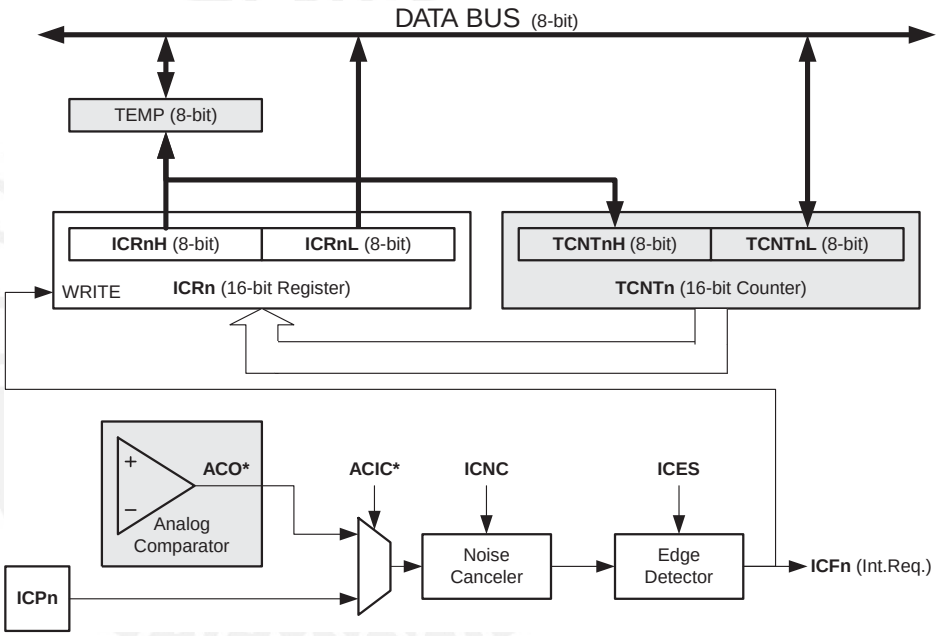
The Timer/Counter Overflow Flag (TOV1) is set according to the mode of operation selected by the WGM13:0 bits. TOV1 can be used for generating a CPU interrupt.

15.6 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 15-3. 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 15-3. 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 (ICIE1 = 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 Genera-*

tion 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 115.

15.6.1 Input Capture Trigger 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 16-1 on page 141). 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.

15.6.2 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.

15.6.3 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).

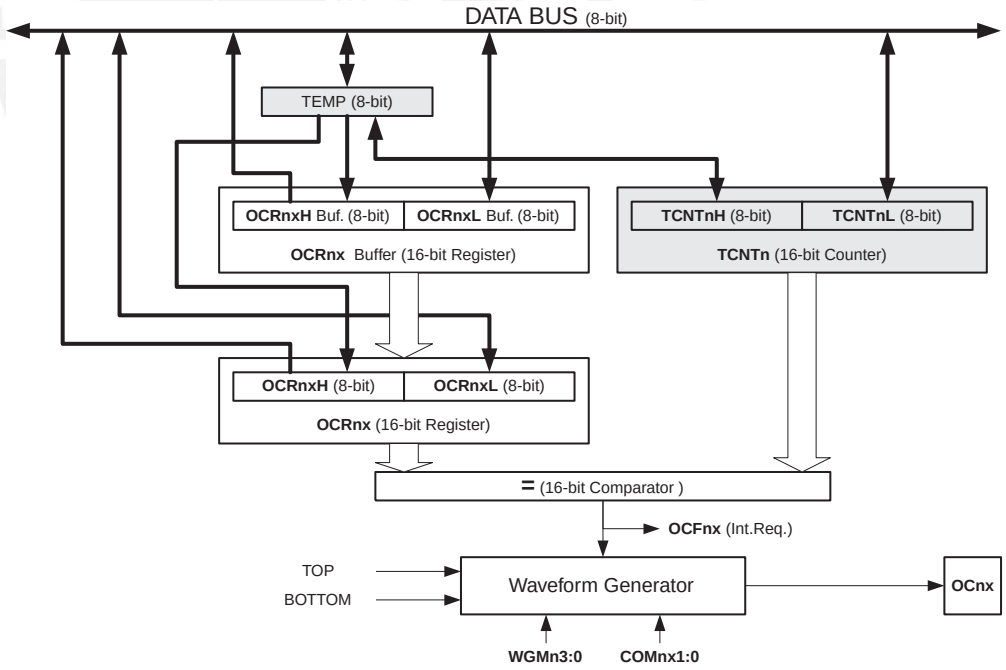
15.7 Output Compare Units

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 Section “15.9” on page 125.)

A special feature of Output Compare unit A allows it to define the Timer/Counter TOP value (i.e., counter resolution). In addition to the counter resolution, the TOP value defines the period time for waveforms generated by the Waveform Generator.

Figure 15-4 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/Counter 1), 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 15-4. 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 115.

15.7.1 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 COM11:0 bits settings define whether the OC1x pin is set, cleared or toggled).

15.7.2 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.

15.7.3 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 channels, 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.

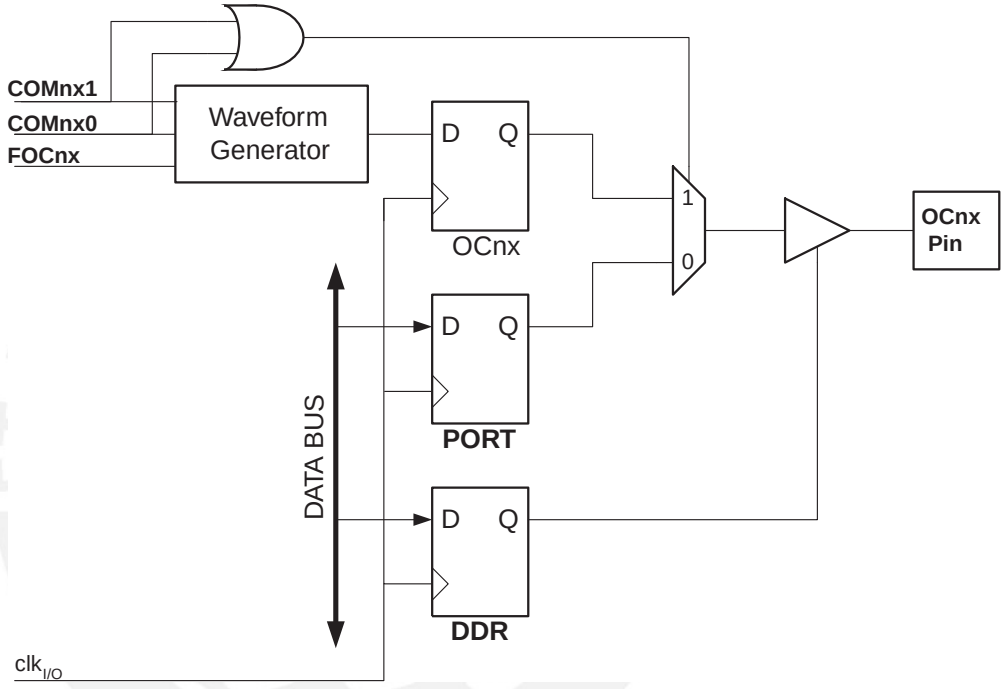
The setup of the OC1x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC1x value is to use the Force Output Compare (FOC1x) strobe bits in Normal mode. The OC1x Register keeps its value even when changing between Waveform Generation 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.

15.8 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 15-5 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 15-5. 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 15-1, Table 15-2 and Table 15-3 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 Section “15.11” on page 134.

The COM1x1:0 bits have no effect on the Input Capture unit.

15.8.1 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 15-1 on page 134. For fast PWM mode refer to Table 15-2 on page 135, and for phase correct and phase and frequency correct PWM refer to Table 15-3 on page 135.

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.

15.9 Modes of Operation

The mode of operation, i.e., 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 Section “15.8” on page 124.)

For detailed timing information refer to "Timer/Counter Timing Diagrams” on page 132.

15.9.1 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.

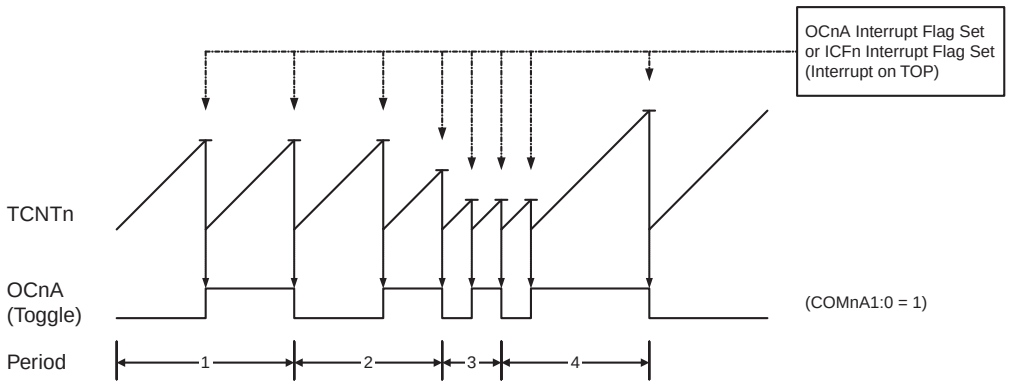
The Output Compare units can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

15.9.2 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 15-6. The counter value (TCNT1) increases until a compare match occurs with either OCR1A or ICR1, and then counter (TCNT1) is cleared.

Figure 15-6. 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.

15.9.3 Fast PWM Mode

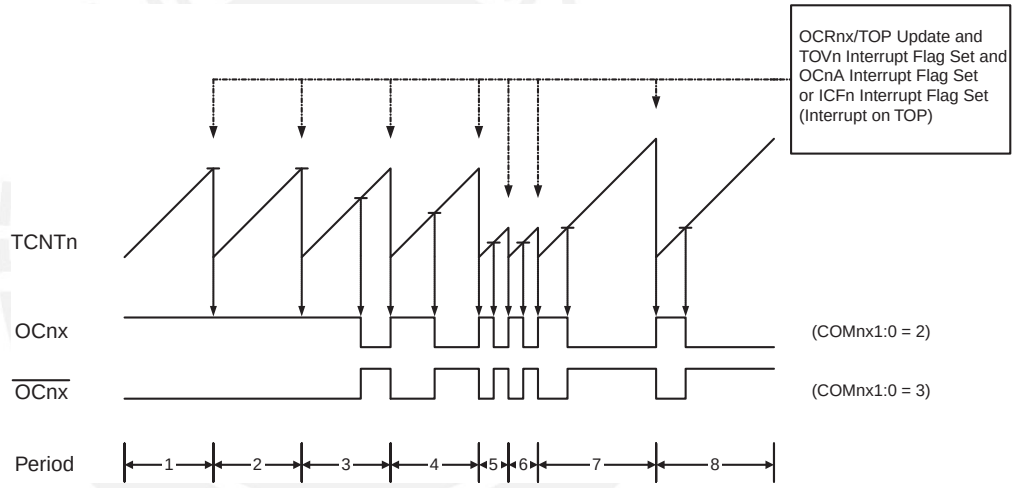
The *fast Pulse Width Modulation* or fast PWM mode (WGM13:0 = 5, 6, 7, 14, or 15) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM options by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. In non-inverting 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-, 9-, 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 15-7. 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 15-7. 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 two will produce a inverted PWM and an non-inverted PWM output can be generated by setting the COM1x1:0 to three (see Table on page 135). 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.

15.9.4 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.

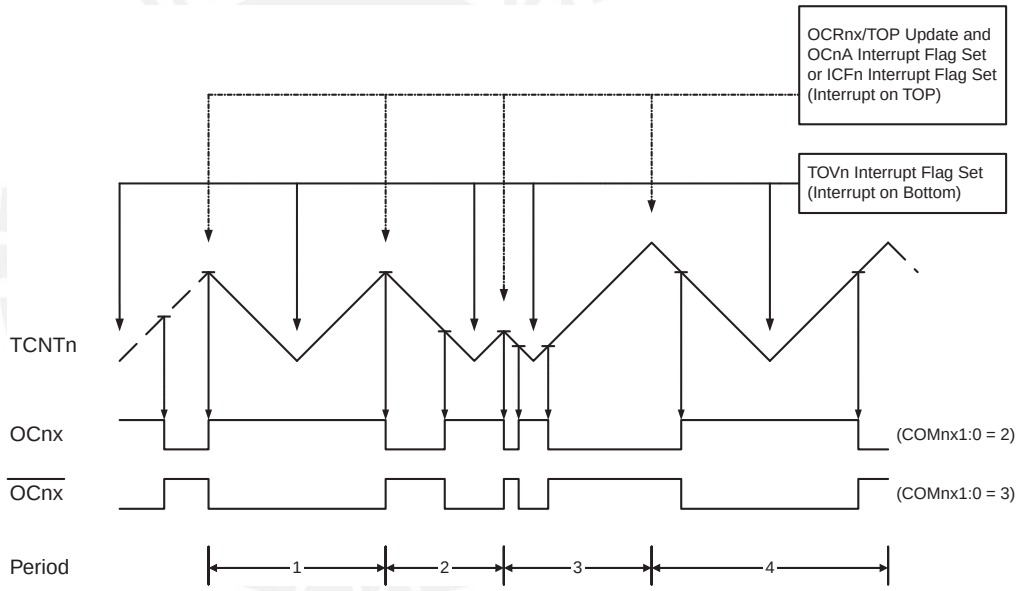
The PWM resolution for the phase correct PWM mode can be fixed to 8-, 9-, or 10-bit, or defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to

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 15-8. 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 15-8. 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 15-8 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 two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to three (See Table on page 135). 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.

15.9.5 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 15-8 and Figure 15-9).

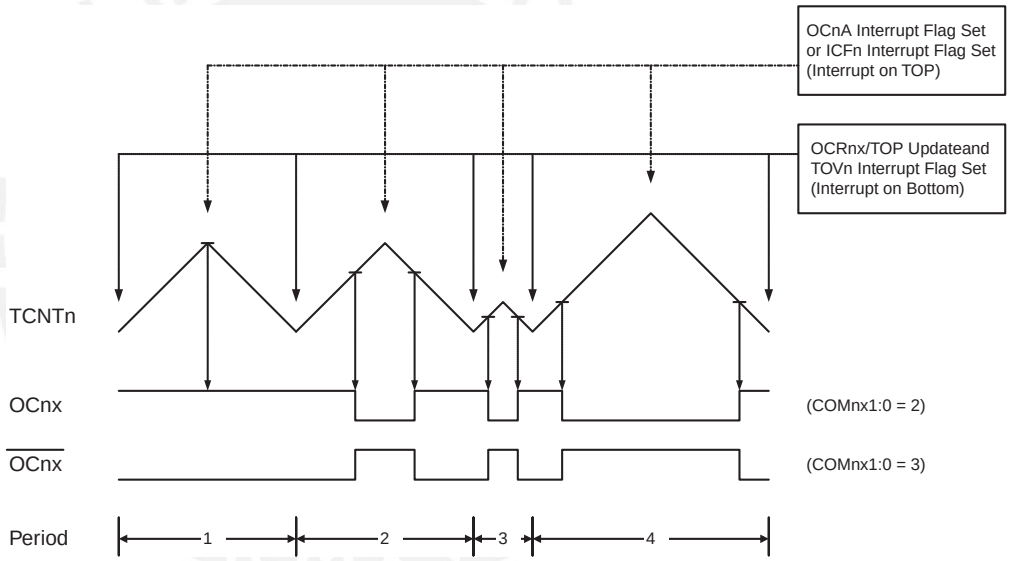
The PWM resolution for the phase and frequency correct PWM mode can be defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and

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 15-9. 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 15-9. 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 15-9 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 two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to three (See Table on page 135). 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.

15.10 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 15-10 shows a timing diagram for the setting of OCF1x.

Figure 15-10. Timer/Counter Timing Diagram, Setting of OCF1x, no Prescaling

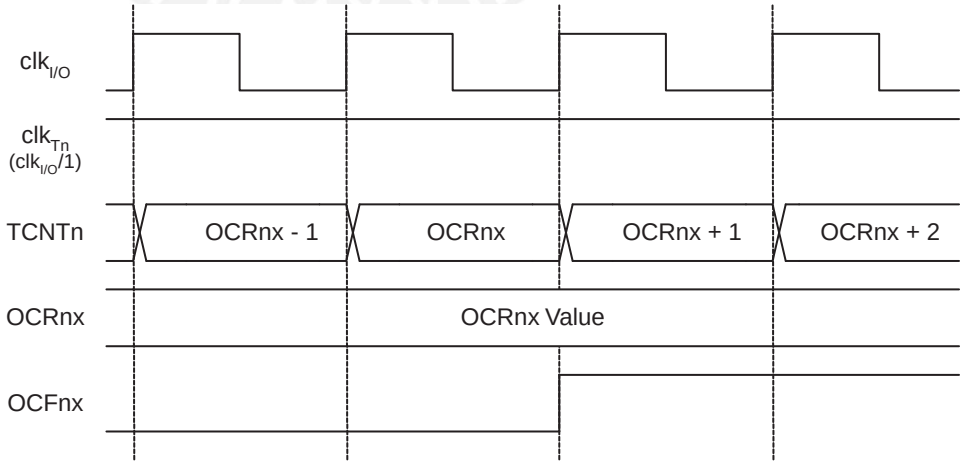


Figure 15-11 shows the same timing data, but with the prescaler enabled.

Figure 15-11. Timer/Counter Timing Diagram, Setting of OCF1x, with Prescaler ($f_{clk\_I/O}/8$)

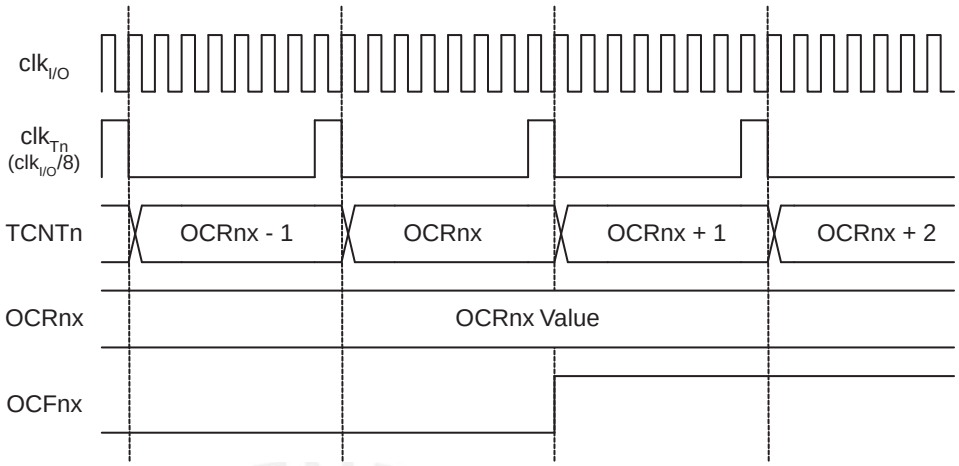


Figure 15-12 shows the count sequence close to TOP in various modes. When using phase and frequency correct PWM mode the OCR1x Register is updated at BOTTOM. The timing diagrams will be the same, but TOP should be replaced by BOTTOM, TOP-1 by BOTTOM+1 and so on. The same renaming applies for modes that set the TOV1 Flag at BOTTOM.

Figure 15-12. Timer/Counter Timing Diagram, no Prescaling

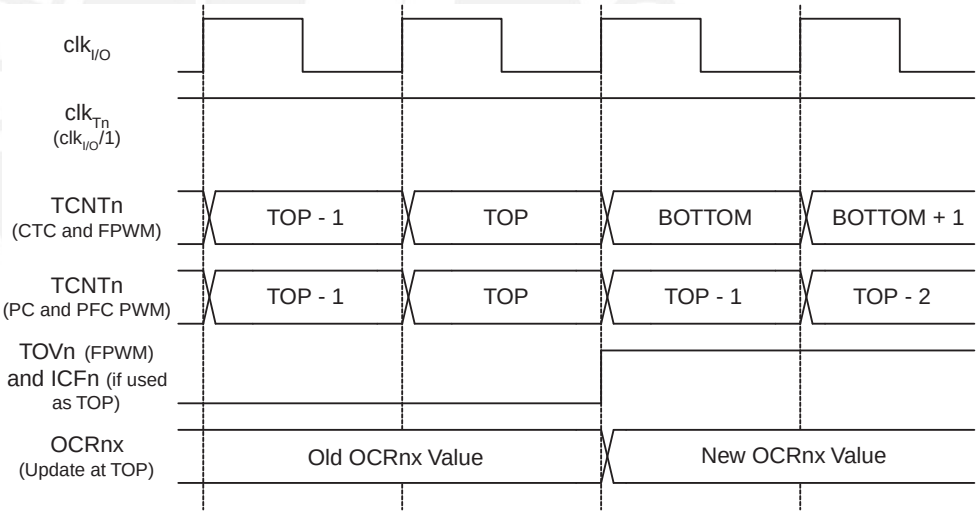
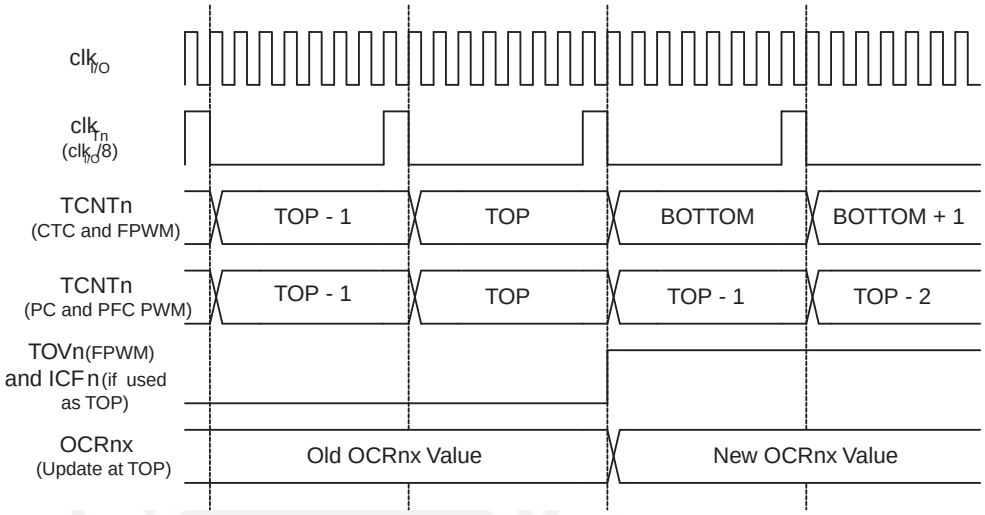


Figure 15-13 shows the same timing data, but with the prescaler enabled.

Figure 15-13. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)



15.11 Register Description

15.11.1 TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	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 15-1 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a Normal or a CTC mode (non-PWM).

Table 15-1. 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 15-2 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

Table 15-2. Compare Output Mode, Fast PWM(1)

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B 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 Section “15.9.3” on page 126. for more details.

Table 15-3 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 15-3. 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 11: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 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 Section “15.9.4” on page 128. for more details.

- **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 15-4. 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 Section “15.9” on page 125.).

Table 15-4. Waveform Generation Mode Bit Description[1]

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	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.

15.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	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 15-10 and Figure 15-11.

Table 15-5. 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.

15.11.3 TCCR1C – Timer/Counter1 Control Register C

Bit	7	6	5	4	3	2	1	0	
(0x82)	FOC1A	FOC1B	–	–	–	–	–	–	TCCR1C
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC1A: Force Output Compare for Channel A**

- **Bit 6 – FOC1B: Force Output Compare for Channel B**

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-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.

15.11.4 TCNT1H and TCNT1L – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	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 Section “15.3” on page 115.

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.

15.11.5 OCR1AH and OCR1AL – Output Compare Register 1 A

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	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	

15.11.6 OCR1BH and OCR1BL – Output Compare Register 1 B

Bit	7	6	5	4	3	2	1	0	
(0x8B)	OCR1B[15:8]								OCR1BH
(0x8A)	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 Section “15.3” on page 115.

15.11.7 ICR1H and ICR1L – Input Capture Register 1

Bit	7	6	5	4	3	2	1	0	
(0x87)	ICR1[15:8]								ICR1H
(0x86)	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 Section “15.3” on page 115.

15.11.8 TIMSK1 – Timer/Counter1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 5 – ICIE1: 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 57) is executed when the ICF1 Flag, located in TIFR1, is set.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 – 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 57) is executed when the OCF1B Flag, located in TIFR1, is set.

- **Bit 1 – 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 57) is executed when the OCF1A Flag, located in TIFR1, is set.

- **Bit 0 – 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 57) is executed when the TOV1 Flag, located in TIFR1, is set.

15.11.9 TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **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, 3 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 2 – 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 1 – 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 0 – 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 15-4 on page 136 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.

16. Timer/Counter0 and Timer/Counter1 Prescalers

"8-bit Timer/Counter0 with PWM" on page 94 and "16-bit Timer/Counter1 with PWM" on page 113 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.

16.1 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 $f_{CLK\_I/O}/1024$.

16.2 Prescaler Reset

The prescaler is free running, i.e., 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).

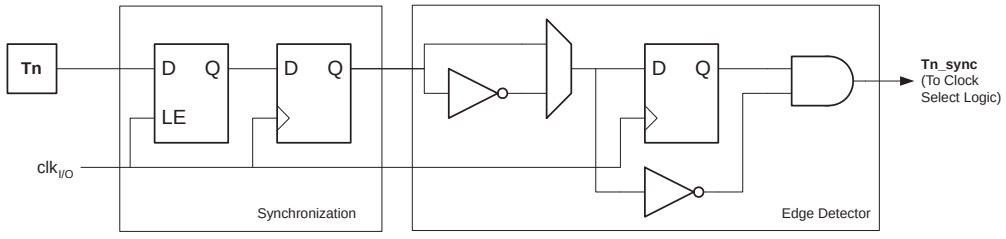
It is possible to use the prescaler reset for synchronizing the Timer/Counter to program execution. However, care must be taken if the other Timer/Counter that shares the same prescaler also uses prescaling. A prescaler reset will affect the prescaler period for all Timer/Counters it is connected to.

16.3 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 16-1 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 16-1. T1/T0 Pin Sampling



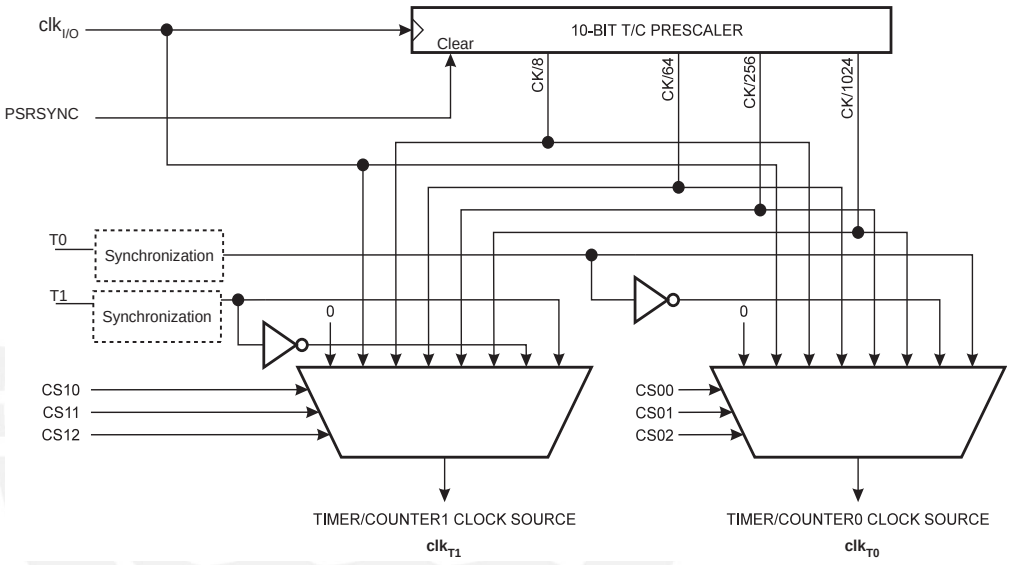
The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1/T0 pin to the counter is updated.

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 16-2. Prescaler for Timer/Counter0 and Timer/Counter1[(1)]



Note: 1. The synchronization logic on the input pins (T1/T0) is shown in Figure 16-1.

16.4 Register Description

16.4.1 GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	TSM	–	–	–	–	–	PSRASY	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – TSM: Timer/Counter Synchronization Mode**

Writing the TSM bit to one activates the Timer/Counter Synchronization mode. In this mode, the value that is written to the PSRASY and PSRSYNC bits is kept, hence keeping the corresponding prescaler reset signals asserted. This ensures that the corresponding Timer/Counters are halted and can be configured to the same value without the risk of one of them advancing during configuration. When the TSM bit is written to zero, the PSRASY and PSRSYNC bits are cleared by hardware, and the Timer/Counters start counting simultaneously.

- **Bit 0 – PSRSYNC: Prescaler Reset**

When this bit is one, Timer/Counter1 and Timer/Counter0 prescaler will be Reset. This bit is normally cleared immediately by hardware, except if the TSM bit is set. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a reset of this prescaler will affect both timers.

17. 8-bit Timer/Counter2 with PWM and Asynchronous Operation

17.1 Features

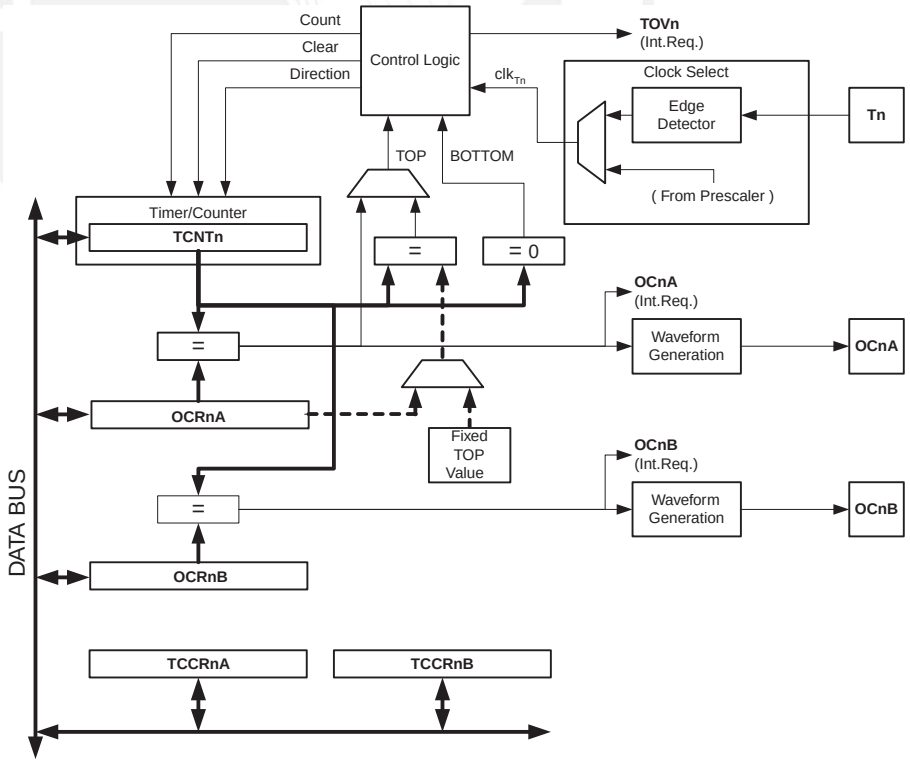
- **Single Channel Counter**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch-free, Phase Correct Pulse Width Modulator (PWM)**
- **Frequency Generator**
- **10-bit Clock Prescaler**
- **Overflow and Compare Match Interrupt Sources (TOV2, OCF2A and OCF2B)**
- **Allows Clocking from External 32 kHz Watch Crystal Independent of the I/O Clock**

17.2 Overview

Timer/Counter2 is a general purpose, single channel, 8-bit Timer/Counter module. A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 17-1. For the actual placement of I/O pins, refer to ”Pinout ATmega48PA/88PA/168PA/328P” 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 ”Register Description” on page 158.

The PRTIM2 bit in ”Minimizing Power Consumption” on page 42 must be written to zero to enable Timer/Counter2 module.

Figure 17-1. 8-bit Timer/Counter Block Diagram



17.2.1 Registers

The Timer/Counter (TCNT2) and Output Compare Register (OCR2A and OCR2B) are 8-bit registers. Interrupt request (shorten as Int.Req.) signals are all visible in the Timer Interrupt Flag Register (TIFR2). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK2). TIFR2 and TIMSK2 are not shown in the figure.

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 he 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 (OCR2A and OCR2B) are 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 pins (OC2A and OC2B). See Section “17.5” on page 146. for details. The compare match event will also set the Compare Flag (OCF2A or OCF2B) which can be used to generate an Output Compare interrupt request.

17.2.2 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, i.e., TCNT2 for accessing Timer/Counter2 counter value and so on.

The definitions in Table 17-1 are also used extensively throughout the section.

Table 17-1. 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 OCR2A Register. The assignment is dependent on the mode of operation.

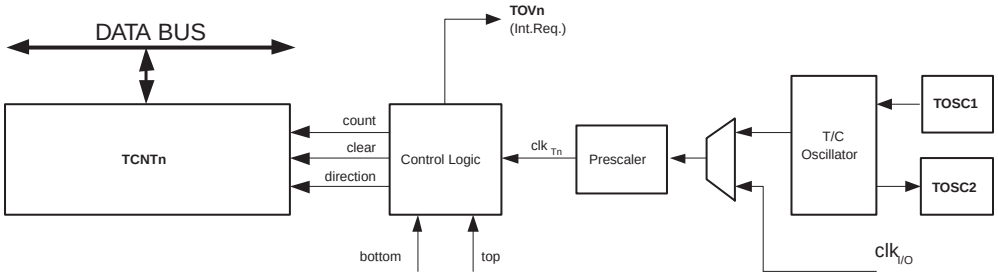
17.3 Timer/Counter Clock Sources

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 ”ASSR – Asynchronous Status Register” on page 164. For details on clock sources and prescaler, see ”Timer/Counter Prescaler” on page 156.

17.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 17-2 on page 146 shows a block diagram of the counter and its surrounding environment.

Figure 17-2. 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_{Tn}	Timer/Counter clock, referred to as clk_{T2} in the following.
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 (TCCR2A) and the WGM22 located in the Timer/Counter Control Register B (TCCR2B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC2A and OC2B. For more details about advanced counting sequences and waveform generation, see ”Modes of Operation” on page 149.

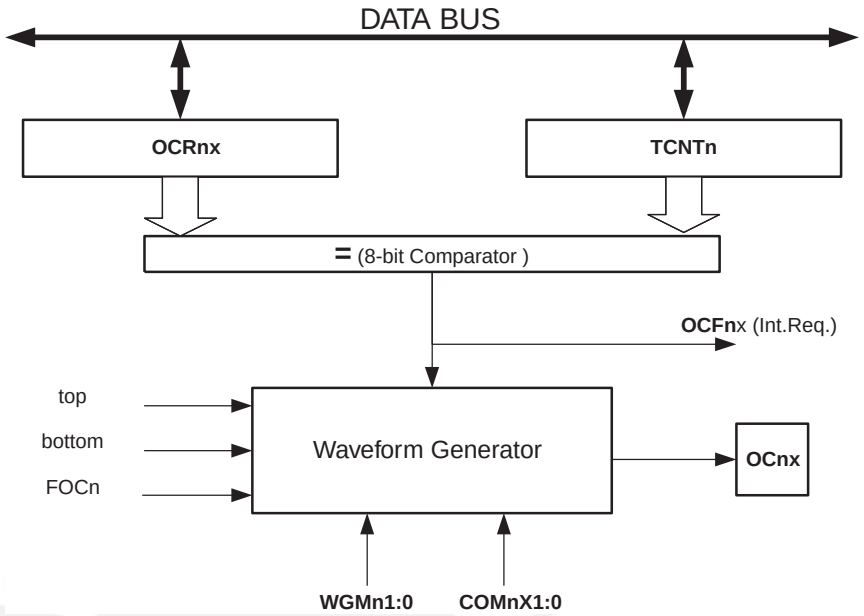
The Timer/Counter Overflow Flag (TOV2) is set according to the mode of operation selected by the WGM22:0 bits. TOV2 can be used for generating a CPU interrupt.

17.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT2 with the Output Compare Register (OCR2A and OCR2B). Whenever TCNT2 equals OCR2A or OCR2B, the comparator signals a match. A match will set the Output Compare Flag (OCF2A or OCF2B) at the next timer clock cycle. If the corresponding interrupt is enabled, the Output Compare Flag generates an Output Compare interrupt. The Output Compare Flag is automatically cleared when the interrupt is executed. Alternatively, the Output Compare 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 WGM22:0 bits and Compare Output mode (COM2x1:0) 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 149).

Figure 17-3 shows a block diagram of the Output Compare unit.

Figure 17-3. Output Compare Unit, Block Diagram



The OCR2x 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 OCR2x 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 OCR2x Register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR2x Buffer Register, and if double buffering is disabled the CPU will access the OCR2x directly.

17.5.1 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 (FOC2x) bit. Forcing compare match will not set the OCF2x Flag or reload/clear the timer, but the OC2x pin will be updated as if a real compare match had occurred (the COM2x1:0 bits settings define whether the OC2x pin is set, cleared or toggled).

17.5.2 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 OCR2x to be initialized to the same value as TCNT2 without triggering an interrupt when the Timer/Counter clock is enabled.

17.5.3 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 channel, independently of whether the Timer/Counter is running or not. If the value written to TCNT2 equals the OCR2x 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.

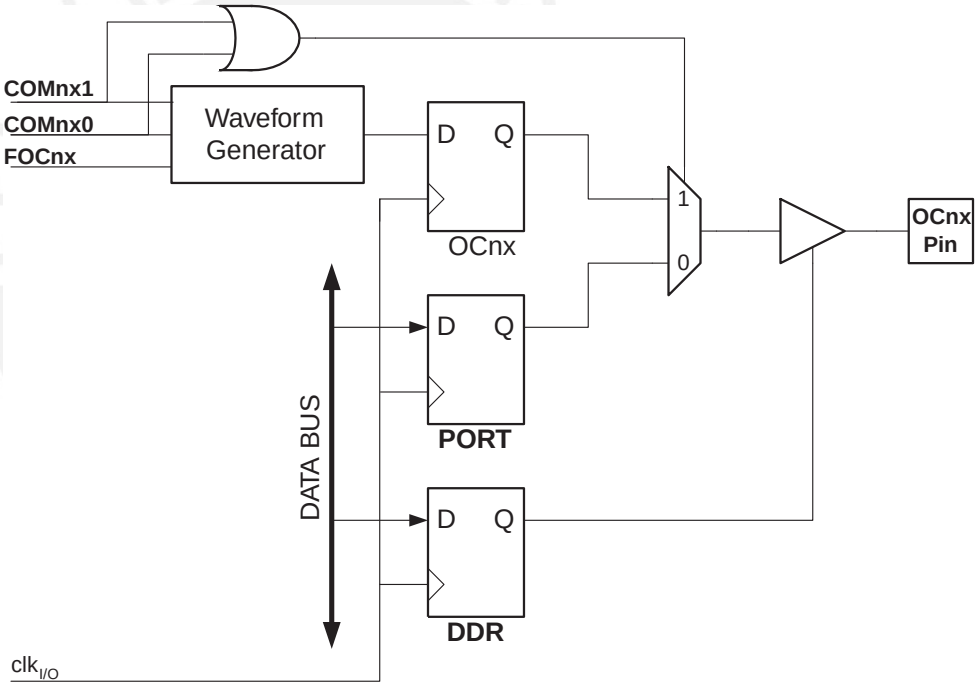
The setup of the OC2x should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC2x value is to use the Force Output Compare (FOC2x) strobe bit in Normal mode. The OC2x Register keeps its value even when changing between Waveform Generation modes.

Be aware that the COM2x1:0 bits are not double buffered together with the compare value. Changing the COM2x1:0 bits will take effect immediately.

17.6 Compare Match Output Unit

The Compare Output mode (COM2x1:0) bits have two functions. The Waveform Generator uses the COM2x1:0 bits for defining the Output Compare (OC2x) state at the next compare match. Also, the COM2x1:0 bits control the OC2x pin output source. Figure 17-4 shows a simplified schematic of the logic affected by the COM2x1:0 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 COM2x1:0 bits are shown. When referring to the OC2x state, the reference is for the internal OC2x Register, not the OC2x pin.

Figure 17-4. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC2x) from the Waveform Generator if either of the COM2x1:0 bits are set. However, the OC2x 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 OC2x pin (DDR_OC2x) must be set as output before the OC2x 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 OC2x state before the output is enabled. Note that some COM2x1:0 bit settings are reserved for certain modes of operation. See Section “17.11” on page 158.

17.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM2x1:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM2x1:0 = 0 tells the Waveform Generator that no action on the OC2x Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 17-5 on page 159. For fast PWM mode, refer to Table 17-6 on page 159, and for phase correct PWM refer to Table 17-7 on page 160.

A change of the COM2x1:0 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 FOC2x strobe bits.

17.7 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM22:0) and Compare Output mode (COM2x1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM2x1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM2x1:0 bits control whether the output should be set, cleared, or toggled at a compare match (See Section “17.6” on page 148.).

For detailed timing information refer to ”Timer/Counter Timing Diagrams” on page 153.

17.7.1 Normal Mode

The simplest mode of operation is the Normal mode (WGM22:0 = 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.

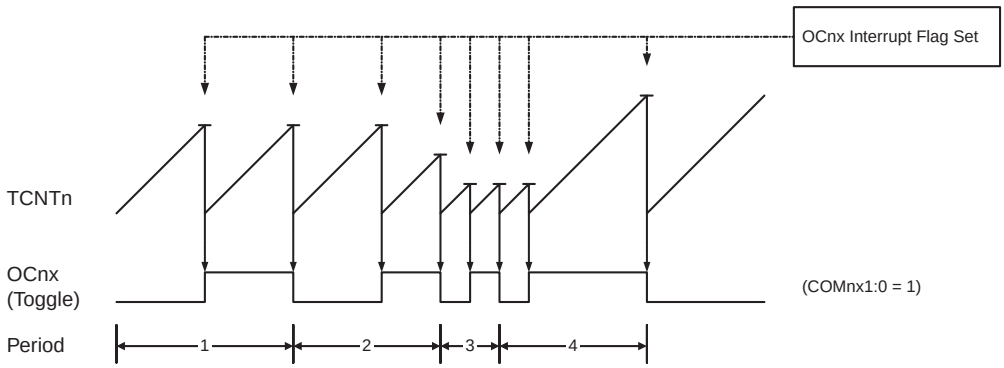
The Output Compare unit can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

17.7.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM22:0 = 2), the OCR2A Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT2) matches the OCR2A. The OCR2A 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 17-5. The counter value (TCNT2) increases until a compare match occurs between TCNT2 and OCR2A, and then counter (TCNT2) is cleared.

Figure 17-5. CTC Mode, Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF2A 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 OCR2A 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 OC2A output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM2A1:0 = 1). The OC2A 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 $f_{OC2A} = f_{clk\_I/O}/2$ when OCR2A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OCnx} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnx)}$$

The *N* variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

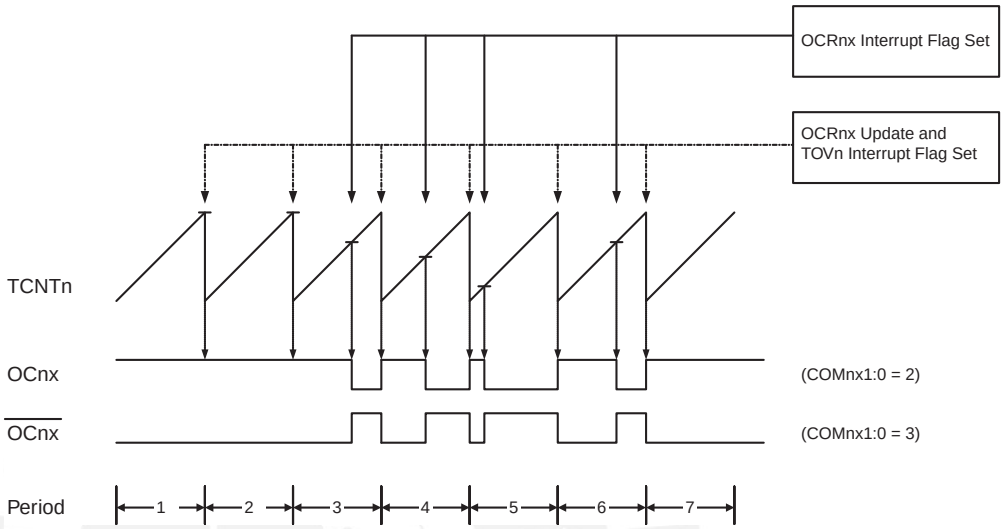
As for the Normal mode of operation, the TOV2 Flag is set in the same timer clock cycle that the counter counts from MAX to 0x00.

17.7.3 Fast PWM Mode

The fast Pulse Width Modulation or fast PWM mode (WGM22:0 = 3 or 7) 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 TOP then restarts from BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when MGM2:0 = 7. In non-inverting Compare Output mode, the Output Compare (OC2x) is cleared on the compare match between TCNT2 and OCR2x, 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 TOP value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 17-6. 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 OCR2x and TCNT2.

Figure 17-6. Fast PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV2) is set each time the counter reaches TOP. 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 OC2x pin. Setting the COM2x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM2x1:0 to three. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when MGM2:0 = 7. (See Table 17-3 on page 158). The actual OC2x 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 OC2x Register at the compare match between OCR2x and TCNT2, and clearing (or setting) the OC2x 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:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

The *N* variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

The extreme values for the OCR2A Register represent special cases when generating a PWM waveform output in the fast PWM mode. If the OCR2A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR2A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM2A1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC2x to toggle its logical level on each compare match (COM2x1:0 = 1). The waveform

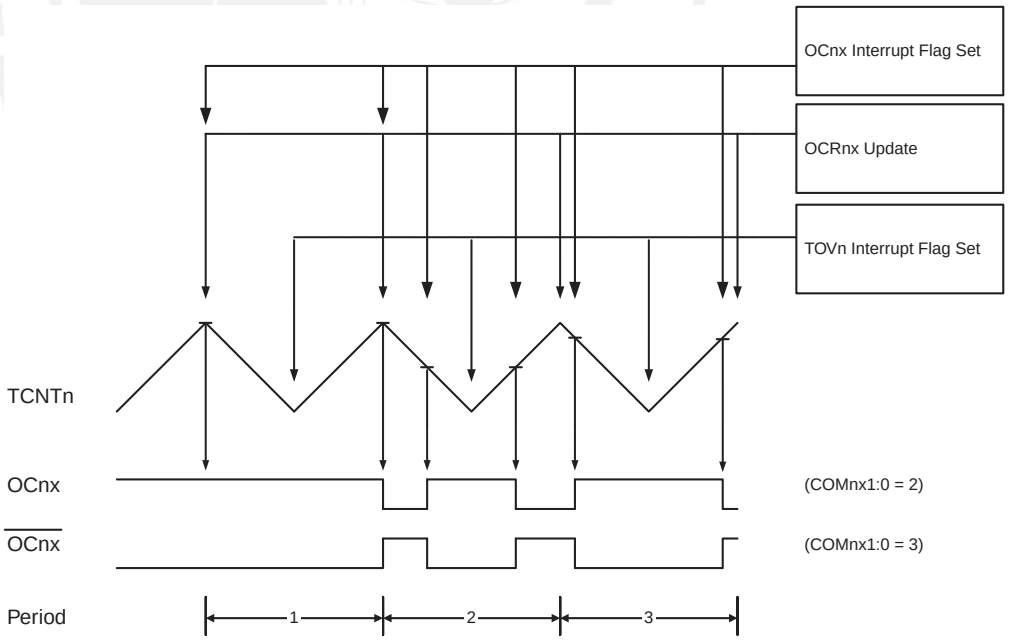
generated will have a maximum frequency of $f_{oc2} = f_{clk\_I/O}/2$ when OCR2A is set to zero. This feature is similar to the OC2A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

17.7.4 Phase Correct PWM Mode

The phase correct PWM mode (WGM22:0 = 1 or 5) 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 TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when MGM2:0 = 7. In non-inverting Compare Output mode, the Output Compare (OC2x) is cleared on the compare match between TCNT2 and OCR2x 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.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT2 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 17-7. 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 OCR2x and TCNT2.

Figure 17-7. 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 OC2x pin. Setting the COM2x1:0 bits to two will produce a non-inverted PWM. An inverted PWM

output can be generated by setting the COM2x1:0 to three. TOP is defined as 0xFF when WGM2:0 = 3, and OCR2A when MGM2:0 = 7 (See Table 17-4 on page 159). The actual OC2x 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 OC2x Register at the compare match between OCR2x and TCNT2 when the counter increments, and setting (or clearing) the OC2x Register at compare match between OCR2x and TCNT2 when the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$

The *N* variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

The extreme values for the OCR2A Register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR2A 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 17-7 OCnx 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.

- OCR2A changes its value from MAX, like in Figure 17-7. 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 correspond to 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 change that would have happened on the way up.

17.8 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 17-8 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 17-8. Timer/Counter Timing Diagram, no Prescaling

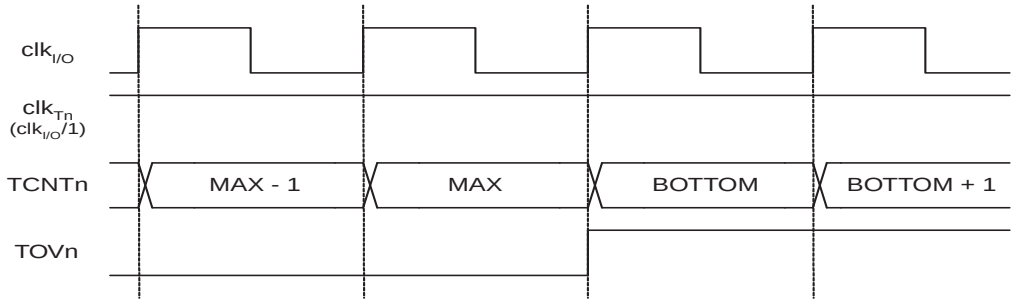


Figure 17-9 shows the same timing data, but with the prescaler enabled.

Figure 17-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)

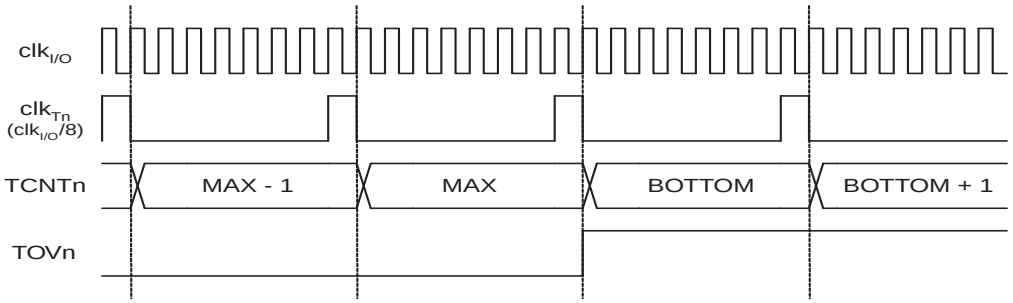


Figure 17-10 shows the setting of OCF2A in all modes except CTC mode.

Figure 17-10. Timer/Counter Timing Diagram, Setting of OCF2A, with Prescaler ($f_{clk\_I/O}/8$)

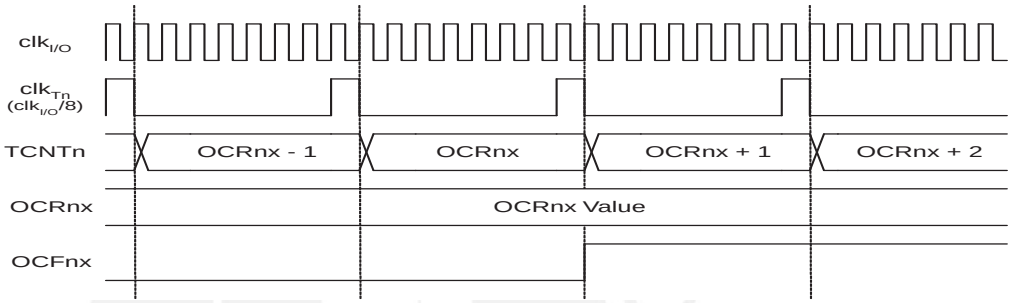
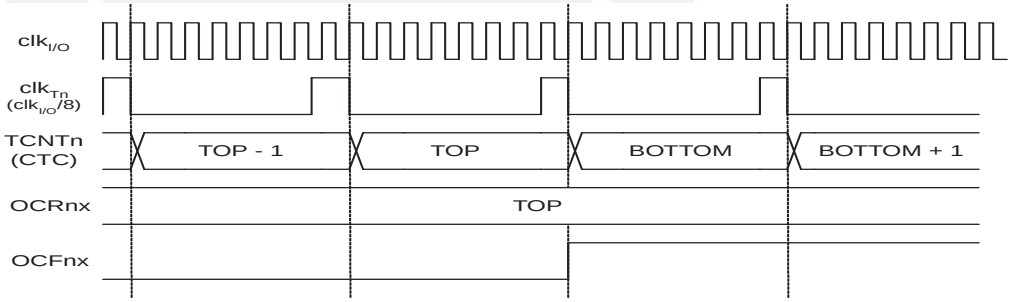


Figure 17-11 shows the setting of OCF2A and the clearing of TCNT2 in CTC mode.

Figure 17-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk\_I/O}/8$)



17.9 Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken.

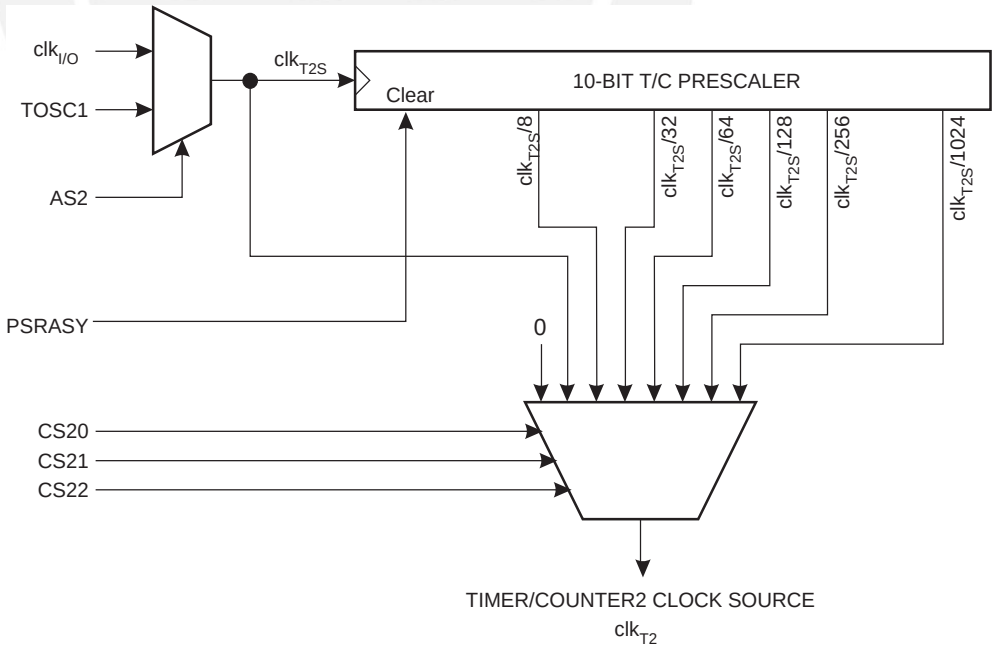
- Warning: When switching between asynchronous and synchronous clocking of Timer/Counter2, the Timer Registers TCNT2, OCR2x, and TCCR2x might be corrupted. A safe procedure for switching clock source is:
  a. Disable the Timer/Counter2 interrupts by clearing OCIE2x and TOIE2.
  b. Select clock source by setting AS2 as appropriate.
  c. Write new values to TCNT2, OCR2x, and TCCR2x.
  d. To switch to asynchronous operation: Wait for TCN2xUB, OCR2xUB, and TCR2xUB.
  e. Clear the Timer/Counter2 Interrupt Flags.
  f. Enable interrupts, if needed.
- The CPU main clock frequency must be more than four times the Oscillator frequency.
- When writing to one of the registers TCNT2, OCR2x, or TCCR2x, 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 five mentioned registers have their individual temporary register, which means that e.g. writing to TCNT2 does not disturb an OCR2x 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 ADC Noise Reduction mode after having written to TCNT2, OCR2x, or TCCR2x, 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 any of the Output Compare2 interrupt is used to wake up the device, since the Output Compare function is disabled during writing to OCR2x or TCNT2. If the write cycle is not finished, and the MCU enters sleep mode before the corresponding OCR2xUB 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 ADC Noise Reduction mode, precautions must be taken if the user wants to re-enter one of these modes: If re-entering sleep mode within the TOSC1 cycle, the interrupt will immidiately occur and the device wake up again. The result is multiple interrupts and wake-ups within one TOSC1 cycle from the first interrupt. If the user is in doubt whether the time before re-entering Power-save or ADC Noise Reduction mode is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
  a. Write a value to TCCR2x, TCNT2, or OCR2x.
  b. Wait until the corresponding Update Busy Flag in ASSR returns to zero.
  c. Enter Power-save or ADC Noise Reduction 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 ADC Noise Reduction 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:
    a. Write any value to either of the registers OCR2x or TCCR2x.
    b. Wait for the corresponding Update Busy Flag to be cleared.
    c. Read TCNT2.

During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes 3 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.

17.10 Timer/Counter Prescaler

Figure 17-12. 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.

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 PSRASY bit in GTCCR resets the prescaler. This allows the user to operate with a predictable prescaler.

17.11 Register Description

17.11.1 TCCR2A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
(0xB0)	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	TCCR2A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM2A1:0: Compare Match Output A Mode**

These bits control the Output Compare pin (OC2A) behavior. If one or both of the COM2A1:0 bits are set, the OC2A 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 OC2A pin must be set in order to enable the output driver.

When OC2A is connected to the pin, the function of the COM2A1:0 bits depends on the WGM22:0 bit setting. Table 17-2 shows the COM2A1:0 bit functionality when the WGM22:0 bits are set to a normal or CTC mode (non-PWM).

Table 17-2. Compare Output Mode, non-PWM Mode

COM2A1	COM2A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC2A on Compare Match
1	0	Clear OC2A on Compare Match
1	1	Set OC2A on Compare Match

Table 17-3 shows the COM2A1:0 bit functionality when the WGM21:0 bits are set to fast PWM mode.

Table 17-3. Compare Output Mode, Fast PWM Mode(1)

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC0A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match, set OC2A at BOTTOM, (non-inverting mode).
1	1	Set OC2A on Compare Match, clear OC2A at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR2A equals TOP and COM2A1 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 150 for more details.

Table 17-4 shows the COM2A1:0 bit functionality when the WGM22:0 bits are set to phase correct PWM mode.

Table 17-4. Compare Output Mode, Phase Correct PWM Mode(1)

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal Port Operation, OC2A Disconnected. WGM22 = 1: Toggle OC2A on Compare Match.
1	0	Clear OC2A on Compare Match when up-counting. Set OC2A on Compare Match when down-counting.
1	1	Set OC2A on Compare Match when up-counting. Clear OC2A on Compare Match when down-counting.

Note: 1. A special case occurs when OCR2A equals TOP and COM2A1 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 152 for more details.

- **Bits 5:4 – COM2B1:0: Compare Match Output B Mode**

These bits control the Output Compare pin (OC2B) behavior. If one or both of the COM2B1:0 bits are set, the OC2B 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 OC2B pin must be set in order to enable the output driver.

When OC2B is connected to the pin, the function of the COM2B1:0 bits depends on the WGM22:0 bit setting. Table 17-5 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to a normal or CTC mode (non-PWM).

Table 17-5. Compare Output Mode, non-PWM Mode

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Toggle OC2B on Compare Match
1	0	Clear OC2B on Compare Match
1	1	Set OC2B on Compare Match

Table 17-6 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to fast PWM mode.

Table 17-6. Compare Output Mode, Fast PWM Mode(1)

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on Compare Match, set OC2B at BOTTOM, (non-inverting mode).
1	1	Set OC2B on Compare Match, clear OC2B at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR2B equals TOP and COM2B1 is set. In this case, the Compare Match is ignored, but the set or clear is done at BOTTOM. See "Phase Correct PWM Mode" on page 152 for more details.

Table 17-7 shows the COM2B1:0 bit functionality when the WGM22:0 bits are set to phase correct PWM mode.

Table 17-7. Compare Output Mode, Phase Correct PWM Mode(1)

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on Compare Match when up-counting. Set OC2B on Compare Match when down-counting.
1	1	Set OC2B on Compare Match when up-counting. Clear OC2B on Compare Match when down-counting.

Note: 1. A special case occurs when OCR2B equals TOP and COM2B1 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 152 for more details.

- **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bits 1:0 – WGM21:0: Waveform Generation Mode**

Combined with the WGM22 bit found in the TCCR2B 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 17-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see "Modes of Operation" on page 149).

Table 17-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX= 0xFF
2. BOTTOM= 0x00

17.11.2 TCCR2B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
(0xB1)	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC2A: Force Output Compare A**

The FOC2A 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 TCCR2B is written when operating in PWM mode. When writing a logical one to the FOC2A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC2A output is changed according to its COM2A1:0 bits setting. Note that the FOC2A bit is implemented as a strobe. Therefore it is the value present in the COM2A1:0 bits that determines the effect of the forced compare.

A FOC2A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR2A as TOP.

The FOC2A bit is always read as zero.

- **Bit 6 – FOC2B: Force Output Compare B**

The FOC2B 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 TCCR2B is written when operating in PWM mode. When writing a logical one to the FOC2B bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC2B output is changed according to its COM2B1:0 bits setting. Note that the FOC2B bit is implemented as a strobe. Therefore it is the value present in the COM2B1:0 bits that determines the effect of the forced compare.

A FOC2B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR2B as TOP.

The FOC2B bit is always read as zero.

- **Bits 5:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 3 – WGM22: Waveform Generation Mode**

See the description in the ”TCCR2A – Timer/Counter Control Register A” on page 158.

- **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 17-9 on page 162.

Table 17-9. 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)

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.

17.11.3 TCNT2 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
(0xB2)	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 OCR2x Registers.

17.11.4 OCR2A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
(0xB3)	OCR2A[7:0]								OCR2A
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 A 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 OC2A pin.

17.11.5 OCR2B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
(0xB4)	OCR2B[7:0]								OCR2B
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 B 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 OC2B pin.

17.11.6 TIMSK2 – Timer/Counter2 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x70)	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	TIMSK2
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – OCIE2B: Timer/Counter2 Output Compare Match B Interrupt Enable**

When the OCIE2B bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match B interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter2 occurs, i.e., when the OCF2B bit is set in the Timer/Counter 2 Interrupt Flag Register – TIFR2.

- **Bit 1 – OCIE2A: Timer/Counter2 Output Compare Match A Interrupt Enable**

When the OCIE2A bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter2 occurs, i.e., when the OCF2A bit is set in the Timer/Counter 2 Interrupt Flag Register – TIFR2.

- **Bit 0 – 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, i.e., when the TOV2 bit is set in the Timer/Counter2 Interrupt Flag Register – TIFR2.

17.11.7 TIFR2 – Timer/Counter2 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	–	–	–	–	–	OCF2B	OCF2A	TOV2	TIFR2
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – OCF2B: Output Compare Flag 2 B**

The OCF2B bit is set (one) when a compare match occurs between the Timer/Counter2 and the data in OCR2B – Output Compare Register2. OCF2B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE2B (Timer/Counter2 Compare match Interrupt Enable), and OCF2B are set (one), the Timer/Counter2 Compare match Interrupt is executed.

- **Bit 1 – OCF2A: Output Compare Flag 2 A**

The OCF2A bit is set (one) when a compare match occurs between the Timer/Counter2 and the data in OCR2A – Output Compare Register2. OCF2A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE2A (Timer/Counter2 Compare match Interrupt Enable), and OCF2A are set (one), the Timer/Counter2 Compare match Interrupt is executed.

- **Bit 0 – 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, TOIE2A (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 0x00.

17.11.8 ASSR – Asynchronous Status Register

Bit	7	6	5	4	3	2	1	0	
(0xB6)	–	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	ASSR
Read/Write	R	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RES: Reserved bit**

This bit is reserved and will always read as zero.

- **Bit 6 – EXCLK: Enable External Clock Input**

When EXCLK is written to one, and asynchronous clock is selected, the external clock input buffer is enabled and an external clock can be input on Timer Oscillator 1 (TOSC1) pin instead of a 32 kHz crystal. Writing to EXCLK should be done before asynchronous operation is selected. Note that the crystal Oscillator will only run when this bit is zero.

- **Bit 5 – AS2: Asynchronous Timer/Counter2**

When AS2 is written to zero, Timer/Counter2 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, OCR2A, OCR2B, TCCR2A and TCCR2B might be corrupted.

- **Bit 4 – 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 3 – OCR2AUB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2A is written, this bit becomes set. When OCR2A has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2A is ready to be updated with a new value.

- **Bit 2 – OCR2BUB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2B is written, this bit becomes set. When OCR2B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2B is ready to be updated with a new value.

- **Bit 1 – TCR2AUB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2A is written, this bit becomes set. When TCCR2A has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2A is ready to be updated with a new value.

- **Bit 0 – TCR2BUB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2B is written, this bit becomes set. When TCCR2B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2B is ready to be updated with a new value.

If a write is performed to any of the five 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, OCR2A, OCR2B, TCCR2A and TCCR2B are different. When reading TCNT2, the actual timer value is read. When reading OCR2A, OCR2B, TCCR2A and TCCR2B the value in the temporary storage register is read.

17.11.9 GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	TSM	–	–	–	–	–	PSRASY	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – PSRASY: Prescaler Reset Timer/Counter2**

When this bit is one, the Timer/Counter2 prescaler will be reset. This bit is normally cleared immediately by hardware. If the bit is written when Timer/Counter2 is operating in asynchronous mode, the bit will remain one until the prescaler has been reset. The bit will not be cleared by hardware if the TSM bit is set. Refer to the description of the ”Bit 7 – TSM: Timer/Counter Synchronization Mode” on page 143 for a description of the Timer/Counter Synchronization mode.

18. SPI – Serial Peripheral Interface

18.1 Features

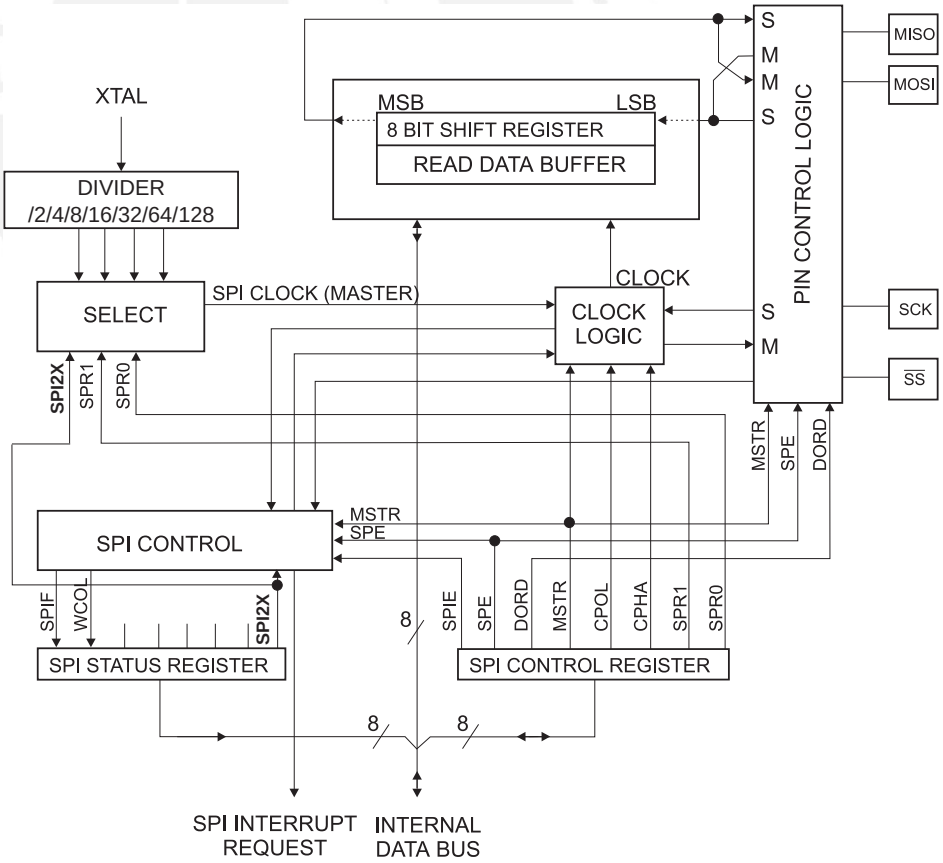
- **Full-duplex, Three-wire Synchronous Data Transfer**
- **Master or Slave Operation**
- **LSB First or MSB First Data Transfer**
- **Seven Programmable Bit Rates**
- **End of Transmission Interrupt Flag**
- **Write Collision Flag Protection**
- **Wake-up from Idle Mode**
- **Double Speed (CK/2) Master SPI Mode**

18.2 Overview

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega48PA/88PA/168PA/328P and peripheral devices or between several AVR devices.

The USART can also be used in Master SPI mode, see “USART in SPI Mode” on page 204. The PRSPI bit in ”Minimizing Power Consumption” on page 42 must be written to zero to enable SPI module.

Figure 18-1. SPI Block Diagram(1)



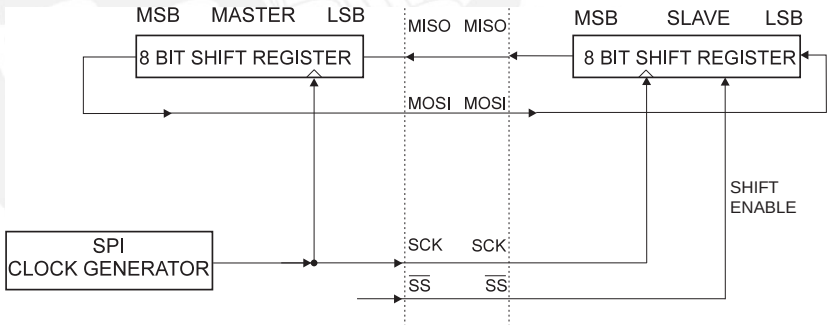
Note: 1. Refer to Figure 1-1 on page 2, and Table 13-3 on page 82 for SPI pin placement.

The interconnection between Master and Slave CPUs with SPI is shown in Figure 18-2 on page 167. 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 18-2. 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 18-1 on page 168. For more details on automatic port overrides, refer to "Alternate Port Functions" on page 80.

Table 18-1. SPI Pin Overrides(Note:)

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 82 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. E.g. 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
  in    r16, SPSR
  sbrsr16, 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.

18.3 $\overline{SS}$ Pin Functionality

18.3.1 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, 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.

18.3.2 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.

18.4 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 18-3 and Figure 18-4 on page 172. 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 18-3 on page 173 and Table 18-4 on page 173, as done in Table 18-2.

Table 18-2. SPI Modes

SPI Mode	Conditions	Leading Edge	Trailing eDge
0	CPOL=0, CPHA=0	Sample (Rising)	Setup (Falling)
1	CPOL=0, CPHA=1	Setup (Rising)	Sample (Falling)
2	CPOL=1, CPHA=0	Sample (Falling)	Setup (Rising)
3	CPOL=1, CPHA=1	Setup (Falling)	Sample (Rising)

Figure 18-3. SPI Transfer Format with CPHA = 0

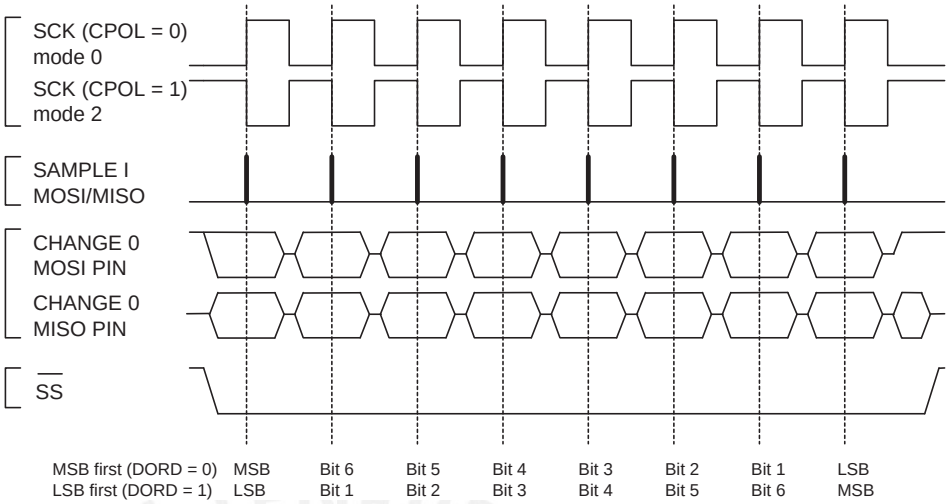
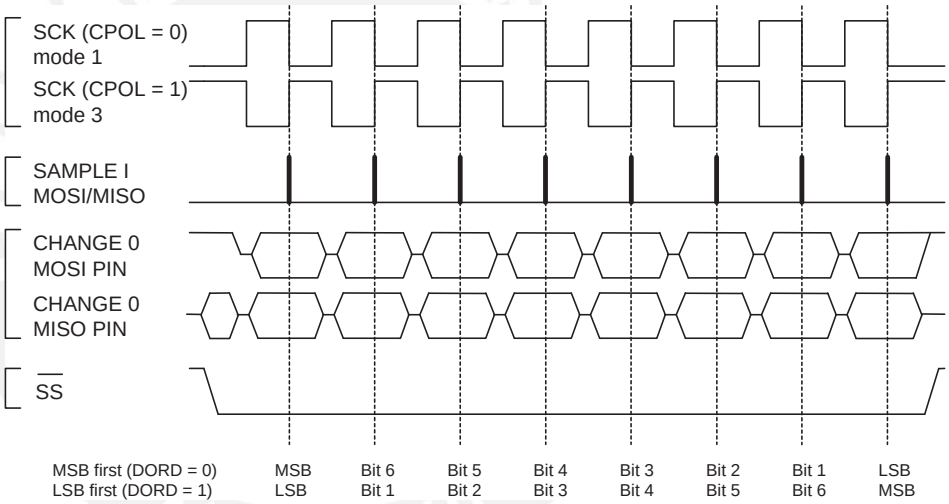


Figure 18-4. SPI Transfer Format with CPHA = 1



18.5 Register Description

18.5.1 SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	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 18-3 and Figure 18-4 for an example. The CPOL functionality is summarized below:

Table 18-3. 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 18-3 and Figure 18-4 for an example. The CPOL functionality is summarized below:

Table 18-4. 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 18-5. 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	$f_{osc}/64$

18.5.2 SPSR – SPI Status Register

Bit	7	6	5	4	3	2	1	0	
0x2D (0x4D)	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 ATmega48PA/88PA/168PA/328P 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 18-5). 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 ATmega48PA/88PA/168PA/328P is also used for program memory and EEPROM downloading or uploading. See page 308 for serial programming and verification.

18.5.3 SPDR – SPI Data Register

Bit	7	6	5	4	3	2	1	0	
0x2E (0x4E)	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.

19. USART0

19.1 Features

- **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**

19.2 Overview

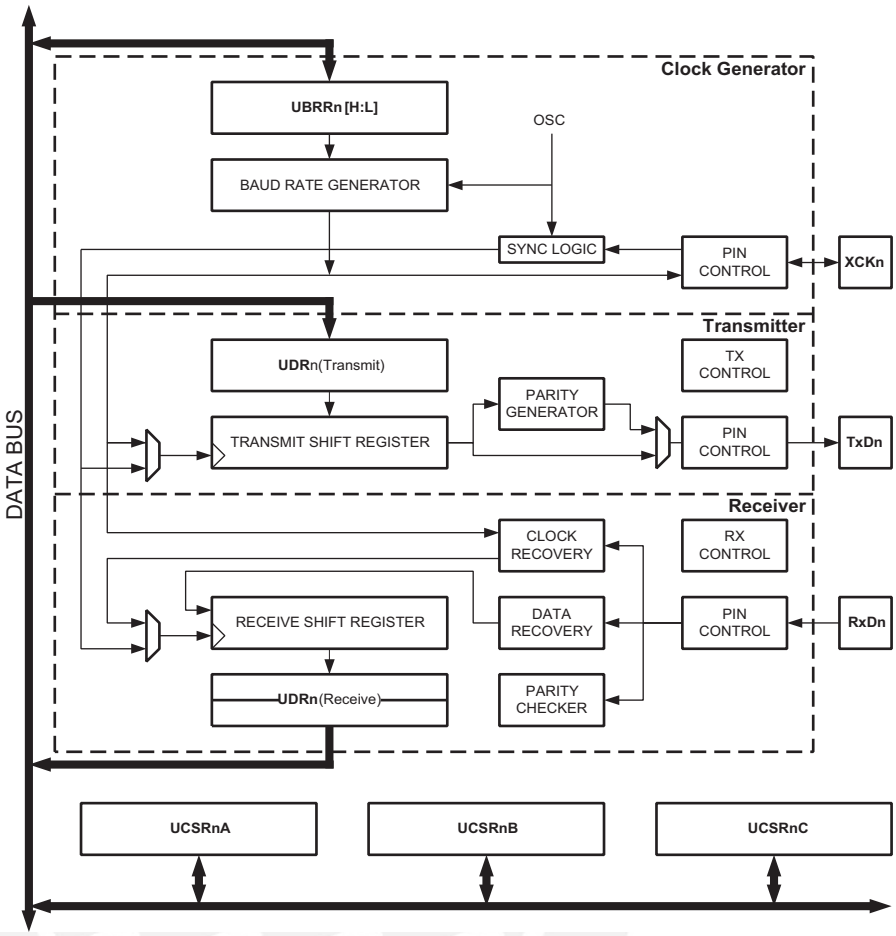
The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device.

The USART0 can also be used in Master SPI mode, see “USART in SPI Mode” on page 204. The Power Reduction USART bit, PRUSART0, in ”Minimizing Power Consumption” on page 42 must be disabled by writing a logical zero to it.

A simplified block diagram of the USART Transmitter is shown in Figure 19-1 on page 177. CPU accessible I/O Registers and I/O pins are shown in bold.

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 XCKn (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 (UDRn). The Receiver supports the same frame formats as the Transmitter, and can detect Frame Error, Data OverRun and Parity Errors.

Figure 19-1. USART Block Diagram[1]



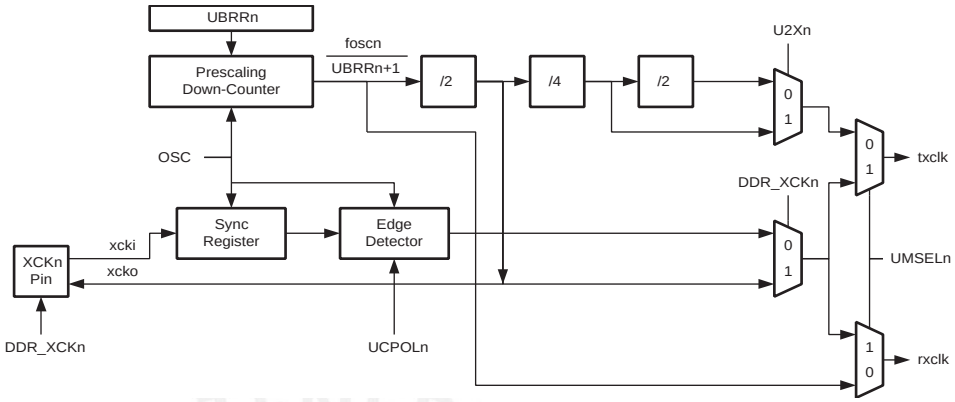
Note: 1. Refer to Figure 1-1 on page 2 and Table 13-9 on page 88 for USART0 pin placement.

19.3 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 UMSELn bit in USART Control and Status Register C (UCSRnC) selects between asynchronous and synchronous operation. Double Speed (asynchronous mode only) is controlled by the U2Xn found in the UCSRnA Register. When using synchronous mode (UMSELn = 1), the Data Direction Register for the XCKn pin (DDR_XCKn) controls whether the clock source is internal (Master mode) or external (Slave mode). The XCKn pin is only active when using synchronous mode.

Figure 19-2 shows a block diagram of the clock generation logic.

Figure 19-2. 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).

19.3.1 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 19-2.

The USART Baud Rate Register (UBRRn) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (f_{osc}), is loaded with the UBRRn value each time the counter has counted down to zero or when the UBRRnL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output (= f_{osc}/(UBRRn+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 UMSELn, U2Xn and DDR_XCKn bits.

Table 19-1 contains equations for calculating the baud rate (in bits per second) and for calculating the UBRRn value for each mode of operation using an internally generated clock source.

Table 19-1. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRRn Value
Asynchronous Normal mode (U2Xn = 0)	$BAUD = \frac{f_{OSC}}{16(\boldsymbol{UBRR}n + 1)}$	$\boldsymbol{UBRR}n = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2Xn = 1)	$BAUD = \frac{f_{OSC}}{8(\boldsymbol{UBRR}n + 1)}$	$\boldsymbol{UBRR}n = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(\boldsymbol{UBRR}n + 1)}$	$\boldsymbol{UBRR}n = \frac{f_{OSC}}{2BAUD} - 1$

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

UBRRn Contents of the UBRRnH and UBRRnL Registers, (0-4095)

Some examples of UBRRn values for some system clock frequencies are found in Table 19-9 (see page 200).

19.3.2 Double Speed Operation (U2Xn)

The transfer rate can be doubled by setting the U2Xn bit in UCSRnA. 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.

19.3.3 External Clock

External clocking is used by the synchronous slave modes of operation. The description in this section refers to Figure 19-2 for details.

External clock input from the XCKn 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 XCKn clock frequency is limited by the following equation:

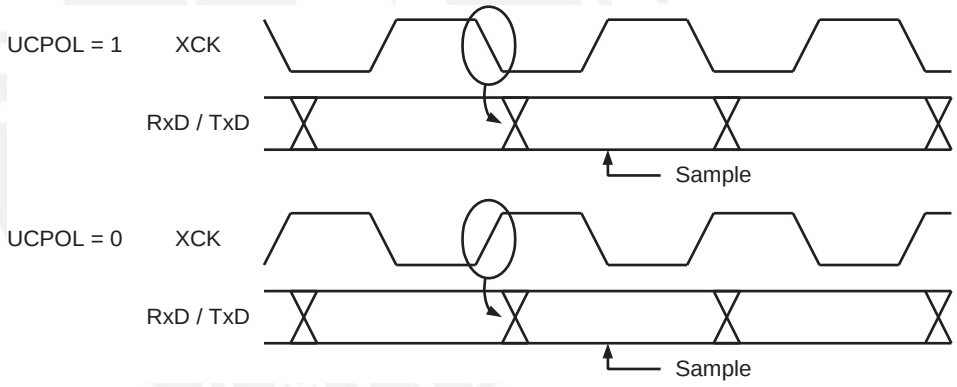
$$f_{XCK} < \frac{f_{OSC}}{4}$$

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.

19.3.4 Synchronous Clock Operation

When synchronous mode is used (UMSELn = 1), the XCKn 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 RxDn) is sampled at the opposite XCKn clock edge of the edge the data output (TxDn) is changed.

Figure 19-3. Synchronous Mode XCKn Timing.



The UCPOLn bit UCRSC selects which XCKn clock edge is used for data sampling and which is used for data change. As Figure 19-3 shows, when UCPOLn is zero the data will be changed at rising XCKn edge and sampled at falling XCKn edge. If UCPOLn is set, the data will be changed at falling XCKn edge and sampled at rising XCKn edge.

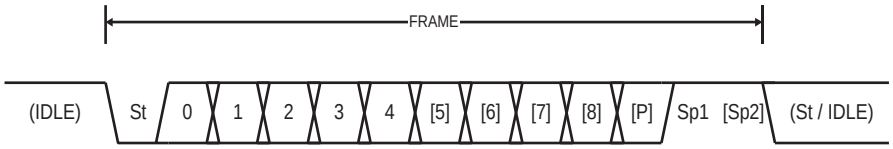
19.4 Frame Formats

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 19-4 illustrates the possible combinations of the frame formats. Bits inside brackets are optional.

Figure 19-4. 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 (RxDn or TxDn). An IDLE line must be high.

The frame format used by the USART is set by the UCSZn2:0, UPMn1:0 and USBSn bits in UCSRnB and UCSRnC. 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 (UCSZn2:0) bits select the number of data bits in the frame. The USART Parity mode (UPMn1:0) 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 (USBSn) 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.

19.4.1 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.

19.5 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 TXCn 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 TXCn Flag must be cleared before each transmission (before UDRn 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.

Assembly Code Example[1]

```
USART_Init:
  ; Set baud rate
  out   UBRRnH, r17
  out   UBRRnL, r16
  ; Enable receiver and transmitter
  ldi   r16, (1<<RXENn)|(1<<TXENn)
  out   UCSRnB,r16
  ; Set frame format: 8data, 2stop bit
  ldi   r16, (1<<USBSn)|(3<<UCSZn0)
  out   UCSRnC,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 */
  UBRR0H = (unsigned char)(ubrr>>8);
  UBRR0L = (unsigned char)ubrr;
  Enable receiver and transmitter */
  UCSR0B = (1<<RXEN0)|(1<<TXEN0);
  /* Set frame format: 8data, 2stop bit */
  UCSR0C = (1<<USBS0)|(3<<UCSZ00);
}
```

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.

19.6 Data Transmission – The USART Transmitter

The USART Transmitter is enabled by setting the *Transmit Enable* (TXEN) bit in the UCSRnB Register. When the Transmitter is enabled, the normal port operation of the TxDn 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 syn-

chronous operation is used, the clock on the XCKn pin will be overridden and used as transmission clock.

19.6.1 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 UDRn 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, U2Xn bit or by XCKn depending on mode of operation.

The following code examples show a simple USART transmit function based on polling of the *Data Register Empty* (UDREn) Flag. When using frames with less than eight bits, the most significant bits written to the UDRn 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 UCSRnA,UDREn
  rjmp USART_Transmit
  ; Put data (r16) into buffer, sends the data
  out  UDRn,r16
  ret
```

C Code Example[1]

```
void USART_Transmit( unsigned char data )
{
  /* Wait for empty transmit buffer */
  while ( !( UCSRnA & (1<<UDREn)) )
       ;
  /* Put data into buffer, sends the data */
  UDRn = data;
}
```

Note: 1. See ”About Code Examples” on page 7.

The function simply waits for the transmit buffer to be empty by checking the UDREn 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.

19.6.2 Sending Frames with 9 Data Bit

If 9-bit characters are used (UCSZn = 7), the ninth bit must be written to the TXB8 bit in UCSRnB before the low byte of the character is written to UDRn. 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][2]

```
USART_Transmit:
  ; Wait for empty transmit buffer
  sbis UCSRnA,UDREn
  rjmp USART_Transmit
  ; Copy 9th bit from r17 to TXB8
  cbi  UCSRnB,TXB8
  sbrc r17,0
  sbi  UCSRnB,TXB8
  ; Put LSB data (r16) into buffer, sends the data
  out  UDRn,r16
  ret
```

C Code Example[1][2]

```
void USART_Transmit( unsigned int data )
{
  /* Wait for empty transmit buffer */
  while ( !( UCSRnA & (1<<UDREn)) )
    ;
  /* Copy 9th bit to TXB8 */
  UCSRnB &= ~(1<<TXB8);
  if ( data & 0x0100 )
    UCSRnB |= (1<<TXB8);
  /* Put data into buffer, sends the data */
  UDRn = data;
}
```

Notes: 1. These transmit functions are written to be general functions. They can be optimized if the contents of the UCSRnB is static. For example, only the TXB8 bit of the UCSRnB Register is used after initialization.
2. See "About Code Examples" on page 7.

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.

19.6.3 Transmitter Flags and Interrupts

The USART Transmitter has two flags that indicate its state: USART Data Register Empty (UDREn) and Transmit Complete (TXCn). Both flags can be used for generating interrupts.

The Data Register Empty (UDREn) 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 UCSRnA Register.

When the Data Register Empty Interrupt Enable (UDRIEn) bit in UCSRnB is written to one, the USART Data Register Empty Interrupt will be executed as long as UDREn is set (provided that global interrupts are enabled). UDREn is cleared by writing UDRn. When interrupt-driven data transmission is used, the Data Register Empty interrupt routine must either write new data to

UDRn in order to clear UDREn or disable the Data Register Empty interrupt, otherwise a new interrupt will occur once the interrupt routine terminates.

The Transmit Complete (TXCn) 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 TXCn 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 TXCn Flag is useful in half-duplex communication interfaces (like the RS-485 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 (TXCIEn) bit in UCSRnB is set, the USART Transmit Complete Interrupt will be executed when the TXCn 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 TXCn Flag, this is done automatically when the interrupt is executed.

19.6.4 Parity Generator

The Parity Generator calculates the parity bit for the serial frame data. When parity bit is enabled (UPMn1 = 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.

19.6.5 Disabling the Transmitter

The disabling of the Transmitter (setting the TXEN to zero) will not become effective until ongoing and pending transmissions are completed, i.e., 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 TxDn pin.

19.7 Data Reception – The USART Receiver

The USART Receiver is enabled by writing the Receive Enable (RXENn) bit in the UCSRnB Register to one. When the Receiver is enabled, the normal pin operation of the RxDn 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 XCKn pin will be used as transfer clock.

19.7.1 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 XCKn 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, i.e., 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 UDRn I/O location.

The following code example shows a simple USART receive function based on polling of the Receive Complete (RXCn) Flag. When using frames with less than eight bits the most significant bits of the data read from the UDRn 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 UCSRnA, RXCn`
`  rjmp USART_Receive`
`  ; Get and return received data from buffer`
`  in  r16, UDRn`
`  ret`

C Code Example[1]

`unsigned char USART_Receive( void )`
`{`
`  /* Wait for data to be received */`
`  while ( !(UCSRnA & (1<<RXCn)) )`
`  ;`
`  /* Get and return received data from buffer */`
`  return UDRn;`
`}`

Note: 1. See "About Code Examples" on page 7.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

The function simply waits for data to be present in the receive buffer by checking the RXCn Flag, before reading the buffer and returning the value.

19.7.2 Receiving Frames with 9 Data Bits

If 9-bit characters are used (UCSZn=7) the ninth bit must be read from the RXB8n bit in UCSRnB **before** reading the low bits from the UDRn. This rule applies to the FEn, DORn and UPEn Status Flags as well. Read status from UCSRnA, then data from UDRn. Reading the UDRn I/O location will change the state of the receive buffer FIFO and consequently the TXB8n, FEn, DORn and UPEn bits, which all are stored in the FIFO, will change.

The following code example shows a simple USART receive function that handles both nine bit characters and the status bits.

Assembly Code Example[1]

```
USART_Receive:
  ; Wait for data to be received
  sbis UCSRnA, RXCn
  rjmp USART_Receive
  ; Get status and 9th bit, then data from buffer
  in   r18, UCSRnA
  in   r17, UCSRnB
  in   r16, UDRn
  ; If error, return -1
  andi r18,(1<<FEn)|(1<<DORn)|(1<<UPEn)
  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 ( !(UCSRnA & (1<<RXCn)) )
       ;
  /* Get status and 9th bit, then data */
  /* from buffer */
  status = UCSRnA;
  resh = UCSRnB;
  resl = UDRn;
  /* If error, return -1 */
  if ( status & (1<<FEn)|(1<<DORn)|(1<<UPEn) )
    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.
For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

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.

19.7.3 Receive Compete Flag and Interrupt

The USART Receiver has one flag that indicates the Receiver state.

The Receive Complete (RXCn) 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 (i.e., does not contain any unread data). If the Receiver is disabled (RXENn = 0), the receive buffer will be flushed and consequently the RXCn bit will become zero.

When the Receive Complete Interrupt Enable (RXCIEn) in UCSRnB is set, the USART Receive Complete interrupt will be executed as long as the RXCn 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 UDRn in order to clear the RXCn Flag, otherwise a new interrupt will occur once the interrupt routine terminates.

19.7.4 Receiver Error Flags

The USART Receiver has three Error Flags: Frame Error (FEn), Data OverRun (DORn) and Parity Error (UPEn). All can be accessed by reading UCSRnA. 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 UCSRnA must be read before the receive buffer (UDRn), since reading the UDRn 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 UCSRnA is written for upward compatibility of future USART implementations. None of the Error Flags can generate interrupts.

The Frame Error (FEn) Flag indicates the state of the first stop bit of the next readable frame stored in the receive buffer. The FEn Flag is zero when the stop bit was correctly read (as one), and the FEn 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 FEn Flag is not affected by the setting of the USBSn bit in UCSRnC 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 UCSRnA.

The Data OverRun (DORn) 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 DORn Flag is set there was one or more serial frame lost between the frame last read from UDRn, and the next frame read from UDRn. For compatibility with future devices, always write this bit to zero when writing to UCSRnA. The DORn Flag is cleared when the frame received was successfully moved from the Shift Register to the receive buffer.

The Parity Error (UPEn) Flag indicates that the next frame in the receive buffer had a Parity Error when received. If Parity Check is not enabled the UPEn bit will always be read zero. For compatibility with future devices, always set this bit to zero when writing to UCSRnA. For more details see ”Parity Bit Calculation” on page 181 and ”Parity Checker” on page 189.

19.7.5 Parity Checker

The Parity Checker is active when the high USART Parity mode (UPMn1) bit is set. Type of Parity Check to be performed (odd or even) is selected by the UPMn0 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 (UPEn) Flag can then be read by software to check if the frame had a Parity Error.

The UPEn 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 (UPMn1 = 1). This bit is valid until the receive buffer (UDRn) is read.

19.7.6 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 (i.e., the RXENn is set to zero) the Receiver will no longer override the normal function of the RxDn port pin. The Receiver buffer FIFO will be flushed when the Receiver is disabled. Remaining data in the buffer will be lost

19.7.7 Flushing the Receive Buffer

The receiver buffer FIFO will be flushed when the Receiver is disabled, i.e., 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 UDRn I/O location until the RXCn Flag is cleared. The following code example shows how to flush the receive buffer.

Assembly Code Example[1]

```
USART_Flush:
  sbis UCSRnA, RXCn
  ret
  in   r16, UDRn
  rjmp USART_Flush
```

C Code Example[1]

```
void USART_Flush( void )
{
  unsigned char dummy;
  while ( UCSRnA & (1<<RXCn) ) dummy = UDRn;
}
```

Note: 1. See "About Code Examples" on page 7.
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

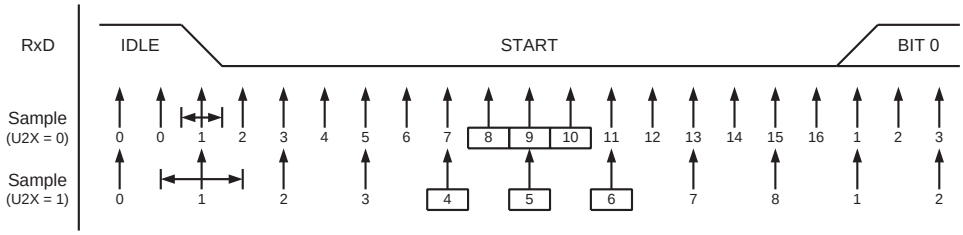
19.8 Asynchronous Data Reception

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 RxDn 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.

19.8.1 Asynchronous Clock Recovery

The clock recovery logic synchronizes internal clock to the incoming serial frames. Figure 19-5 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 eight 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 (U2Xn = 1) of operation. Samples denoted zero are samples done when the RxDn line is idle (i.e., no communication activity).

Figure 19-5. Start Bit Sampling

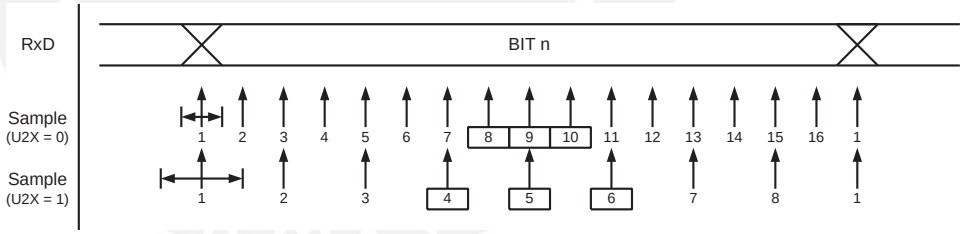


When the clock recovery logic detects a high (idle) to low (start) transition on the RxDn line, the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample as shown in the figure. The clock recovery logic then uses samples 8, 9, and 10 for Normal mode, and samples 4, 5, and 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.

19.8.2 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 eight states for each bit in Double Speed mode. Figure 19-6 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.

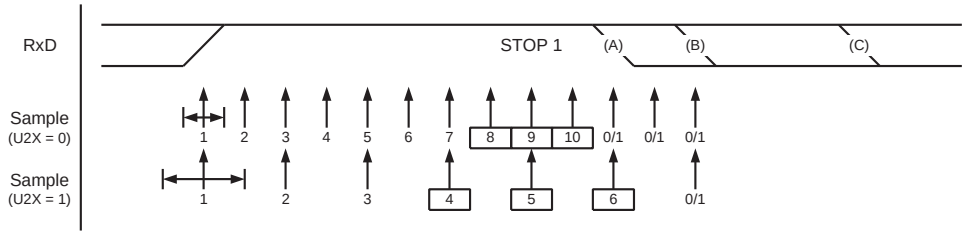
Figure 19-6. Sampling of Data and Parity Bit



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 RxDn 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 19-7 on page 192 shows the sampling of the stop bit and the earliest possible beginning of the start bit of the next frame.

Figure 19-7. 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 (FEn) 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 19-7. 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.

19.8.3 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 19-2 on page 193) 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.

$$R_{slow} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \qquad R_{fast} = \frac{(D+2)S}{(D+1)S+S_M}$$

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 19-2 on page 193 and Table 19-3 on page 193 list the maximum receiver baud rate error that can be tolerated. Note that Normal Speed mode has higher toleration of baud rate variations.

Table 19-2. Recommended Maximum Receiver Baud Rate Error for Normal Speed Mode (U2Xn = 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 19-3. Recommended Maximum Receiver Baud Rate Error for Double Speed Mode (U2Xn = 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
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 UBRRn value that gives an acceptable low error can be used if possible.

19.9 Multi-processor Communication Mode

Setting the Multi-processor Communication mode (MPCMn) bit in UCSRnA 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 MPCMn 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 (RXB8n) 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.

19.9.1 Using MPCMn

For an MCU to act as a master MCU, it can use a 9-bit character frame format (UCSZn = 7). The ninth bit (TXB8n) must be set when an address frame (TXB8n = 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 (MPCMn in UCSRnA is set).
2. The Master MCU sends an address frame, and all slaves receive and read this frame. In the Slave MCUs, the RXCn Flag in UCSRnA will be set as normal.
3. Each Slave MCU reads the UDRn Register and determines if it has been selected. If so, it clears the MPCMn bit in UCSRnA, otherwise it waits for the next address byte and keeps the MPCMn 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 MPCMn bit set, will ignore the data frames.
5. When the last data frame is received by the addressed MCU, the addressed MCU sets the MPCMn bit and waits for a new address frame from master. The process then repeats from 2.

Using any of the 5- 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- to 8-bit character frames are used, the Transmitter must be set to use two stop bit (USBSn = 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 MPCMn bit. The MPCMn bit shares the same I/O location as the TXCn Flag and this might accidentally be cleared when using SBI or CBI instructions.

19.10 Register Description

19.10.1 UDRn – USART I/O Data Register n

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDRn (Read)
	TXB[7:0]								UDRn (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 UDRn. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDRn Register location. Reading the UDRn Register location will return the contents of the Receive Data Buffer Register (RXB).

For 5-, 6-, 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 UDREn Flag in the UCSRnA Register is set. Data written to UDRn when the UDREn 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 TxDn 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.

19.10.2 UCSRnA – USART Control and Status Register n A

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
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 – RXCn: 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 (i.e., does not contain any unread data). If the Receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn Flag can be used to generate a Receive Complete interrupt (see description of the RXCIEn bit).

- **Bit 6 – TXCn: 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 (UDRn). The TXCn 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 TXCn Flag can generate a Transmit Complete interrupt (see description of the TXCIEn bit).

- **Bit 5 – UDREn: USART Data Register Empty**

The UDREn Flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREn is one, the buffer is empty, and therefore ready to be written. The UDREn Flag can generate a

Data Register Empty interrupt (see description of the UDRIEn bit). UDREn is set after a reset to indicate that the Transmitter is ready.

- **Bit 4 – FEn: Frame Error**

This bit is set if the next character in the receive buffer had a Frame Error when received. I.e., when the first stop bit of the next character in the receive buffer is zero. This bit is valid until the receive buffer (UDRn) is read. The FEn bit is zero when the stop bit of received data is one. Always set this bit to zero when writing to UCSRnA.

- **Bit 3 – DORn: 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 (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 2 – UPEn: USART 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 (UPMn1 = 1). This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 1 – U2Xn: 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 – MPCMn: Multi-processor Communication Mode**

This bit enables the Multi-processor Communication mode. When the MPCMn 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 MPCMn setting. For more detailed information see ”Multi-processor Communication Mode” on page 193.

19.10.3 UCSRnB – USART Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
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 – RXCIEn: RX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the RXCn Flag. A USART Receive Complete interrupt will be generated only if the RXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXCn bit in UCSRnA is set.

- **Bit 6 – TXCIEn: TX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the TXCn Flag. A USART Transmit Complete interrupt will be generated only if the TXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXCn bit in UCSRnA is set.

- **Bit 5 – UDRIEn: USART Data Register Empty Interrupt Enable n**

Writing this bit to one enables interrupt on the UDREn Flag. A Data Register Empty interrupt will be generated only if the UDRIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDREn bit in UCSRnA is set.

- **Bit 4 – RXENn: Receiver Enable n**

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxDn pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FEn, DORn, and UPEn Flags.

- **Bit 3 – TXENn: Transmitter Enable n**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxDn pin when enabled. The disabling of the Transmitter (writing TXENn to zero) will not become effective until ongoing and pending transmissions are completed, i.e., 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 TxDn port.

- **Bit 2 – UCSZn2: Character Size n**

The UCSZn2 bits combined with the UCSZn1:0 bit in UCSRnC sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

- **Bit 1 – RXB8n: Receive Data Bit 8 n**

RXB8n 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 UDRn.

- **Bit 0 – TXB8n: Transmit Data Bit 8 n**

TXB8n 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 UDRn.

19.10.4 UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
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	1	1	0	

- **Bits 7:6 – UMSELn1:0 USART Mode Select**

These bits select the mode of operation of the USARTn as shown in Table 19-4.

Table 19-4. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)(1)

Note: 1. See "USART in SPI Mode" on page 204 for full description of the Master SPI Mode (MSPIM) operation

- **Bits 5:4 – UPMn1:0: 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 UPMn setting. If a mismatch is detected, the UPEn Flag in UCSRnA will be set.

Table 19-5. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

- **Bit 3 – USBSn: Stop Bit Select**

This bit selects the number of stop bits to be inserted by the Transmitter. The Receiver ignores this setting.

Table 19-6. USBS Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bit

- **Bit 2:1 – UCSZn1:0: Character Size**

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

Table 19-7. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	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 – UCPOLn: Clock Polarity**

This bit is used for synchronous mode only. Write this bit to zero when asynchronous mode is used. The UCPOLn bit sets the relationship between data output change and data input sample, and the synchronous clock (XCKn).

Table 19-8. UCPOLn Bit Settings

UCPOLn	Transmitted Data Changed (Output of TxDn Pin)	Received Data Sampled (Input on RxDn Pin)
0	Rising XCKn Edge	Falling XCKn Edge
1	Falling XCKn Edge	Rising XCKn Edge

19.10.5 UBRRnL and UBRRnH – USART Baud Rate Registers

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRRn[11:8]				UBRRnH
	UBRRn[7:0]								UBRRnL
	7	6	5	4	3	2	1	0	
Read/Write	R	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	

- **Bit 15:12 – Reserved Bits**

These bits are reserved for future use. For compatibility with future devices, these bit must be written to zero when UBRRnH is written.

- **Bit 11:0 – UBRR11:0: USART Baud Rate Register**

This is a 12-bit register which contains the USART baud rate. The UBRRnH contains the four most significant bits, and the UBRRnL contains the eight 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 UBRRnL will trigger an immediate update of the baud rate prescaler.

19.11 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 UBRRn settings in Table 19-9. UBRRn 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 192). 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 19-9. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	f_{osc} = 1.0000 MHz				f_{osc} = 1.8432 MHz				f_{osc} = 2.0000 MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	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	

Note: 1. UBRRn = 0, Error = 0.0%

Table 19-10. Examples of UBRRn 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			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	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. UBRRn = 0, Error = 0.0%

Table 19-11. Examples of UBRRn 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			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	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. UBRRn = 0, Error = 0.0%

Table 19-12. Examples of UBRRn 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			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	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. UBRRn = 0, Error = 0.0%

20. USART in SPI Mode

20.1 Features

- **Full Duplex, Three-wire Synchronous Data Transfer**
- **Master Operation**
- **Supports all four SPI Modes of Operation (Mode 0, 1, 2, and 3)**
- **LSB First or MSB First Data Transfer (Configurable Data Order)**
- **Queued Operation (Double Buffered)**
- **High Resolution Baud Rate Generator**
- **High Speed Operation ($f_{XCKmax} = f_{CK}/2$)**
- **Flexible Interrupt Generation**

20.2 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) can be set to a master SPI compliant mode of operation.

Setting both UMSELn1:0 bits to one enables the USART in MSPIM logic. In this mode of operation the SPI master control logic takes direct control over the USART resources. These resources include the transmitter and receiver shift register and buffers, and the baud rate generator. The parity generator and checker, the data and clock recovery logic, and the RX and TX control logic is disabled. The USART RX and TX control logic is replaced by a common SPI transfer control logic. However, the pin control logic and interrupt generation logic is identical in both modes of operation.

The I/O register locations are the same in both modes. However, some of the functionality of the control registers changes when using MSPIM.

20.3 Clock Generation

The Clock Generation logic generates the base clock for the Transmitter and Receiver. For USART MSPIM mode of operation only internal clock generation (i.e. master operation) is supported. The Data Direction Register for the XCKn pin (DDR_XCKn) must therefore be set to one (i.e. as output) for the USART in MSPIM to operate correctly. Preferably the DDR_XCKn should be set up before the USART in MSPIM is enabled (i.e. TXENn and RXENn bit set to one).

The internal clock generation used in MSPIM mode is identical to the USART synchronous master mode. The baud rate or UBRRn setting can therefore be calculated using the same equations, see Table 20-1:

Table 20-1. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRRn Value
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(\boldsymbol{UBRR}n + 1)}$	$\boldsymbol{UBRR}n = \frac{f_{OSC}}{2BAUD} - 1$

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

UBRRn Contents of the UBRRnH and UBRRnL Registers, (0-4095)

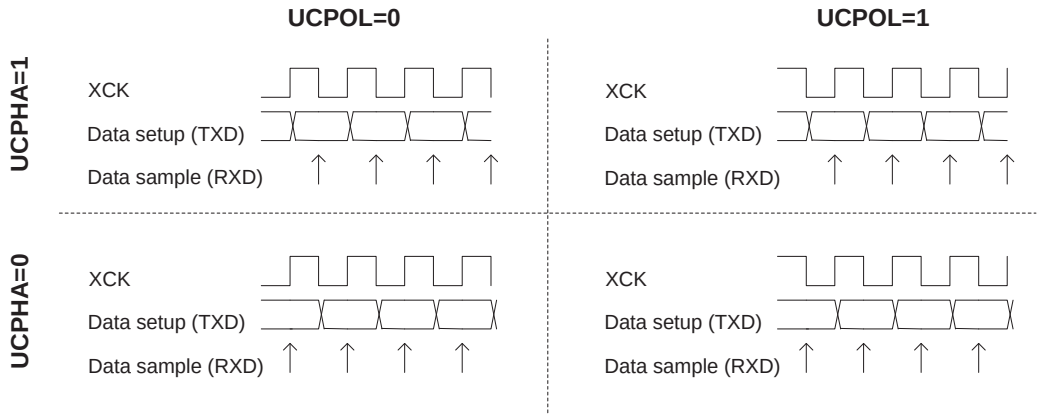
20.4 SPI Data Modes and Timing

There are four combinations of XCKn (SCK) phase and polarity with respect to serial data, which are determined by control bits UCPHAn and UCPOLn. The data transfer timing diagrams are shown in Figure 20-1. Data bits are shifted out and latched in on opposite edges of the XCKn signal, ensuring sufficient time for data signals to stabilize. The UCPOLn and UCPHAn functionality is summarized in Table 20-2. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter.

Table 20-2. UCPOLn and UCPHAn Functionality-

UCPOLn	UCPHAn	SPI Mode	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
0	1	1	Setup (Rising)	Sample (Falling)
1	0	2	Sample (Falling)	Setup (Rising)
1	1	3	Setup (Falling)	Sample (Rising)

Figure 20-1. UCPHAn and UCPOLn data transfer timing diagrams.



20.5 Frame Formats

A serial frame for the MSPIM is defined to be one character of 8 data bits. The USART in MSPIM mode has two valid frame formats:

- 8-bit data with MSB first
- 8-bit data with LSB first

A frame starts with the least or most significant data bit. Then the next data bits, up to a total of eight, are succeeding, ending with the most or least significant bit accordingly. When a complete frame is transmitted, a new frame can directly follow it, or the communication line can be set to an idle (high) state.

The UDORDn bit in UCSRnC sets the frame format used by the USART in MSPIM mode. 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.

16-bit data transfer can be achieved by writing two data bytes to UDRn. A UART transmit complete interrupt will then signal that the 16-bit value has been shifted out.

20.5.1 USART MSPIM Initialization

The USART in MSPIM mode has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting master mode of operation (by setting DDR_XCKn to one), setting frame format and enabling the Transmitter and the Receiver. Only the transmitter can operate independently. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and thus interrupts globally disabled) when doing the initialization.

Note: To ensure immediate initialization of the XCKn output the baud-rate register (UBRRn) must be zero at the time the transmitter is enabled. Contrary to the normal mode USART operation the UBRRn must then be written to the desired value after the transmitter is enabled, but before the first transmission is started. Setting UBRRn to zero before enabling the transmitter is not necessary if the initialization is done immediately after a reset since UBRRn is reset to zero.

Before doing a re-initialization with changed baud rate, data mode, or frame format, be sure that there is no ongoing transmissions during the period the registers are changed. The TXCn Flag can be used to check that the Transmitter has completed all transfers, and the RXCn Flag can

be used to check that there are no unread data in the receive buffer. Note that the TXCn Flag must be cleared before each transmission (before UDRn 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 polling (no interrupts enabled). 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.

Assembly Code Example[1]

```
USART_Init:
  clr r18
  out UBRRnH,r18
  out UBRRnL,r18
  ; Setting the XCKn port pin as output, enables master mode.
  sbi XCKn_DDR, XCKn
  ; Set MSPI mode of operation and SPI data mode 0.
  ldi r18, (1<<UMSELn1)|(1<<UMSELn0)|(0<<UCPHAn)|(0<<UCPOLn)
  out UCSRnC,r18
  ; Enable receiver and transmitter.
  ldi r18, (1<<RXENn)|(1<<TXENn)
  out UCSRnB,r18
  ; Set baud rate.
  ; IMPORTANT: The Baud Rate must be set after the transmitter is enabled!
  out UBRRnH, r17
  out UBRRnL, r18
  ret
```

C Code Example[1]

```
void USART_Init( unsigned int baud )
{
  UBRRn = 0;
  /* Setting the XCKn port pin as output, enables master mode. */
  XCKn_DDR |= (1<<XCKn);
  /* Set MSPI mode of operation and SPI data mode 0. */
  UCSRnC = (1<<UMSELn1)|(1<<UMSELn0)|(0<<UCPHAn)|(0<<UCPOLn);
  /* Enable receiver and transmitter. */
  UCSRnB = (1<<RXENn)|(1<<TXENn);
  /* Set baud rate. */
  /* IMPORTANT: The Baud Rate must be set after the transmitter is enabled
  */
  UBRRn = baud;
}
```

Note: 1. See "About Code Examples" on page 7.

20.6 Data Transfer

Using the USART in MSPI mode requires the Transmitter to be enabled, i.e. the TXENn bit in the UCSRnB register is set to one. When the Transmitter is enabled, the normal port operation of the TxDn pin is overridden and given the function as the Transmitter's serial output. Enabling the receiver is optional and is done by setting the RXENn bit in the UCSRnB register to one. When the receiver is enabled, the normal pin operation of the RxDn pin is overridden and given the function as the Receiver's serial input. The XCKn will in both cases be used as the transfer clock.

After initialization the USART is ready for doing data transfers. A data transfer is initiated by writing to the UDRn I/O location. This is the case for both sending and receiving data since the transmitter controls the transfer clock. The data written to UDRn is moved from the transmit buffer to the shift register when the shift register is ready to send a new frame.

Note: To keep the input buffer in sync with the number of data bytes transmitted, the UDRn register must be read once for each byte transmitted. The input buffer operation is identical to normal USART mode, i.e. if an overflow occurs the character last received will be lost, not the first data in the buffer. This means that if four bytes are transferred, byte 1 first, then byte 2, 3, and 4, and the UDRn is not read before all transfers are completed, then byte 3 to be received will be lost, and not byte 1.

The following code examples show a simple USART in MSPIM mode transfer function based on polling of the Data Register Empty (UDREn) Flag and the Receive Complete (RXCn) Flag. 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 and the data received will be available in the same register (R16) after the function returns.

The function simply waits for the transmit buffer to be empty by checking the UDREn Flag, before loading it with new data to be transmitted. The function then waits for data to be present in the receive buffer by checking the RXCn Flag, before reading the buffer and returning the value.

Assembly Code Example[1]

```
USART_MSPIM_Transfer:
  ; Wait for empty transmit buffer
  sbis UCSRnA, UDREn
  rjmp USART_MSPIM_Transfer
  ; Put data (r16) into buffer, sends the data
  out UDRn,r16
  ; Wait for data to be received
USART_MSPIM_Wait_RXCn:
  sbis UCSRnA, RXCn
  rjmp USART_MSPIM_Wait_RXCn
  ; Get and return received data from buffer
  in r16, UDRn
  ret
```

C Code Example[1]

```
unsigned char USART_Receive( void )
{
  /* Wait for empty transmit buffer */
  while ( !( UCSRnA & (1<<UDREn)) );
  /* Put data into buffer, sends the data */
  UDRn = data;
  /* Wait for data to be received */
  while ( !(UCSRnA & (1<<RXCn)) );
  /* Get and return received data from buffer */
  return UDRn;
}
```

Note: 1. See "About Code Examples" on page 7.

20.6.1 Transmitter and Receiver Flags and Interrupts

The RXCn, TXCn, and UDREn flags and corresponding interrupts in USART in MSPIM mode are identical in function to the normal USART operation. However, the receiver error status flags (FE, DOR, and PE) are not in use and is always read as zero.

20.6.2 Disabling the Transmitter or Receiver

The disabling of the transmitter or receiver in USART in MSPIM mode is identical in function to the normal USART operation.

20.7 AVR USART MSPIM vs. AVR SPI

The USART in MSPIM mode is fully compatible with the AVR SPI regarding:

- Master mode timing diagram.
- The UCPOLn bit functionality is identical to the SPI CPOL bit.
- The UCPHAn bit functionality is identical to the SPI CPHA bit.
- The UDORDn bit functionality is identical to the SPI DORD bit.

However, since the USART in MSPIM mode reuses the USART resources, the use of the USART in MSPIM mode is somewhat different compared to the SPI. In addition to differences of the control register bits, and that only master operation is supported by the USART in MSPIM mode, the following features differ between the two modules:

- The USART in MSPIM mode includes (double) buffering of the transmitter. The SPI has no buffer.
- The USART in MSPIM mode receiver includes an additional buffer level.
- The SPI WCOL (Write Collision) bit is not included in USART in MSPIM mode.
- The SPI double speed mode (SPI2X) bit is not included. However, the same effect is achieved by setting UBRRn accordingly.
- Interrupt timing is not compatible.
- Pin control differs due to the master only operation of the USART in MSPIM mode.

A comparison of the USART in MSPIM mode and the SPI pins is shown in Table 20-3 on page 210.

Table 20-3. Comparison of USART in MSPIM mode and SPI pins.

USART_MSPIM	SPI	Comment
TxDn	MOSI	Master Out only
RxDn	MISO	Master In only
XCKn	SCK	(Functionally identical)
(N/A)	$\overline{SS}$	Not supported by USART in MSPIM

20.8 Register Description

The following section describes the registers used for SPI operation using the USART.

20.8.1 UDRn – USART MSPIM I/O Data Register

The function and bit description of the USART data register (UDRn) in MSPI mode is identical to normal USART operation. See “UDRn – USART I/O Data Register n” on page 195.

20.8.2 UCSRnA – USART MSPIM Control and Status Register n A

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	-	-	-	-	-	UCSRnA
Read/Write	R	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 7 - RXCn: 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 (i.e., does not contain any unread data). If the Receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn Flag can be used to generate a Receive Complete interrupt (see description of the RXCIEn bit).

- **Bit 6 - TXCn: 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 (UDRn). The TXCn 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 TXCn Flag can generate a Transmit Complete interrupt (see description of the TXCIEn bit).

- **Bit 5 - UDREn: USART Data Register Empty**

The UDREn Flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREn is one, the buffer is empty, and therefore ready to be written. The UDREn Flag can generate a Data Register Empty interrupt (see description of the UDRIE bit). UDREn is set after a reset to indicate that the Transmitter is ready.

- **Bit 4:0 - Reserved Bits in MSPI mode**

When in MSPI mode, these bits are reserved for future use. For compatibility with future devices, these bits must be written to zero when UCSRnA is written.

20.8.3 UCSRnB – USART MSPIM Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIE	RXENn	TXENn	-	-	-	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 7 - RXCIEn: RX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the RXCn Flag. A USART Receive Complete interrupt will be generated only if the RXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXCn bit in UCSRnA is set.

- **Bit 6 - TXCIEn: TX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the TXCn Flag. A USART Transmit Complete interrupt will be generated only if the TXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXCn bit in UCSRnA is set.

- **Bit 5 - UDRIE: USART Data Register Empty Interrupt Enable**

Writing this bit to one enables interrupt on the UDREn 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 UDREn bit in UCSRnA is set.

- **Bit 4 - RXENn: Receiver Enable**

Writing this bit to one enables the USART Receiver in MSPIM mode. The Receiver will override normal port operation for the RxDn pin when enabled. Disabling the Receiver will flush the receive buffer. Only enabling the receiver in MSPI mode (i.e. setting RXENn=1 and TXENn=0) has no meaning since it is the transmitter that controls the transfer clock and since only master mode is supported.

- **Bit 3 - TXENn: Transmitter Enable**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxDn pin when enabled. The disabling of the Transmitter (writing TXENn to zero) will not become effective until ongoing and pending transmissions are completed, i.e., 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 TxDn port.

- **Bit 2:0 - Reserved Bits in MSPI mode**

When in MSPI mode, these bits are reserved for future use. For compatibility with future devices, these bits must be written to zero when UCSRnB is written.

20.8.4 UCSRnC – USART MSPIM Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	-	-	-	UDORDn	UCPHAn	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 7:6 - UMSELn1:0: USART Mode Select**

These bits select the mode of operation of the USART as shown in Table 20-4. See "UCSRnC – USART Control and Status Register n C" on page 197 for full description of the normal USART operation. The MSPIM is enabled when both UMSELn bits are set to one. The UDORDn, UCPHAn, and UCPOLn can be set in the same write operation where the MSPIM is enabled.

Table 20-4. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	Reserved
1	1	Master SPI (MSPIM)

- **Bit 5:3 - Reserved Bits in MSPI mode**

When in MSPI mode, these bits are reserved for future use. For compatibility with future devices, these bits must be written to zero when UCSRnC is written.

- **Bit 2 - UDORDn: Data Order**

When set to one the LSB of the data word is transmitted first. When set to zero the MSB of the data word is transmitted first. Refer to the Frame Formats section page 4 for details.

- **Bit 1 - UCPHAn: Clock Phase**

The UCPHAn bit setting determine if data is sampled on the leasing edge (first) or tailing (last) edge of XCKn. Refer to the SPI Data Modes and Timing section page 4 for details.

- **Bit 0 - UCPOLn: Clock Polarity**

The UCPOLn bit sets the polarity of the XCKn clock. The combination of the UCPOLn and UCPHAn bit settings determine the timing of the data transfer. Refer to the SPI Data Modes and Timing section page 4 for details.

20.8.5 USART MSPIM Baud Rate Registers - UBRRnL and UBRRnH

The function and bit description of the baud rate registers in MSPI mode is identical to normal USART operation. See “UBRRnL and UBRRnH – USART Baud Rate Registers” on page 199.

21. 2-wire Serial Interface

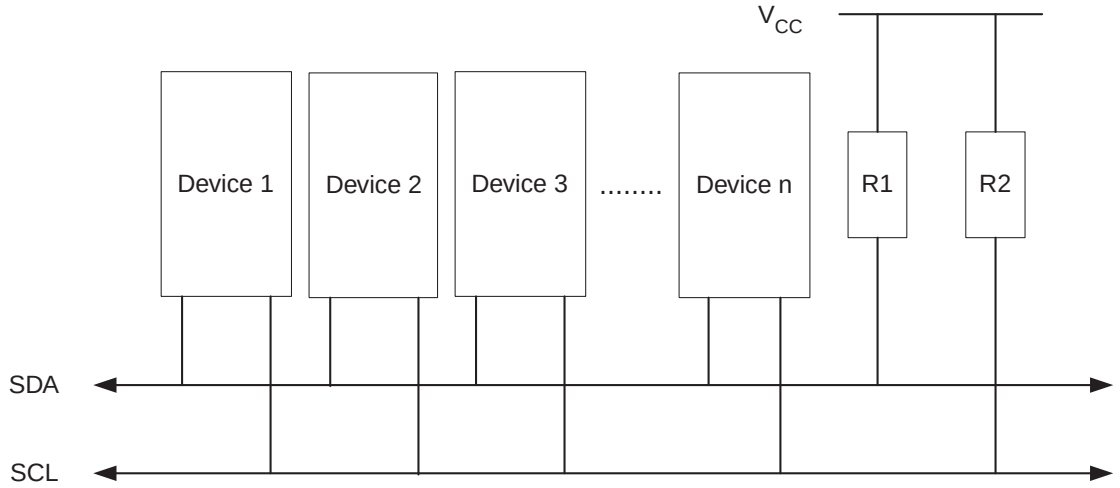
21.1 Features

- **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**
- **Compatible with Philips' I^2C protocol**

21.2 2-wire Serial Interface Bus Definition

The 2-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 21-1. TWI Bus Interconnection



21.2.1 TWI Terminology

The following definitions are frequently encountered in this section.

Table 21-1. 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.

The PRTWI bit in ”Minimizing Power Consumption” on page 42 must be written to zero to enable the 2-wire Serial Interface.

21.2.2 Electrical Interconnection

As depicted in Figure 21-1, 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.

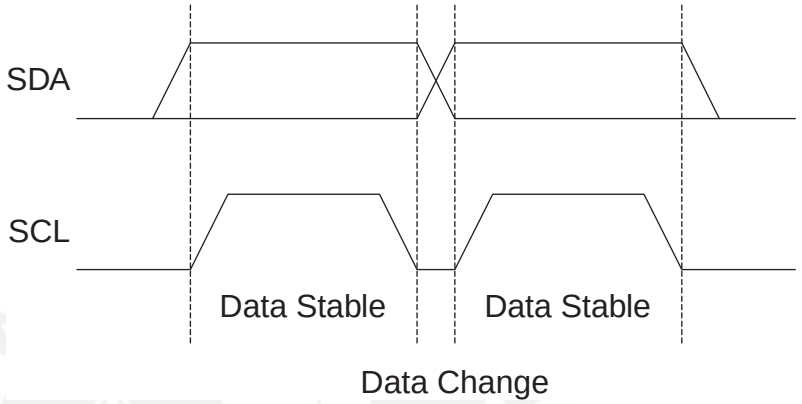
The number of devices that can be connected to the bus is only limited by the bus capacitance limit of 400 pF and the 7-bit slave address space. A detailed specification of the electrical characteristics of the TWI is given in ”2-wire Serial Interface Characteristics” on page 321. Two different sets of specifications are presented there, one relevant for bus speeds below 100 kHz, and one valid for bus speeds up to 400 kHz.

21.3 Data Transfer and Frame Format

21.3.1 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.

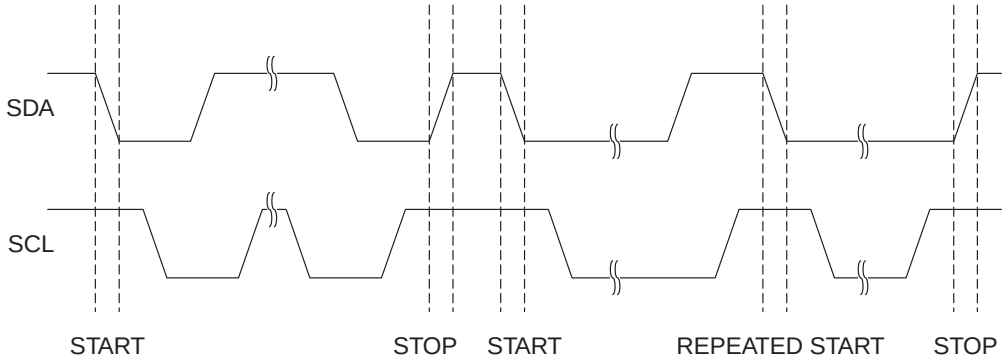
Figure 21-2. Data Validity



21.3.2 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 relinquishing 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 21-3. START, REPEATED START and STOP conditions



21.3.3 Address Packet Format

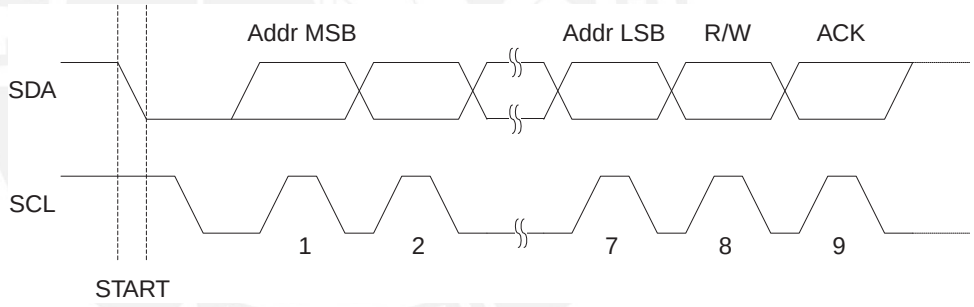
All address packets transmitted on the TWI bus are 9 bits long, consisting of 7 address bits, one READ/WRITE control bit and an acknowledge bit. If the READ/WRITE bit is set, a read operation is to be performed, otherwise a write operation should be performed. When a Slave recognizes that it is being 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.

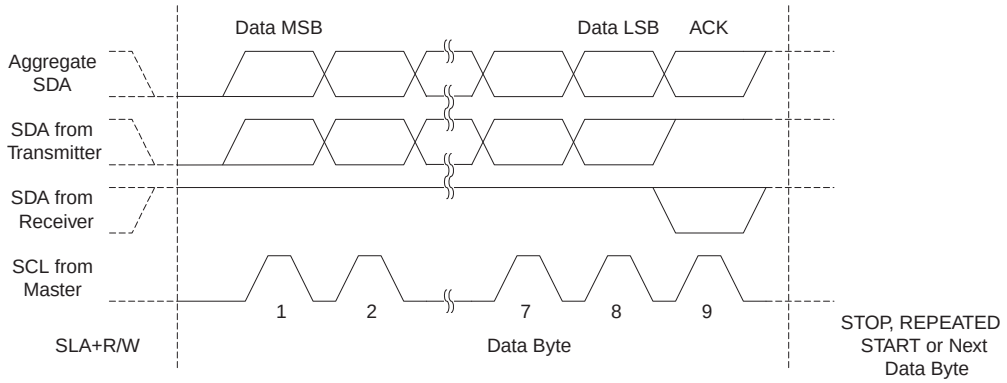
Figure 21-4. Address Packet Format



21.3.4 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 21-5. Data Packet Format

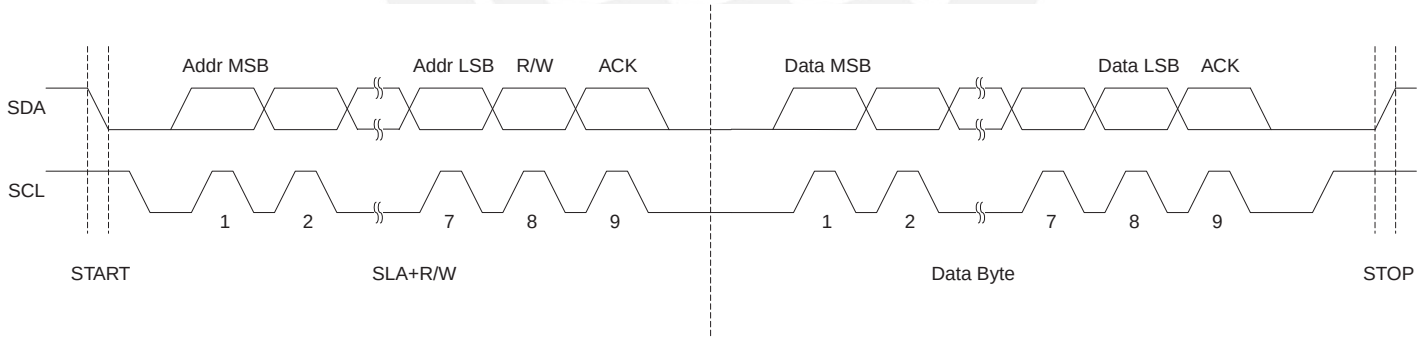


21.3.5 Combining Address and Data Packets into a Transmission

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 21-6 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 21-6. Typical Data Transmission



21.4 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:

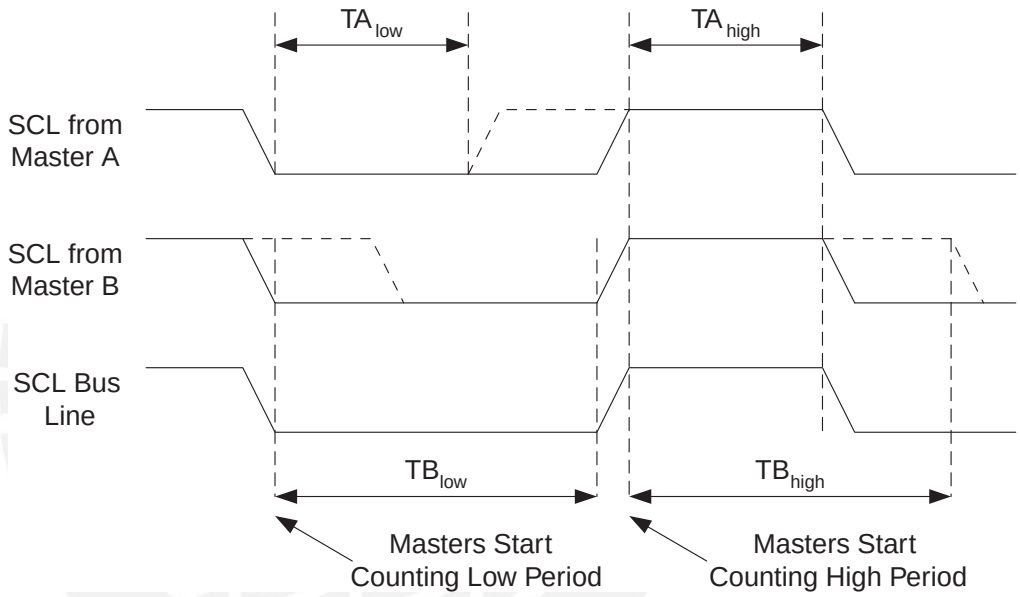
- An algorithm must be implemented allowing only one of the masters to complete the transmission. All other masters should cease transmission when they discover that they have lost the selection process. This selection process is called arbitration. When a contending master discovers that it has lost 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, i.e. 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.

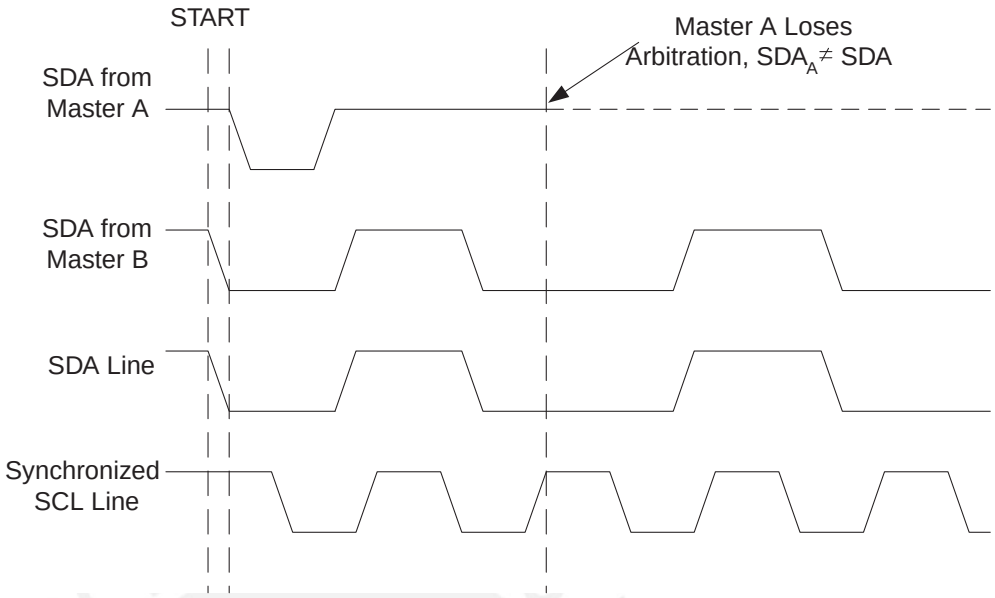
The wired-ANDing of the bus lines is used to solve both these problems. The serial clocks from all masters will be wired-ANDed, yielding a combined clock with a high period equal to the one from the Master with the shortest high period. The low period of the combined clock is equal to the low period 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 21-7. 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 21-8. Arbitration Between Two Masters



Note that arbitration is not allowed between:

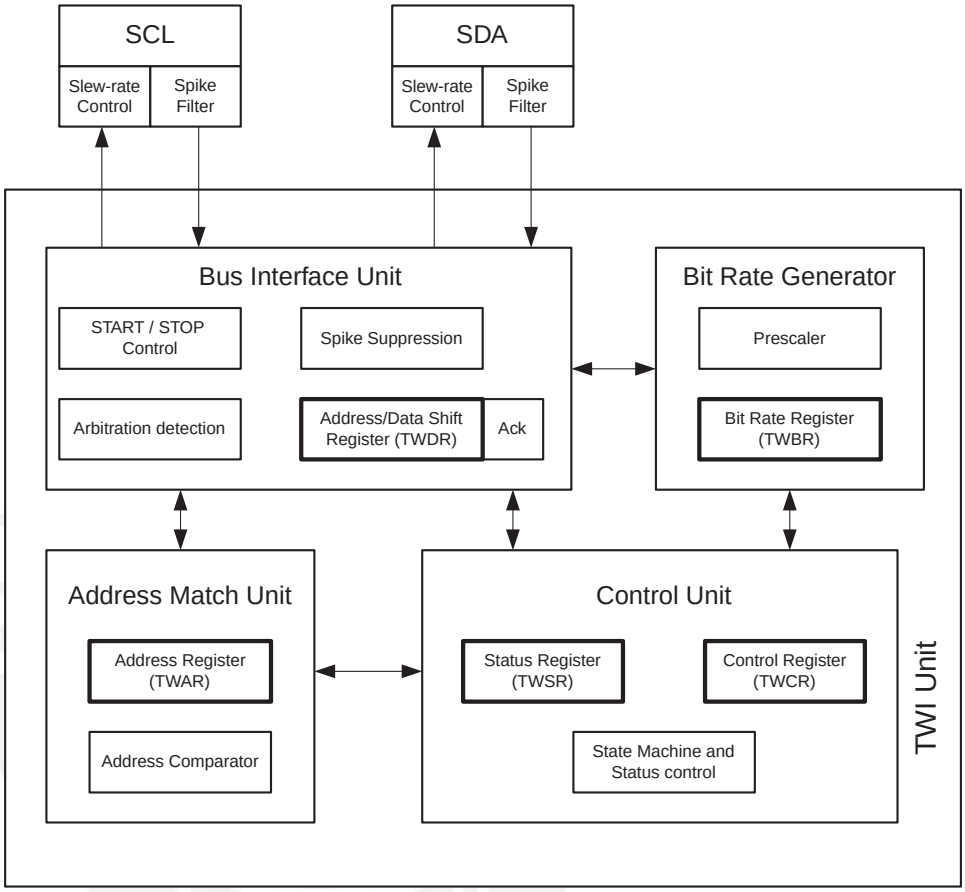
- A REPEATED START condition and a data bit.
- A STOP condition and a data bit.
- A REPEATED START and a STOP condition.

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.

21.5 Overview of the TWI Module

The TWI module is comprised of several submodules, as shown in Figure 21-9. All registers drawn in a thick line are accessible through the AVR data bus.

Figure 21-9. Overview of the TWI Module



21.5.1 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.

21.5.2 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:

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot (PrescalerValue)}$$

- TWBR = Value of the TWI Bit Rate Register.
- *PrescalerValue* = Value of the prescaler, see Table 21-7 on page 243.

Note: Pull-up resistor values should be selected according to the SCL frequency and the capacitive bus line load. See Table 28-6 on page 321 for value of pull-up resistor.

21.5.3 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.

21.5.4 Address Match Unit

The Address Match unit checks if received address bytes match the seven-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.

If another interrupt (e.g., INT0) occurs during TWI Power-down address match and wakes up the CPU, the TWI aborts operation and return to it's idle state. If this cause any problems, ensure that TWI Address Match is the only enabled interrupt when entering Power-down[(1)].

Note: 1. This applies to all device revisions except ATmega88PA revision C or newer.

21.5.5 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 avail-

able. 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.

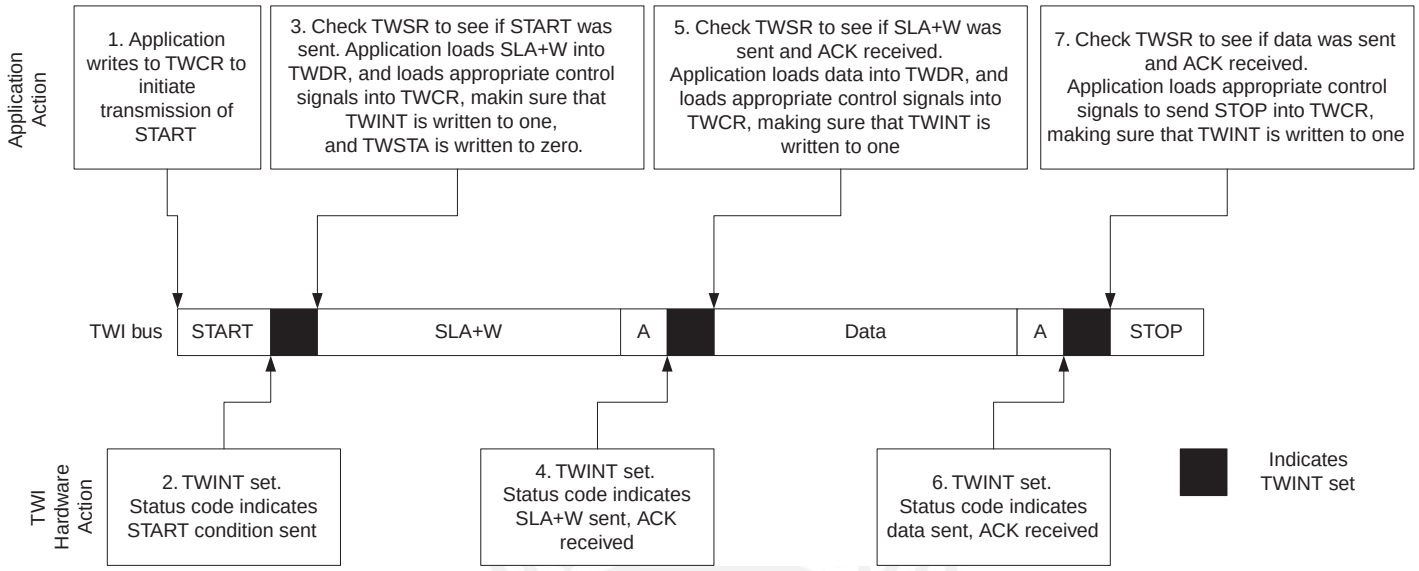
21.6 Using the TWI

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 21-10 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 21-10. 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

21.7 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

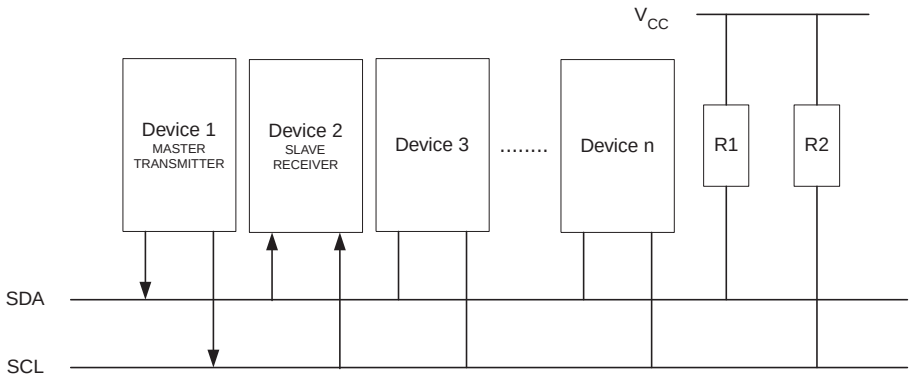
In Figure 21-12 to Figure 21-18, circles are used to indicate that the TWINT Flag is set. The numbers in the circles show the status code held in TWSR, with the prescaler bits masked to zero. At these points, actions must be taken by the application to continue or complete the TWI transfer. The TWI transfer 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 21-2 to Table 21-5. Note that the prescaler bits are masked to zero in these tables.

21.7.1 Master Transmitter Mode

In the Master Transmitter mode, a number of data bytes are transmitted to a Slave Receiver (see Figure 21-11). 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 21-11. 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 2-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 2-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 0x08 (see Table 21-2). 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 0x18, 0x20, or 0x38. The appropriate action to be taken for each of these status codes is detailed in Table 21-2.

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:

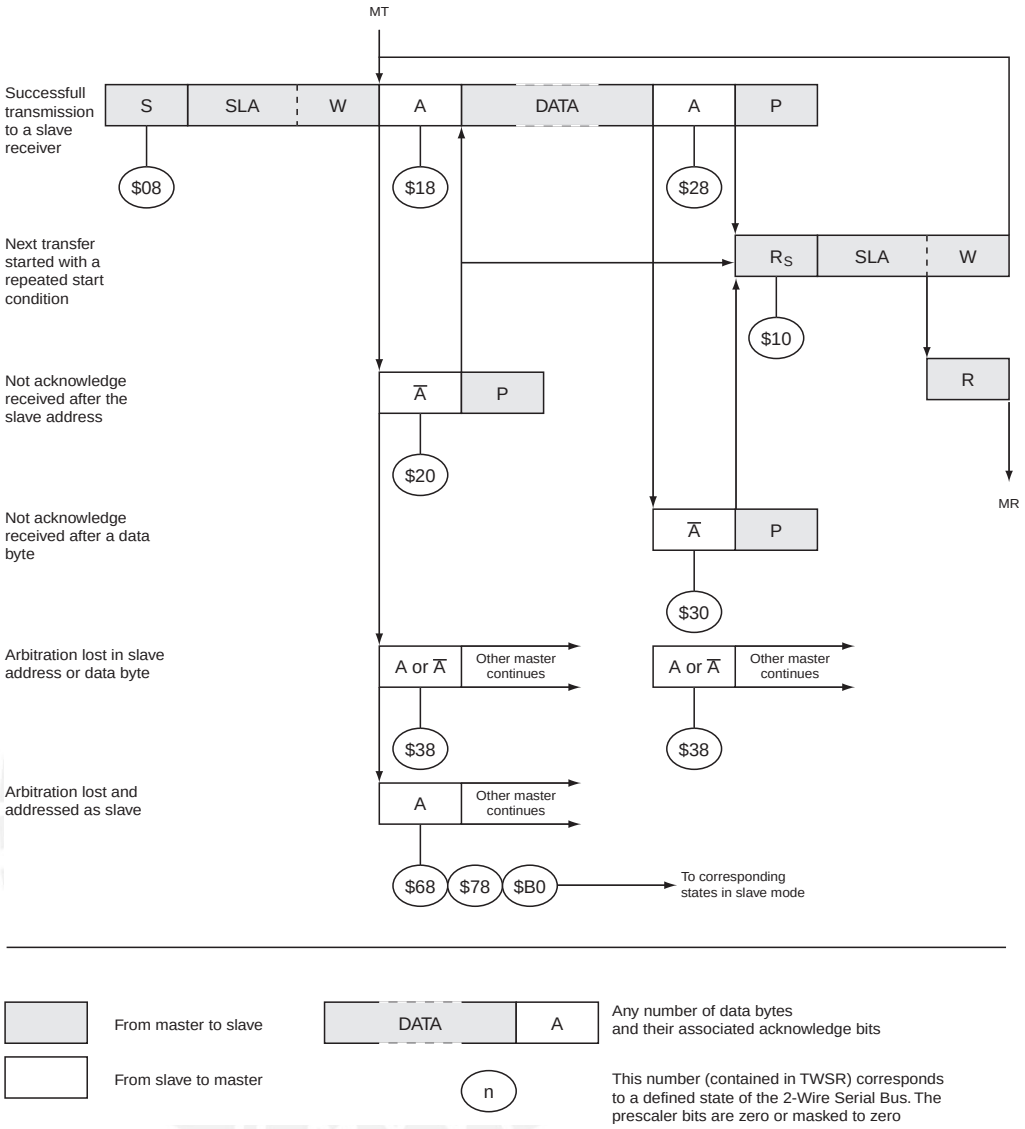
TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	1	0	X	1	0	X

After a repeated START condition (state 0x10) the 2-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 21-2. Status codes for Master Transmitter Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0x08	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
0x10	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
0x18	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
0x20	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
0x28	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
0x30	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
0x38	Arbitration lost in SLA+W or data bytes	No TWDR action or	0	0	1	X	2-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

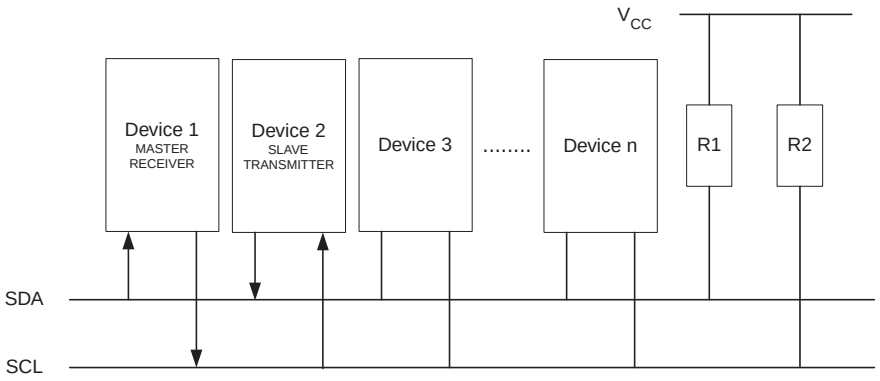
Figure 21-12. Formats and States in the Master Transmitter Mode



21.7.2 Master Receiver Mode

In the Master Receiver mode, a number of data bytes are received from a Slave Transmitter (Slave see Figure 21-13). 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 21-13. 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 2-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 2-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 0x08 (See Table 21-2). 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 0x38, 0x40, or 0x48. The appropriate action to be taken for each of these status codes is detailed in Table 21-3. 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

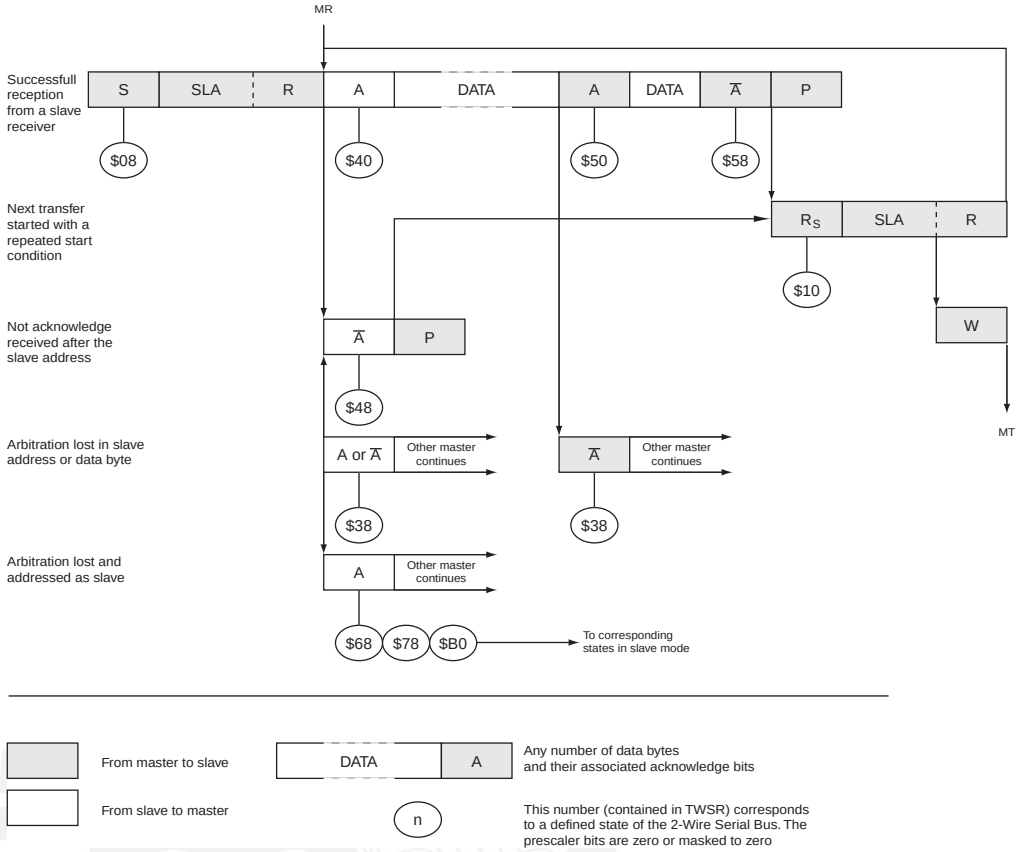
After a repeated START condition (state 0x10) the 2-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables

the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control over the bus.

Table 21-3. Status codes for Master Receiver Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0x08	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
0x10	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
0x38	Arbitration lost in SLA+R or NOT ACK bit	No TWDR action or	0	0	1	X	2-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
0x40	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
0x48	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
0x50	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
0x58	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

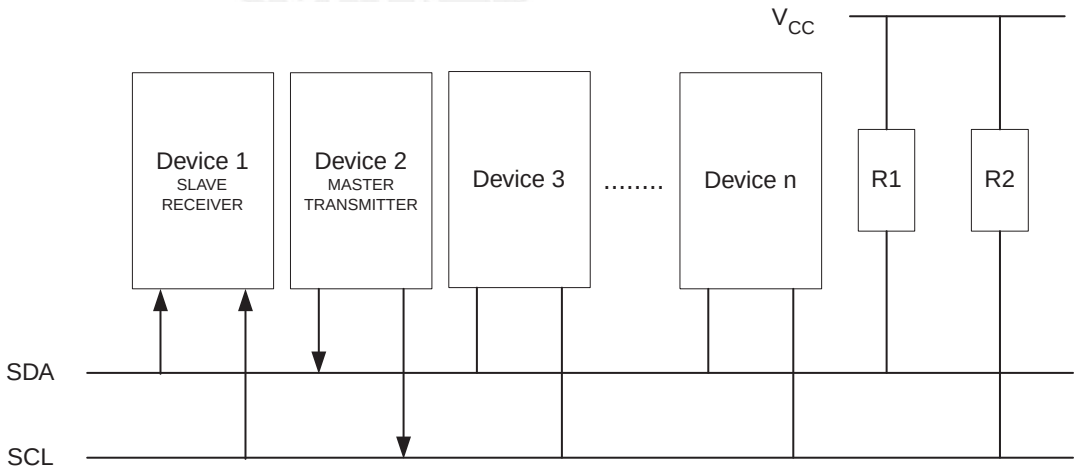
Figure 21-14. Formats and States in the Master Receiver Mode



21.7.3 Slave Receiver Mode

In the Slave Receiver mode, a number of data bytes are received from a Master Transmitter (see Figure 21-15). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 21-15. 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 7 bits are the address to which the 2-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address (0x00), 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 software action. The appropriate action to be taken for each status code is detailed in Table 21-4. The Slave Receiver mode may also be entered if arbitration is lost while the TWI is in the Master mode (see states 0x68 and 0x78).

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 2-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 2-wire Serial Bus.

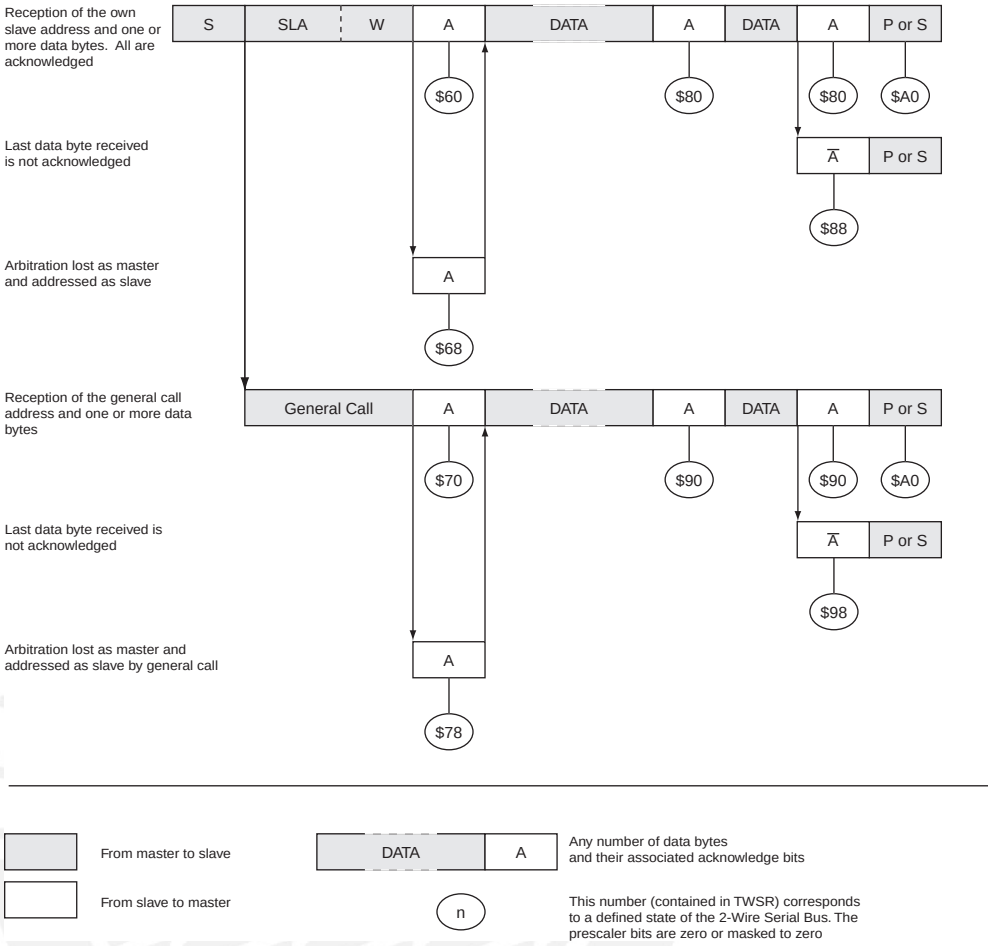
In all sleep modes other than Idle mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own slave address or the general call address by using the 2-wire Serial Bus clock as a clock source. The part will then wake up from sleep and the TWI will 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 2-wire Serial Interface Data Register – TWDR does not reflect the last byte present on the bus when waking up from these Sleep modes.

Table 21-4. Status Codes for Slave Receiver Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0x60	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
0x68	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
0x70	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
0x78	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
0x80	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
0x88	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
0x90	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
0x98	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
0xA0	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

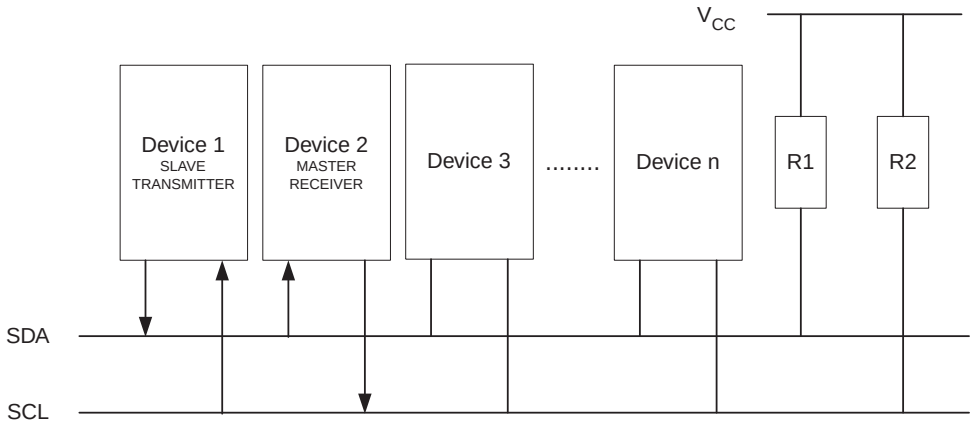
Figure 21-16. Formats and States in the Slave Receiver Mode



21.7.4 Slave Transmitter Mode

In the Slave Transmitter mode, a number of data bytes are transmitted to a Master Receiver (see Figure 21-17). All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 21-17. 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 2-wire Serial Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address (0x00), 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 21-5. The Slave Transmitter mode may also be entered if arbitration is lost while the TWI is in the Master mode (see state 0xB0).

If the TWEA bit is written to zero during a transfer, the TWI will transmit the last byte of the transfer. State 0xC0 or state 0xC8 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 0xC8 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 2-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 2-wire Serial Bus.

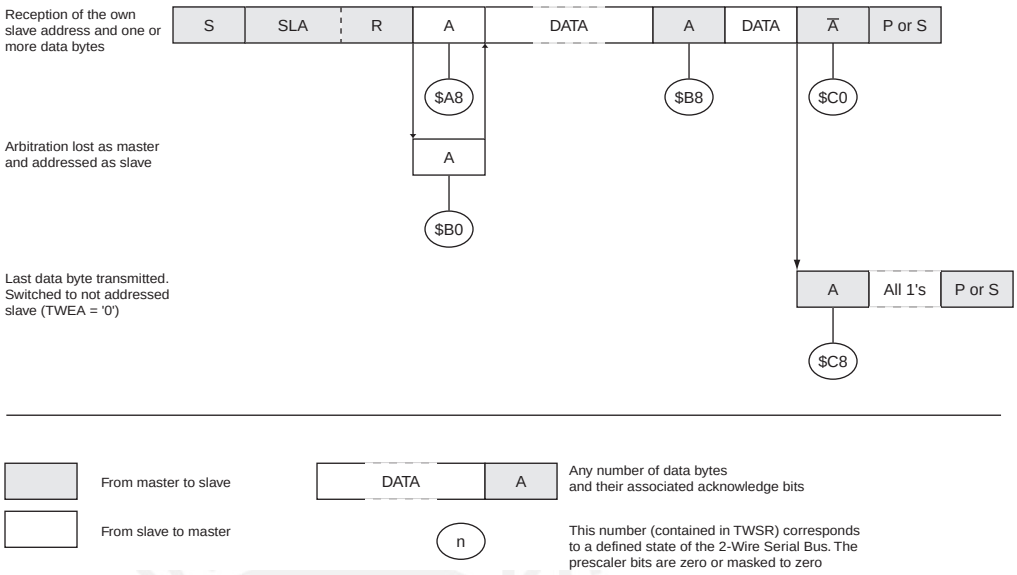
In all sleep modes other than Idle mode, the clock system to the TWI is turned off. If the TWEA bit is set, the interface can still acknowledge its own slave address or the general call address by using the 2-wire Serial Bus clock as a clock source. The part will then wake up from sleep and the TWI will 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 2-wire Serial Interface Data Register – TWDR does not reflect the last byte present on the bus when waking up from these sleep modes.

Table 21-5. Status Codes for Slave Transmitter Mode

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0xA8	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
0xB0	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
0xB8	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
0xC0	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
0xC8	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 21-18. Formats and States in the Slave Transmitter Mode



21.7.5 Miscellaneous States

There are two status codes that do not correspond to a defined TWI state, see Table 21-6.

Status 0xF8 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 0x00 indicates that a bus error has occurred during a 2-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 21-6. Miscellaneous States

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0xF8	No relevant state information available; TWINT = “0”	No TWDR action	No TWCR action				Wait or proceed current transfer
0x00	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.

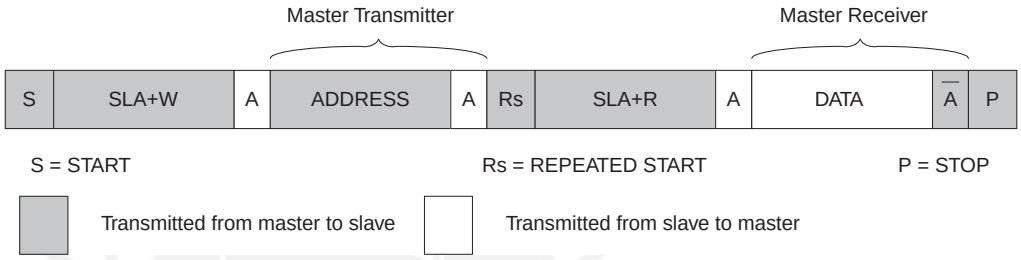
21.7.6 Combining Several TWI Modes

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.

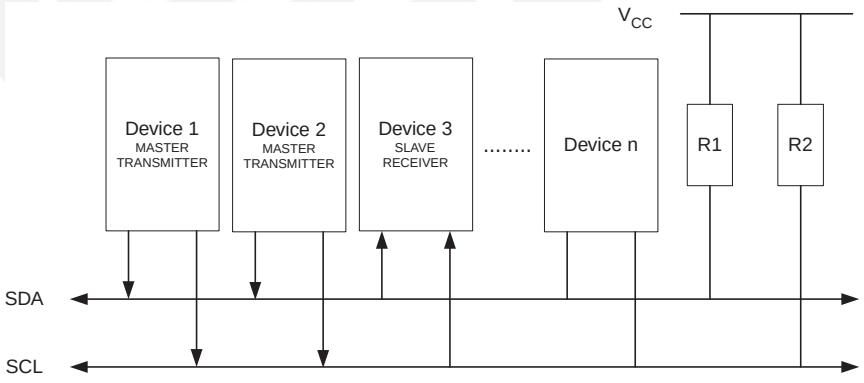
Figure 21-19. Combining Several TWI Modes to Access a Serial EEPROM



21.8 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 21-20. An Arbitration Example



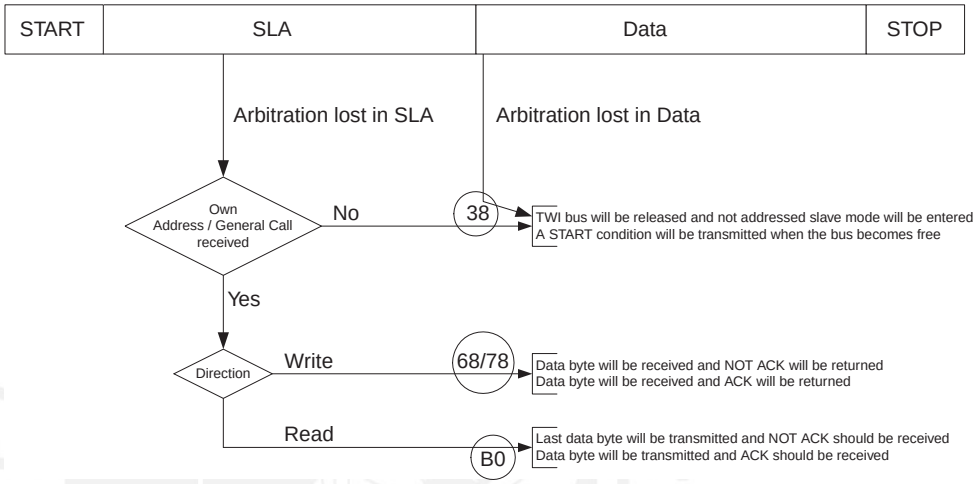
Several different scenarios may arise during arbitration, as described below:

- 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 21-21. Possible status values are given in circles.

Figure 21-21. Possible Status Codes Caused by Arbitration



21.9 Register Description

21.9.1 TWBR – TWI Bit Rate Register

Bit	7	6	5	4	3	2	1	0	
(0xB8)	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 221 for calculating bit rates.

21.9.2 TWCR – TWI Control Register

Bit	7	6	5	4	3	2	1	0	
(0xBC)	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 2-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 2-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 2-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.

21.9.3 TWSR – TWI Status Register

Bit	7	6	5	4	3	2	1	0	
(0xB9)	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 5 bits reflect the status of the TWI logic and the 2-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 21-7. 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 221. The value of TWPS1..0 is used in the equation.

21.9.4 TWDR – TWI Data Register

Bit	7	6	5	4	3	2	1	0	
(0xBB)	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 constitute the next data byte to be transmitted, or the latest data byte received on the 2-wire Serial Bus.

21.9.5 TWAR – TWI (Slave) Address Register

Bit	7	6	5	4	3	2	1	0	
(0xBA)	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, and not needed in the Master modes. 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 (0x00). 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 2-wire Serial Bus.

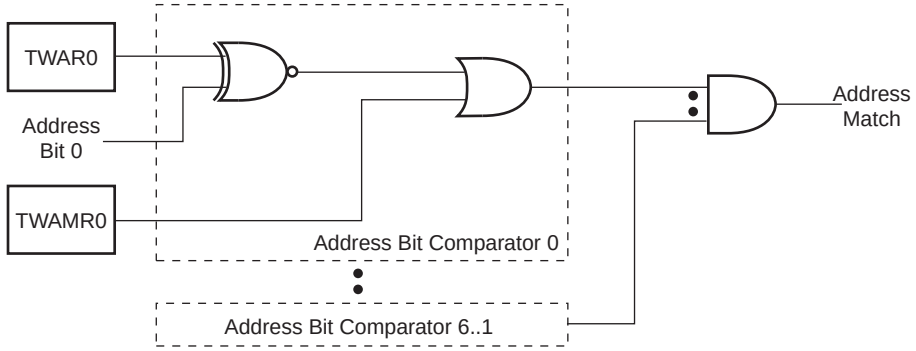
21.9.6 TWAMR – TWI (Slave) Address Mask Register

Bit	7	6	5	4	3	2	1	0	
(0xBD)	TWAM[6:0]							–	TWAMR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..1 – TWAM: TWI Address Mask**

The TWAMR can be loaded with a 7-bit Salve Address mask. Each of the bits in TWAMR can mask (disable) the corresponding address bits in the TWI Address Register (TWAR). If the mask bit is set to one then the address match logic ignores the compare between the incoming address bit and the corresponding bit in TWAR. Figure 21-22 shown the address match logic in detail.

Figure 21-22. TWI Address Match Logic, Block Diagram



- **Bit 0 – Res: Reserved Bit**

This bit is an unused bit in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

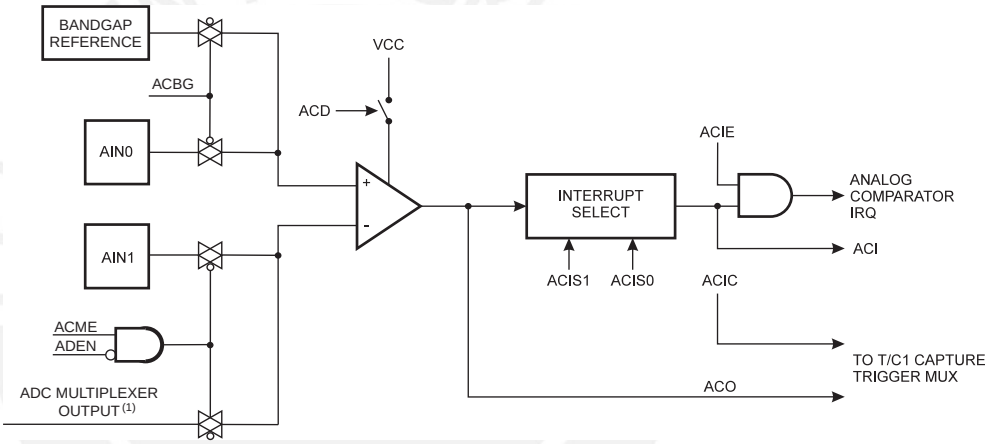
22. Analog Comparator

22.1 Overview

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 22-1.

The Power Reduction ADC bit, PRADC, in "Minimizing Power Consumption" on page 42 must be disabled by writing a logical zero to be able to use the ADC input MUX.

Figure 22-1. Analog Comparator Block Diagram(2)



Notes: 1. See Table 22-1 on page 247.
2. Refer to Figure 1-1 on page 2 and Table 13-9 on page 88 for Analog Comparator pin placement.

22.2 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 ADCSRB) 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 22-1. If ACME is cleared or ADEN is set, AIN1 is applied to the negative input to the Analog Comparator

.

Table 22-1. 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

22.3 Register Description

22.3.1 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – 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 246.

22.3.2 ACSR – Analog Comparator Control and Status Register

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	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. When the bandgap referance is used as input to the Analog Comparator, it will take a

certain time for the voltage to stabilize. If not stabilized, the first conversion may give a wrong value. See ”Internal Voltage Reference” on page 49

- **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 ICIE1 bit in the Timer Interrupt Mask Register (TIMSK1) 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 22-2.

Table 22-2. 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.

22.3.3 DIDR1 – Digital Input Disable Register 1

Bit	7	6	5	4	3	2	1	0	
(0x7F)	–	–	–	–	–	–	AIN1D	AIN0D	DIDR1
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..2 – Res: Reserved Bits**

These bits are unused bits in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bit 1, 0 – AIN1D, AIN0D: AIN1, AIN0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the AIN1/0 pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the AIN1/0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

23. Analog-to-Digital Converter

23.1 Features

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 µs Conversion Time
- Up to 76.9 kSPS (Up to 15 kSPS at Maximum Resolution)
- 6 Multiplexed Single Ended Input Channels
- 2 Additional Multiplexed Single Ended Input Channels (TQFP and QFN/MLF Package only)
- Temperature Sensor Input Channel
- Optional Left Adjustment for ADC Result Readout
- 0 - V_{CC} ADC Input Voltage Range
- Selectable 1.1V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

23.2 Overview

The ATmega48PA/88PA/168PA/328P features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).

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 23-1 on page 251.

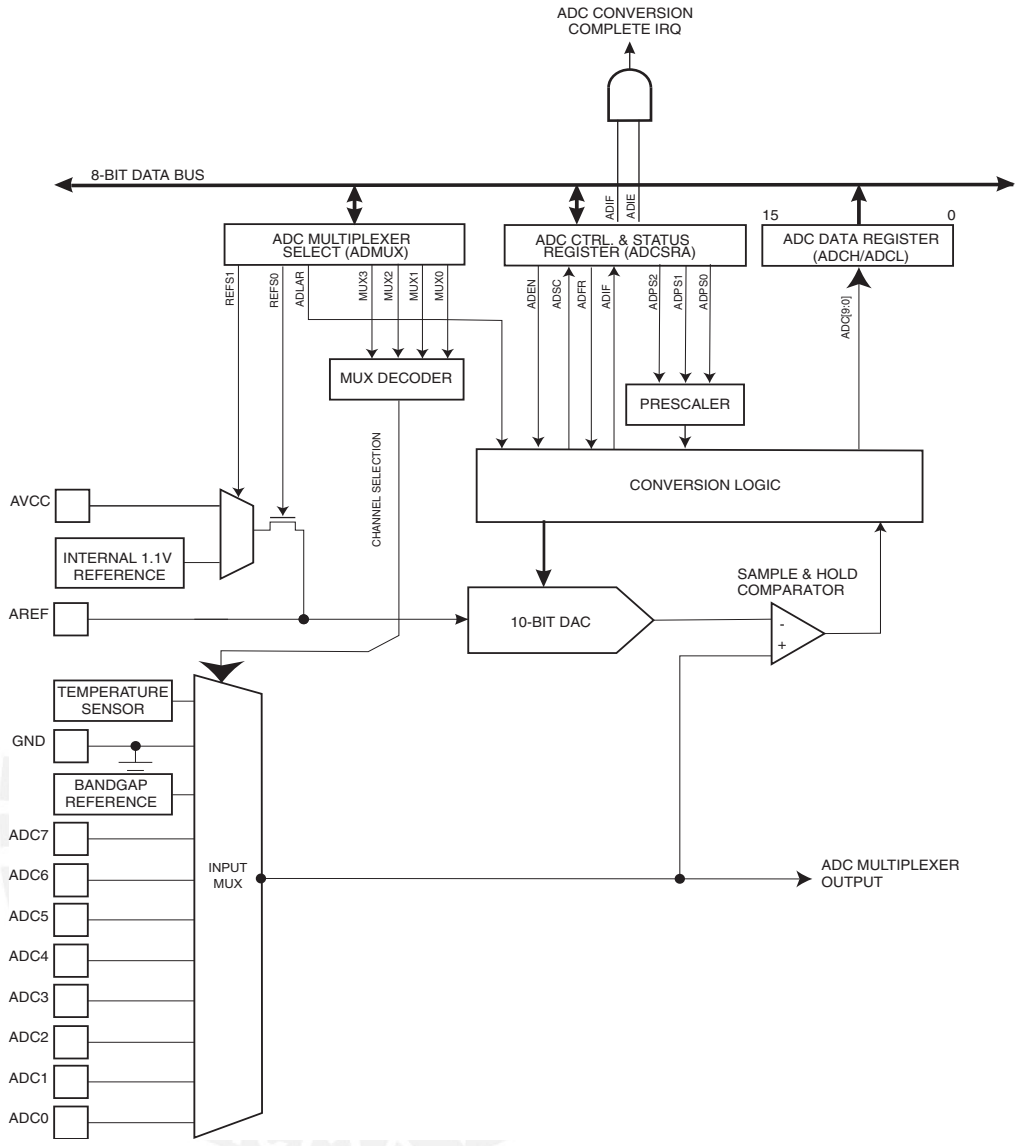
The ADC has a separate analog supply voltage pin, AV_{CC}. AV_{CC} must not differ more than ±0.3V from V_{CC}. See the paragraph ”ADC Noise Canceler” on page 256 on how to connect this pin.

Internal reference voltages of nominally 1.1V or AV_{CC} are provided On-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.

The Power Reduction ADC bit, PRADC, in ”Minimizing Power Consumption” on page 42 must be disabled by writing a logical zero to enable the ADC.

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, AV_{CC} or an internal 1.1V 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.

Figure 23-1. Analog to Digital Converter Block Schematic Operation,



The analog input channel is 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. 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.

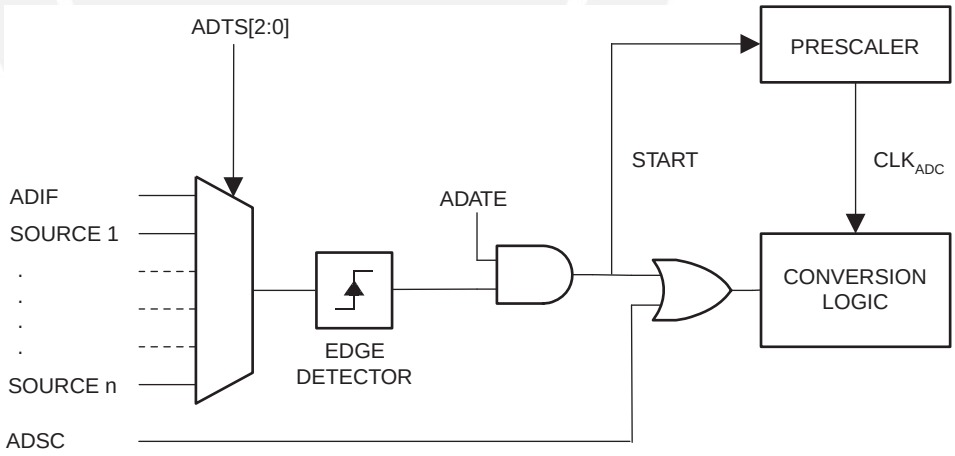
The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the Data Registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

23.3 Starting a Conversion

A single conversion is started by disabling the Power Reduction ADC bit, PRADC, in "Minimizing Power Consumption" on page 42 by writing a logical zero to it and 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 ADCSRB (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 23-2. ADC Auto Trigger Logic

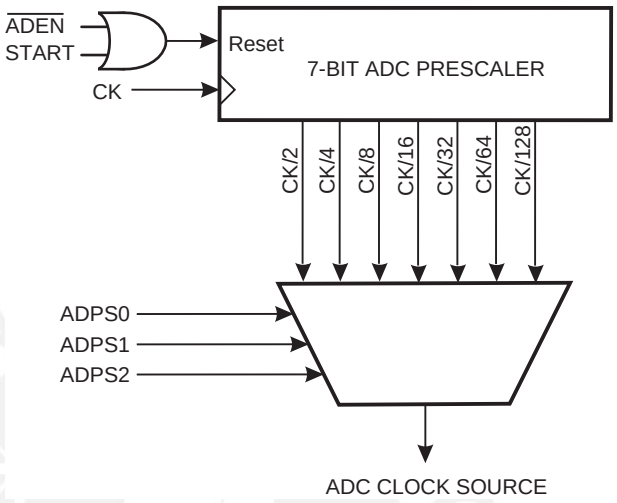


Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the 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.

23.4 Prescaling and Conversion Timing

Figure 23-3. 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.

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.

When the bandgap reference voltage is used as input to the ADC, it will take a certain time for the voltage to stabilize. If not stabilized, the first value read after the first conversion may be wrong.

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 an 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 two ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic.

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 23-1 on page 255.

Figure 23-4. ADC Timing Diagram, First Conversion (Single Conversion Mode)

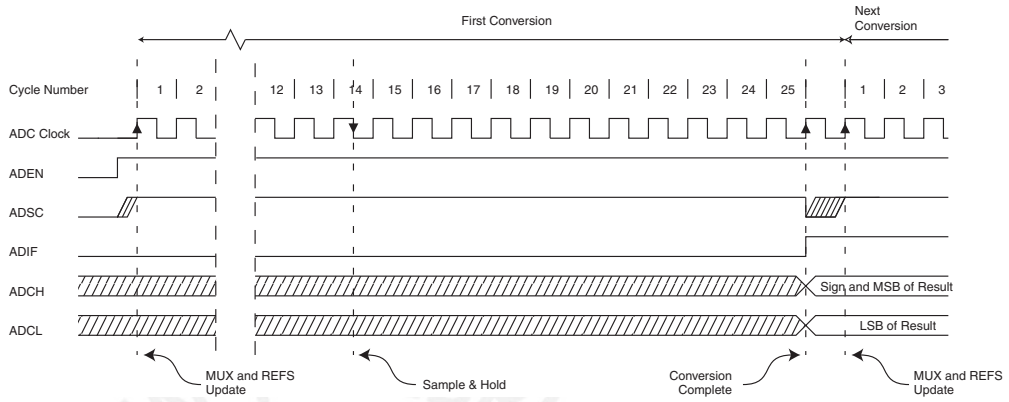


Figure 23-5. ADC Timing Diagram, Single Conversion

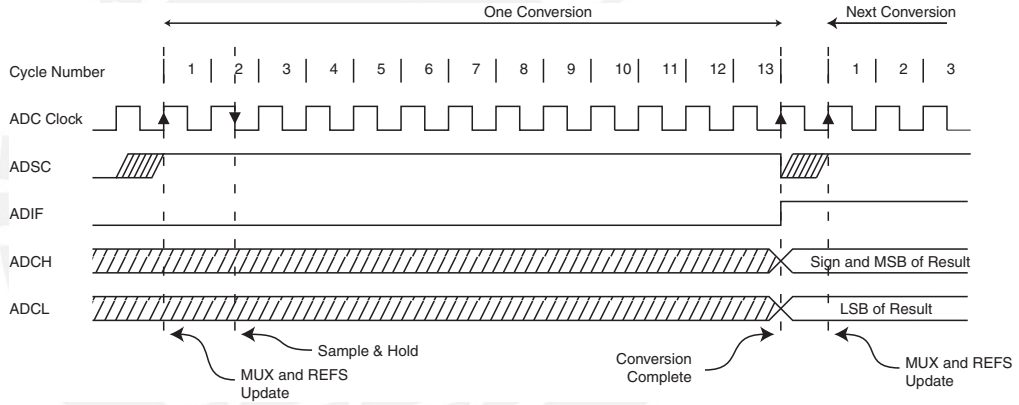


Figure 23-6. ADC Timing Diagram, Auto Triggered Conversion

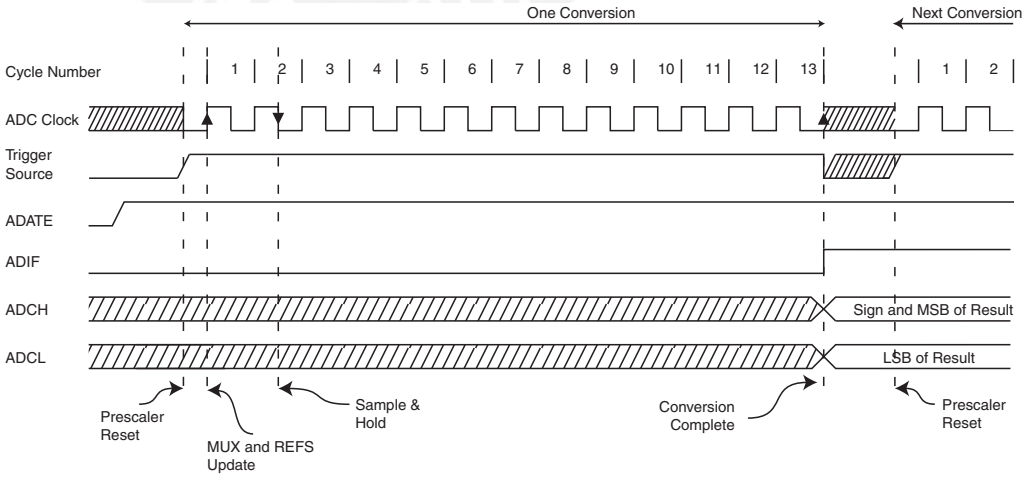


Figure 23-7. ADC Timing Diagram, Free Running Conversion

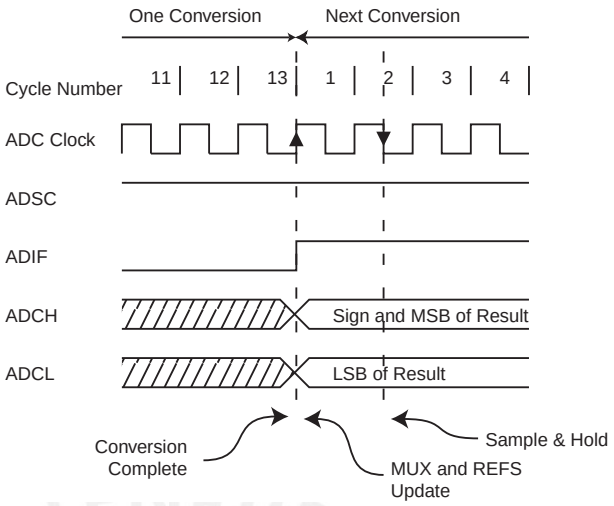


Table 23-1. ADC Conversion Time

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

23.5 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:

a. When ADATE or ADEN is cleared.
b. During conversion, minimum one ADC clock cycle after the trigger event.
c. 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.

23.5.1 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.

23.5.2 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 AV_{CC}, internal 1.1V reference, or external AREF pin.

AV_{CC} is connected to the ADC through a passive switch. The internal 1.1V 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 impedance voltmeter. Note that V_{REF} is a high impedance 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 AV_{CC} and 1.1V 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.

23.6 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:

a. 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.
b. Enter ADC Noise Reduction mode (or Idle mode). The ADC will start a conversion once the CPU has been halted.
c. 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.

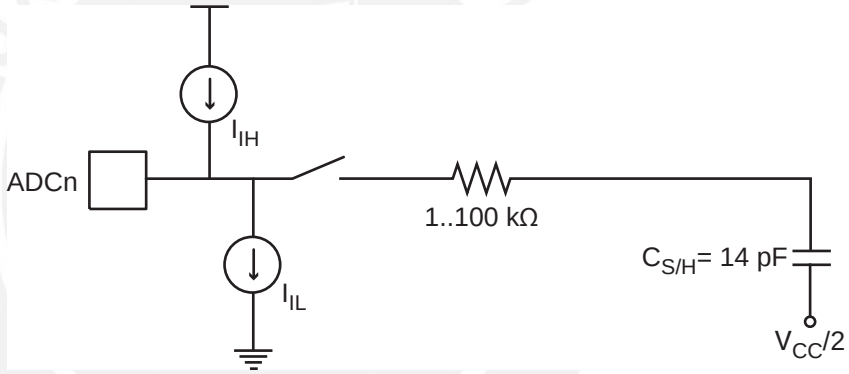
23.6.1 Analog Input Circuitry

The analog input circuitry for single ended channels is illustrated in Figure 23-8. 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 impedance sources with slowly varying signals, since this minimizes the required charge transfer to the S/H capacitor.

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 23-8. Analog Input Circuitry



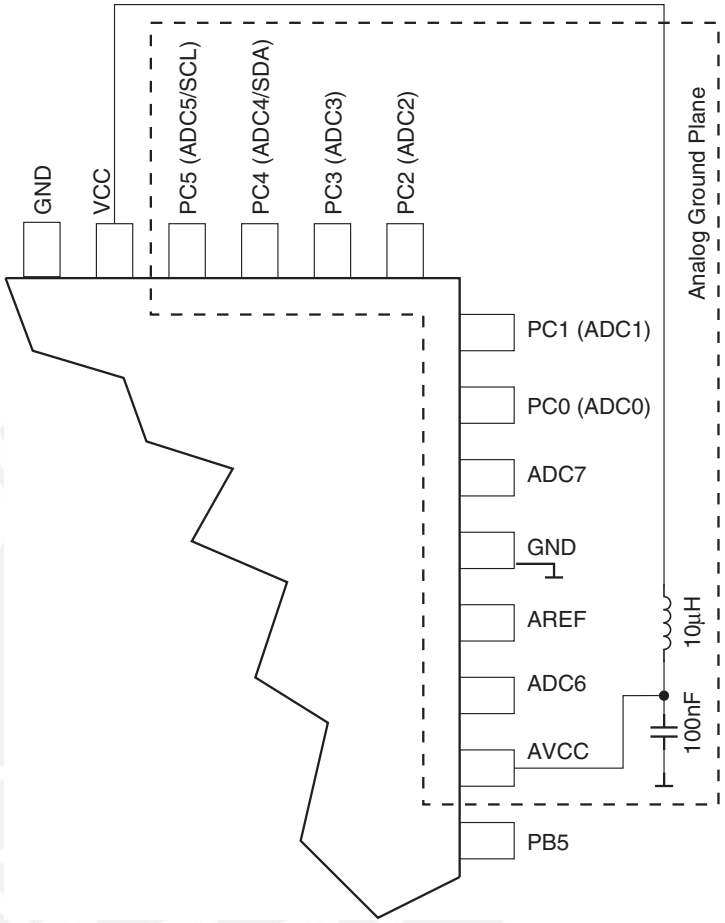
23.6.2 Analog Noise Canceling Techniques

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:

a. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
b. The AV_{CC} pin on the device should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 23-9.
c. Use the ADC noise canceler function to reduce induced noise from the CPU.
d. If any ADC [3..0] port pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress. However, using the 2-wire Interface (ADC4

and ADC5) will only affect the conversion on ADC4 and ADC5 and not the other ADC channels.

Figure 23-9. ADC Power Connections



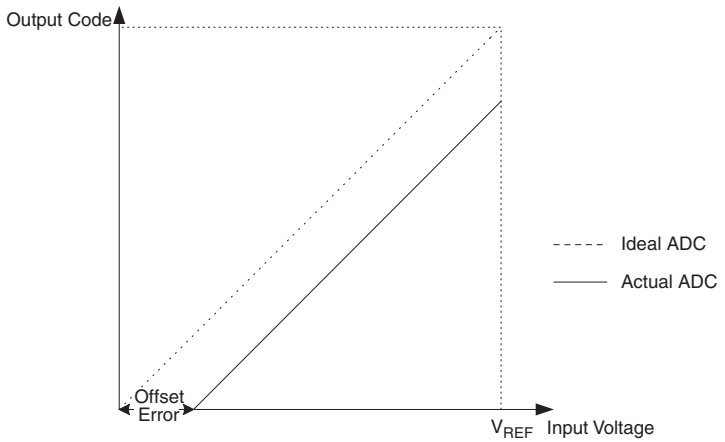
23.6.3 ADC Accuracy Definitions

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:

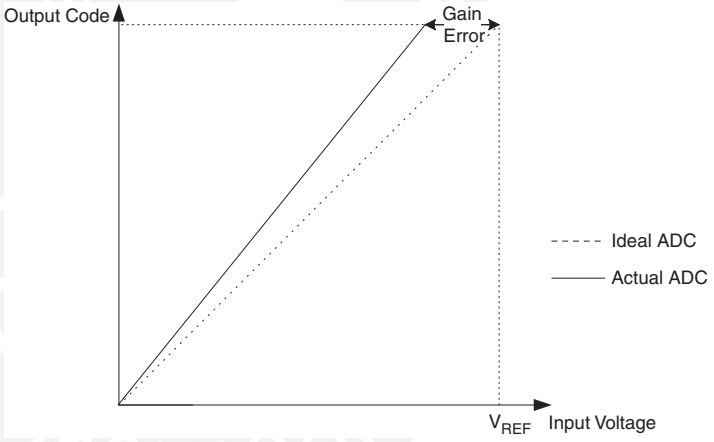
- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

Figure 23-10. Offset Error



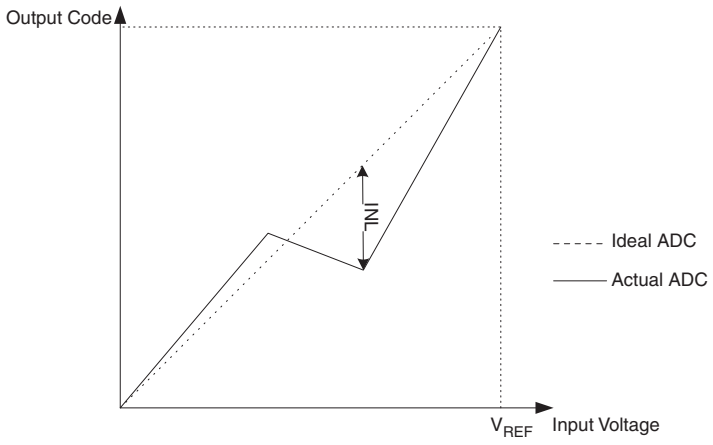
- Gain error: After adjusting for offset, the gain error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB

Figure 23-11. Gain Error



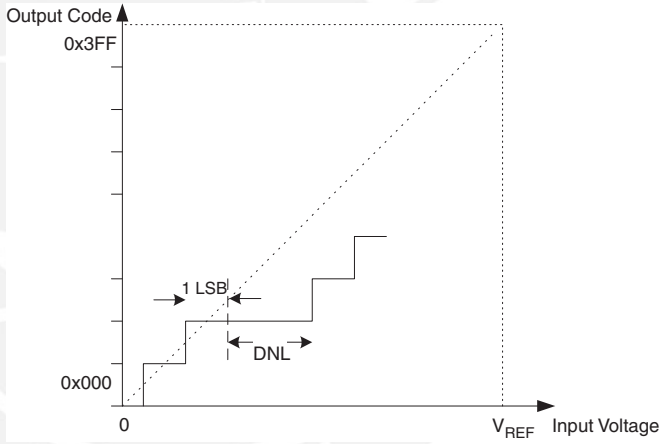
- Integral Non-linearity (INL): After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0 LSB.

Figure 23-12. 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 23-13. 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.

23.7 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 23-3 on page 262 and Table 23-4 on page 263). 0x000 represents analog ground, and 0x3FF represents the selected reference voltage minus one LSB.

23.8 Temperature Measurement

The temperature measurement is based on an on-chip temperature sensor that is coupled to a single ended ADC8 channel. Selecting the ADC8 channel by writing the MUX3..0 bits in ADMUX register to "1000" enables the temperature sensor. The internal 1.1V voltage reference must also be selected for the ADC voltage reference source in the temperature sensor measurement. When the temperature sensor is enabled, the ADC converter can be used in single conversion mode to measure the voltage over the temperature sensor.

The measured voltage has a linear relationship to the temperature as described in Table 23-2. The voltage sensitivity is approximately 1 mV/°C and the accuracy of the temperature measurement is +/- 10°C.

Table 23-2. Temperature vs. Sensor Output Voltage (Typical Case)

Temperature / °C	**-45°C**	**+25°C**	**+85°C**
Voltage / mV	242 mV	314 mV	380 mV

The values described in Table 23-2 are typical values. However, due to the process variation the temperature sensor output voltage varies from one chip to another. To be capable of achieving more accurate results the temperature measurement can be calibrated in the application software. The software calibration requires that a calibration value is measured and stored in a register or EEPROM for each chip, as a part of the production test. The software calibration can be done utilizing the formula:

$$T = \{ [(ADCH << 8) \mid ADCL] - T_{OS} \} / k$$

where ADCn are the ADC data registers, k is a fixed coefficient and T_{OS} is the temperature sensor offset value determined and stored into EEPROM as a part of the production test.

23.9 Register Description

23.9.1 ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
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 7:6 – REFS1:0: Reference Selection Bits**

These bits select the voltage reference for the ADC, as shown in Table 23-3. 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 23-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V 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 conversions. For a complete description of this bit, see ”ADCL and ADCH – The ADC Data Register” on page 265.

- **Bit 4 – Res: Reserved Bit**

This bit is an unused bit in the ATmega48PA/88PA/168PA/328P, and will always read as zero.

- **Bits 3:0 – MUX3:0: Analog Channel Selection Bits**

The value of these bits selects which analog inputs are connected to the ADC. See Table 23-4 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 23-4. Input Channel Selections

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8[1]
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V_{BG})
1111	0V (GND)

Note: 1. For Temperature Sensor.

23.9.2 ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	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 ADCSRB.

- **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 system clock frequency and the input clock to the ADC.

Table 23-5. 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

23.9.3 ADCL and ADCH – The ADC Data Register

23.9.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	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	

23.9.3.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	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.

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 261.

23.9.4 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 5:3 – Res: Reserved Bits**

These bits are reserved for future use. To ensure compatibility with future devices, these bist must be written to zero when ADCSRB is written.

- **Bit 2:0 – 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 23-6. 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 A
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

23.9.5 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – Res: Reserved Bits**

These bits are reserved for future use. To ensure compatibility with future devices, these bits must be written to zero when DIDR0 is written.

- **Bit 5:0 – ADC5D..ADC0D: ADC5..0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC5..0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

Note that ADC pins ADC7 and ADC6 do not have digital input buffers, and therefore do not require Digital Input Disable bits.

24. debugWIRE On-chip Debug System

24.1 Features

- **Complete Program Flow Control**
- **Emulates All On-chip Functions, Both Digital and Analog, except RESET Pin**
- **Real-time Operation**
- **Symbolic Debugging Support (Both at C and Assembler Source Level, or for Other HLLs)**
- **Unlimited Number of Program Break Points (Using Software Break Points)**
- **Non-intrusive Operation**
- **Electrical Characteristics Identical to Real Device**
- **Automatic Configuration System**
- **High-Speed Operation**
- **Programming of Non-volatile Memories**

24.2 Overview

The debugWIRE On-chip debug system uses a One-wire, bi-directional interface to control the program flow, execute AVR instructions in the CPU and to program the different non-volatile memories.

24.3 Physical Interface

When the debugWIRE Enable (DWEN) Fuse is programmed and Lock bits are unprogrammed, the debugWIRE system within the target device is activated. The RESET port pin is configured as a wire-AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.

Figure 24-1. The debugWIRE Setup

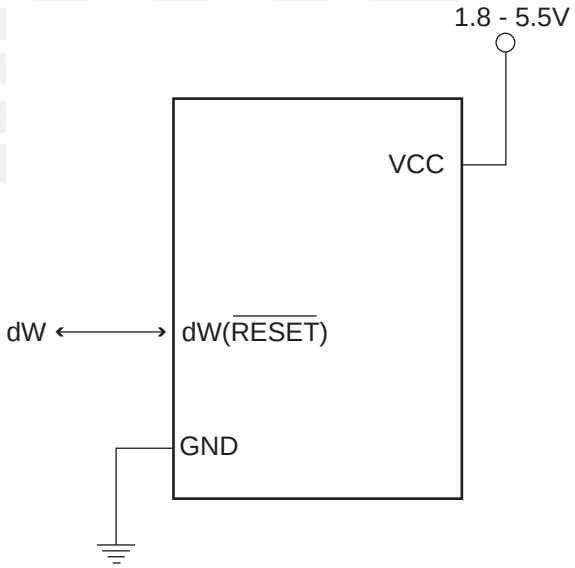


Figure 24-1 shows the schematic of a target MCU, with debugWIRE enabled, and the emulator connector. The system clock is not affected by debugWIRE and will always be the clock source selected by the CKSEL Fuses.

When designing a system where debugWIRE will be used, the following observations must be made for correct operation:

- Pull-up resistors on the dW/(RESET) line must not be smaller than 10kΩ. The pull-up resistor is not required for debugWIRE functionality.
- Connecting the RESET pin directly to V_{CC} will not work.
- Capacitors connected to the RESET pin must be disconnected when using debugWire.
- All external reset sources must be disconnected.

24.4 Software Break Points

debugWIRE supports Program memory Break Points by the AVR Break instruction. Setting a Break Point in AVR Studio® will insert a BREAK instruction in the Program memory. The instruction replaced by the BREAK instruction will be stored. When program execution is continued, the stored instruction will be executed before continuing from the Program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The Flash must be re-programmed each time a Break Point is changed. This is automatically handled by AVR Studio through the debugWIRE interface. The use of Break Points will therefore reduce the Flash Data retention. Devices used for debugging purposes should not be shipped to end customers.

24.5 Limitations of debugWIRE

The debugWIRE communication pin (dW) is physically located on the same pin as External Reset (RESET). An External Reset source is therefore not supported when the debugWIRE is enabled.

A programmed DWEN Fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the DWEN Fuse should be disabled when debugWire is not used.

24.6 Register Description

The following section describes the registers used with the debugWire.

24.6.1 DWDR – debugWire Data Register

Bit	7	6	5	4	3	2	1	0	
	DWDR[7:0]								DWDR
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 DWDR Register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in the normal operations.

25. Self-Programming the Flash, ATmega48PA

25.1 Overview

In ATmega48PA, there is no Read-While-Write support, and no separate Boot Loader Section. The SPM instruction can be executed from the entire Flash.

The device provides a Self-Programming mechanism for downloading and uploading program code by the MCU itself. The Self-Programming can use any available data interface and associated protocol to read code and write (program) that code into the Program memory.

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 re-written. 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.

25.1.1 Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write “00000011” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. 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 will be ignored during this operation.

- The CPU is halted during the Page Erase operation.

25.1.2 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 SPMCSR and execute SPM within four clock cycles after writing SPMCSR. 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 SPMCSR. 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.

If the EEPROM is written in the middle of an SPM Page Load operation, all data loaded will be lost.

25.1.3 Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write “00000101” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. 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 to zero during this operation.

- The CPU is halted during the Page Write operation.

25.2 Addressing the Flash During Self-Programming

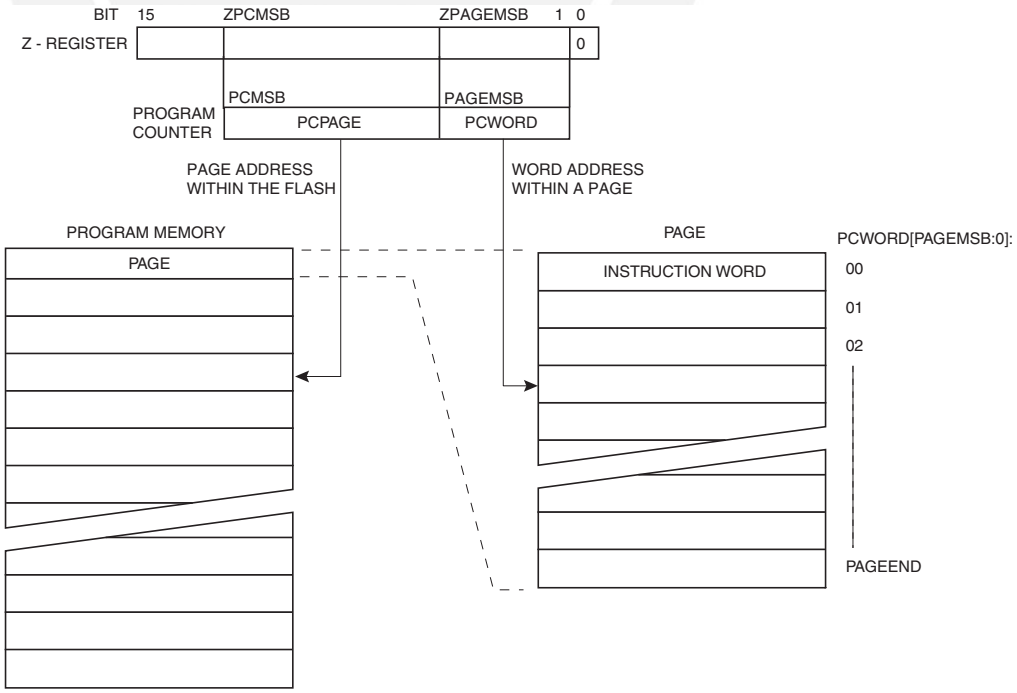
The Z-pointer is used to address the SPM commands.

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 27-11 on page 299), 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 26-3 on page 282. Note that the Page Erase and Page Write operations are addressed independently. Therefore it is of major importance that the software addresses the same page in both the Page Erase and Page Write operation.

The LPM instruction uses 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 25-1. Addressing the Flash During SPM(1)



Note: 1. The different variables used in Figure 26-3 are listed in Table 27-11 on page 299.

25.2.1 EEPROM Write Prevents Writing to SPMCSR

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 (EEPE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCSR Register.

25.2.2 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 0x0001 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the BLBSET and SELFPRGEN bits are set in SPMCSR, the value of the Lock bits will be loaded in the destination register. The BLBSET and SELFPRGEN 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 SELFPRGEN are cleared, LPM will work as described in the Instruction set Manual.

Bit	7	6	5	4	3	2	1	0
Rd	–	–	–	–	–	–	**LB2**	**LB1**

The algorithm for reading the Fuse Low byte is similar to the one described above for reading the Lock bits. To read the Fuse Low byte, load the Z-pointer with 0x0000 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse Low byte (FLB) will be loaded in the destination register as shown below.See Table 27-5 on page 296 for a detailed description and mapping of the Fuse Low byte.

Bit	7	6	5	4	3	2	1	0
Rd	**FLB7**	**FLB6**	**FLB5**	**FLB4**	**FLB3**	**FLB2**	**FLB1**	**FLB0**

Similarly, when reading the Fuse High byte (FHB), load 0x0003 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse High byte will be loaded in the destination register as shown below. See Table 27-5 on page 296 for detailed description and mapping of the Extended Fuse byte.

Bit	7	6	5	4	3	2	1	0
Rd	**FHB7**	**FHB6**	**FHB5**	**FHB4**	**FHB3**	**FHB2**	**FHB1**	**FHB0**

Similarly, when reading the Extended Fuse byte (EFB), load 0x0002 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Extended Fuse byte will be loaded in the destination register as shown below. See Table 27-5 on page 296 for detailed description and mapping of the Extended Fuse byte.

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.

25.2.3 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. 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.
2. 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 SPMCSR Register and thus the Flash from unintentional writes.

25.2.4 Programming Time for Flash when Using SPM

The calibrated RC Oscillator is used to time Flash accesses. Table 26-6 shows the typical programming time for Flash accesses from the CPU.

Table 25-1. SPM Programming Time(1)

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

Note: 1. Minimum and maximum programming time is per individual operation.

25.2.5 Simple Assembly Code Example for a Boot Loader

Note that the RWWSB bit will always be read as zero in ATmega48PA. Nevertheless, it is recommended to check this bit as shown in the code example, to ensure compatibility with devices supporting Read-While-Write.

```
  ;-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<<SELFPRGEN)
  rcallDo_spm

  ; re-enable the RWW section
  ldi  spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
  rcallDo_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<<SELFPRGEN)
  rcallDo_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<<SELFPRGEN)
  rcallDo_spm

  ; re-enable the RWW section
  ldi  spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
  rcallDo_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
  rjmp 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, SPMCSR
  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<<SELFPRGEN)
  rcallDo_spm
  rjmp Return

Do_spm:
  ; check for previous SPM complete
Wait_spm:
  in   temp1, SPMCSR
  sbrc temp1, SELFPRGEN
  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, EEPE
  rjmp Wait_ee
  ; SPM timed sequence
  out  SPMCSR, spmcrval
  spm
  ; restore SREG (to enable interrupts if originally enabled)
  out  SREG, temp2
  ret
```

25.3 Register Description

25.3.1 SPMCSR – Store Program Memory Control and Status Register

The Store Program Memory Control and Status Register contains the control bits needed to control the Program memory operations.

Bit	7	6	5	4	3	2	1	0	
0x37 (0x57)	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SELFPRGEN	SPMCSR
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 SELFPRGEN bit in the SPMCSR Register is cleared. The interrupt will not be generated during EEPROM write or SPM.

- **Bit 6 – RWWSB: Read-While-Write Section Busy**

This bit is for compatibility with devices supporting Read-While-Write. It will always read as zero in ATmega48PA.

- **Bit 5 – Res: Reserved Bit**

This bit is a reserved bit in the ATmega48PA/88PA/168PA/328P and will always read as zero.

- **Bit 4 – RWWSRE: Read-While-Write Section Read Enable**

The functionality of this bit in ATmega48PA is a subset of the functionality in ATmega88PA/168PA. If the RWWSRE bit is written while filling the temporary page buffer, the temporary page buffer will be cleared and the data will be lost.

- **Bit 3 – BLBSET: Boot Lock Bit Set**

The functionality of this bit in ATmega48PA is a subset of the functionality in ATmega88PA/168PA. An LPM instruction within three cycles after BLBSET and SELFPRGEN are set in the SPMCSR 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 271 for details.

- **Bit 2 – PGWRT: Page Write**

If this bit is written to one at the same time as SELFPRGEN, 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.

- **Bit 1 – PGERS: Page Erase**

If this bit is written to one at the same time as SELFPRGEN, 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.

- **Bit 0 – SELFPRGEN: Self Programming 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 SELFPRGEN 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 SELFPRGEN 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 SELFPRGEN 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.

26. Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P

26.1 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 27-11 on page 299) used during programming. The page organization does not affect normal operation.

26.2 Overview

In ATmega88PA, ATmega168PA and ATmega328P, 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.

26.3 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 26-2). The size of the different sections is configured by the BOOTSZ Fuses as shown in Table 26-7 on page 289 and Figure 26-2. These two sections can have different level of protection since they have different sets of Lock bits.

26.3.1 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 26-2 on page 281. The Application section can never store any Boot Loader code since the SPM instruction is disabled when executed from the Application section.

26.3.2 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 26-3 on page 281.

26.4 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 26-8 on page 289 and Figure 26-2 on page 280. 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.

26.4.1 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 ongoing 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 (i.e., 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 and Status Register (SPMCSR) 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 Section “26.9.1” on page 292. for details on how to clear RWWSB.

26.4.2 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 26-1. Read-While-Write Features

Which Section does the Z-pointer Address during the Programming?	Which Section can be read during Programming?	CPU Halted?	Read-While-Write Supported?
RWW Section	NRWW Section	No	Yes
NRWW Section	None	Yes	No

Figure 26-1. Read-While-Write vs. No Read-While-Write

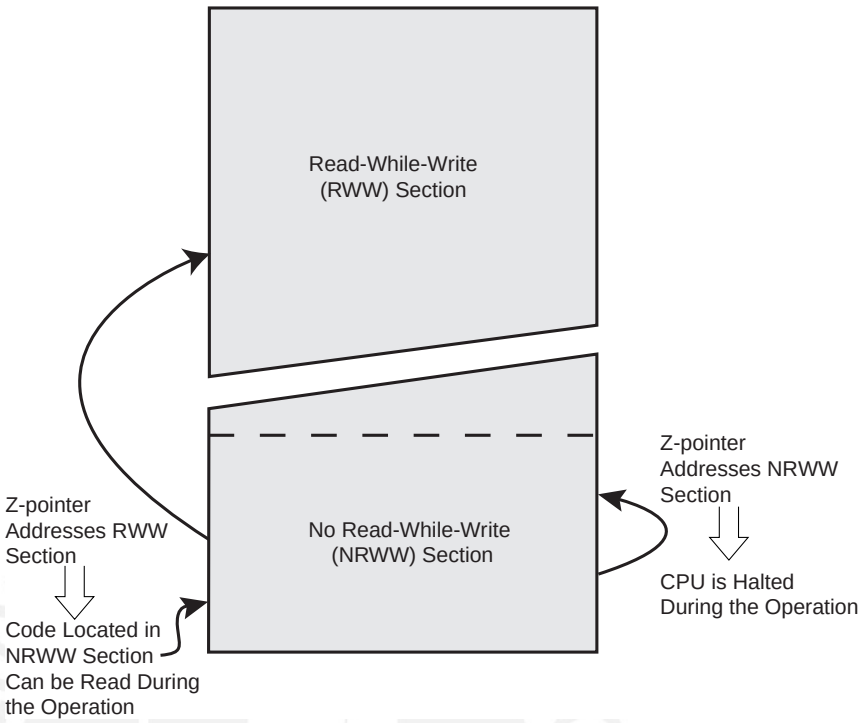
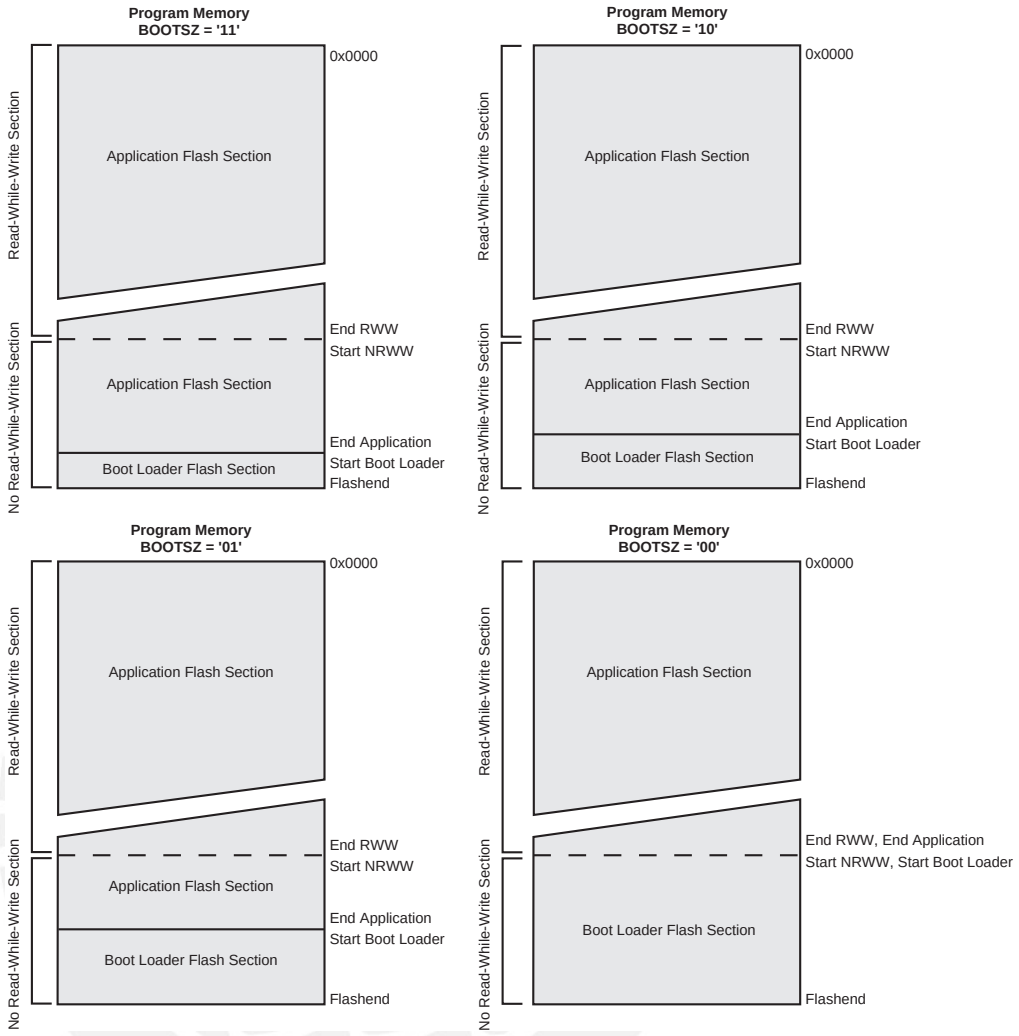


Figure 26-2. Memory Sections



Note: 1. The parameters in the figure above are given in Table 26-7 on page 289.

26.5 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 26-2 and Table 26-3 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 1) does not control reading nor writing by LPM/SPM, if it is attempted.

Table 26-2. 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 26-3. 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

26.6 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 26-4. Boot Reset Fuse[1]

BOOTRST	Reset Address
1	Reset Vector = Application Reset (address 0x0000)
0	Reset Vector = Boot Loader Reset (see Table 26-7 on page 289)

Note: 1. “1” means unprogrammed, “0” means programmed

26.7 Addressing the Flash During Self-Programming

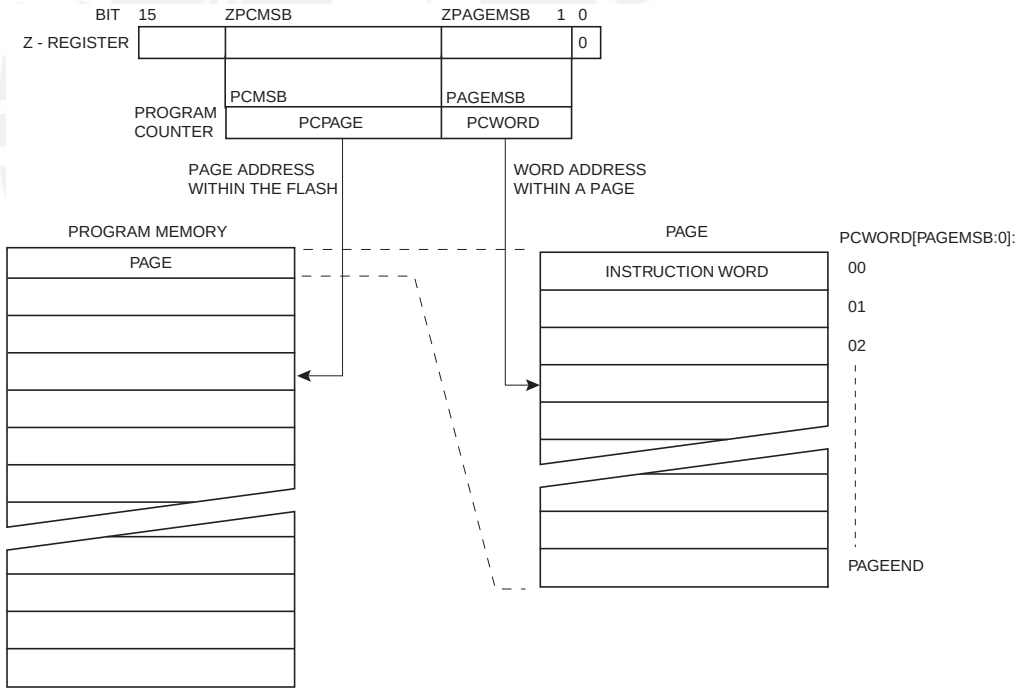
The Z-pointer is used to address the SPM commands.

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 27-11 on page 299), 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 is1 shown in Figure 26-3. 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 26-3. Addressing the Flash During SPM(1)



Note: 1. The different variables used in Figure 26-3 are listed in Table 26-9 on page 289.

26.8 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 286 for an assembly code example.

26.8.1 Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write “X0000011” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. 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 will be ignored 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.

26.8.2 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 SPMCSR and execute SPM within four clock cycles after writing SPMCSR. 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 SPMCSR. 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.

If the EEPROM is written in the middle of an SPM Page Load operation, all data loaded will be lost.

26.8.3 Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write “X0000101” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. 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 to 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.

26.8.4 Using the SPM Interrupt

If the SPM interrupt is enabled, the SPM interrupt will generate a constant interrupt when the SELFPRGEN bit in SPMCSR is cleared. This means that the interrupt can be used instead of polling the SPMCSR 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 57.

26.8.5 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.

26.8.6 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 SPMCSR 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 ”Watchdog Timer” on page 50, 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 286 for an example.

26.8.7 Setting the Boot Loader Lock Bits by SPM

To set the Boot Loader Lock bits and general Lock Bits, write the desired data to R0, write “X0001001” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR.

Bit	7	6	5	4	3	2	1	0
R0	**1**	**1**	**BLB12**	**BLB11**	**BLB02**	**BLB01**	**LB2**	**LB1**

See Table 26-2 and Table 26-3 for how the different settings of the Boot Loader bits affect the Flash access.

If bits 5..0 in R0 are cleared (zero), the corresponding Lock bit will be programmed if an SPM instruction is executed within four cycles after BLBSET and SELFPRGEN are set in SPMCSR. The Z-pointer is don’t care during this operation, but for future compatibility it is recommended to load the Z-pointer with 0x0001 (same as used for reading the lO_{ck} bits). For future compatibility it is also recommended to set bits 7 and 6 in R0 to “1” when writing the Lock bits. When programming the Lock bits the entire Flash can be read during the operation.

26.8.8 EEPROM Write Prevents Writing to SPMCSR

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 (EEPE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCSR Register.

26.8.9 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 0x0001 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM

instruction is executed within three CPU cycles after the BLBSET and SELFPRGEN bits are set in SPMCSR, the value of the Lock bits will be loaded in the destination register. The BLBSET and SELFPRGEN 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 SELFPRGEN 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 byte is similar to the one described above for reading the Lock bits. To read the Fuse Low byte, load the Z-pointer with 0x0000 and set the BLBSET and SELFPRGEN bits in SPMCSR. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse Low byte (FLB) will be loaded in the destination register as shown below. Refer to Table 27-5 on page 296 for a detailed description and mapping of the Fuse Low byte.

Bit	7	6	5	4	3	2	1	0
Rd	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

Similarly, when reading the Fuse High byte, load 0x0003 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Fuse High byte (FHB) will be loaded in the destination register as shown below. Refer to Table 27-7 on page 296 for detailed description and mapping of the Fuse High byte.

Bit	7	6	5	4	3	2	1	0
Rd	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

When reading the Extended Fuse byte, load 0x0002 in the Z-pointer. When an LPM instruction is executed within three cycles after the BLBSET and SELFPRGEN bits are set in the SPMCSR, the value of the Extended Fuse byte (EFB) will be loaded in the destination register as shown below. Refer to Table 27-5 on page 296 for detailed description and mapping of the Extended Fuse byte.

Bit	7	6	5	4	3	2	1	0
Rd	–	–	–	–	EFB3	EFB2	EFB1	EFB0

Fuse and Lock bits that are programmed, will be read as zero. Fuse and Lock bits that are unprogrammed, will be read as one.

26.8.10 Reading the Signature Row from Software

To read the Signature Row from software, load the Z-pointer with the signature byte address given in Table 26-5 on page 286 and set the SIGRD and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the SIGRD and SPMEN bits are set in SPMCSR, the signature byte value will be loaded in the destination register. The SIGRD and SPMEN bits will auto-clear upon completion of reading the Signature Row Lock bits or if no LPM instruction is executed within three CPU cycles. When SIGRD and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Table 26-5. Signature Row Addressing

Signature Byte	Z-Pointer Address
Device Signature Byte 1	0x0000
Device Signature Byte 2	0x0002
Device Signature Byte 3	0x0004
RC Oscillator Calibration Byte	0x0001

Note: All other addresses are reserved for future use.

26.8.11 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 SPMCSR Register and thus the Flash from unintentional writes.

26.8.12 Programming Time for Flash when Using SPM

The calibrated RC Oscillator is used to time Flash accesses. Table 26-6 shows the typical programming time for Flash accesses from the CPU.

Table 26-6. SPM Programming Time(1)

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

Note: 1. Minimum and maximum programming time is per individual operation.

26.8.13 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<<SELFPRGEN)
  call Do_spm

  ; re-enable the RWW section
  ldi  spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
  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<<SELFPRGEN)
  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<<SELFPRGEN)
  call Do_spm

  ; re-enable the RWW section
  ldi  spmcrval, (1<<RWWSRE) | (1<<SELFPRGEN)
  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, SPMCSR
```

```
  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<<SELFPRGEN)
  call Do_spm
  rjmp Return

Do_spm:
  ; check for previous SPM complete
Wait_spm:
  in   temp1, SPMCSR
  sbrc temp1, SELFPRGEN
  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, EEPE
  rjmp Wait_ee
  ; SPM timed sequence
  out  SPMCSR, spmcrval
  spm
  ; restore SREG (to enable interrupts if originally enabled)
  out  SREG, temp2
  ret
```

26.8.14 ATmega88PA Boot Loader Parameters

In Table 26-7 through Table 26-9, the parameters used in the description of the self programming are given.

Table 26-7. Boot Size Configuration, ATmega88PA

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	4	0x000 - 0xF7F	0xF80 - 0xFFF	0xF7F	0xF80
1	0	256 words	8	0x000 - 0xEFF	0xF00 - 0xFFF	0xEFF	0xF00
0	1	512 words	16	0x000 - 0xDFF	0xE00 - 0xFFF	0xDFF	0xE00
0	0	1024 words	32	0x000 - 0xBFF	0xC00 - 0xFFF	0xBFF	0xC00

Note: The different BOOTSZ Fuse configurations are shown in Figure 26-2 on page 280.

Table 26-8. Read-While-Write Limit, ATmega88PA

Section	Pages	Address
Read-While-Write section (RWW)	96	0x000 - 0xBFF
No Read-While-Write section (NRWW)	32	0xC00 - 0xFFF

For details about these two section, see "NRWW – No Read-While-Write Section" on page 278 and "RWW – Read-While-Write Section" on page 278

Table 26-9. Explanation of Different Variables used in Figure 26-3 and the Mapping to the Z-pointer, ATmega88PA

Variable		Corresponding Z-value(1)	Description
PCMSB	11		Most significant bit in the Program Counter. (The Program Counter is 12 bits PC[11:0])
PAGEMSB	4		Most significant bit which is used to address the words within one page (32 words in a page requires 5 bits PC [4:0]).
ZPCMSB		Z12	Bit in Z-register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z5	Bit in Z-register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.
PCPAGE	PC[11:5]	Z12:Z6	Program counter page address: Page select, for page erase and page write
PCWORD	PC[4:0]	Z5:Z1	Program counter word address: Word select, for filling temporary buffer (must be zero during page write operation)

Note: 1. Z15:Z13: 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 282 for details about the use of Z-pointer during Self-Programming.

26.8.15 ATmega168PA Boot Loader Parameters

In Table 26-10 through Table 26-12, the parameters used in the description of the self programming are given.

Table 26-10. Boot Size Configuration, ATmega168PA

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	0x0000 - 0x1F7F	0x1F80 - 0x1FFF	0x1F7F	0x1F80
1	0	256 words	4	0x0000 - 0x1EFF	0x1F00 - 0x1FFF	0x1EFF	0x1F00
0	1	512 words	8	0x0000 - 0x1DFF	0x1E00 - 0x1FFF	0x1DFF	0x1E00
0	0	1024 words	16	0x0000 - 0x1BFF	0x1C00 - 0x1FFF	0x1BFF	0x1C00

Note: The different BOOTSZ Fuse configurations are shown in Figure 26-2 on page 280.

Table 26-11. Read-While-Write Limit, ATmega168PA

Section	Pages	Address
Read-While-Write section (RWW)	112	0x0000 - 0x1BFF
No Read-While-Write section (NRWW)	16	0x1C00 - 0x1FFF

For details about these two section, see "NRWW – No Read-While-Write Section" on page 278 and "RWW – Read-While-Write Section" on page 278

Table 26-12. Explanation of Different Variables used in Figure 26-3 and the Mapping to the Z-pointer, ATmega168PA

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 282 for details about the use of Z-pointer during Self-Programming.

26.8.16 ATmega328P Boot Loader Parameters

In Table 26-13 through Table 26-15, the parameters used in the description of the self programming are given.

Table 26-13. Boot Size Configuration, ATmega328P

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	256 words	4	0x0000 - 0x3EFF	0x3F00 - 0x3FFF	0x3EFF	0x3F00
1	0	512 words	8	0x0000 - 0x3DFF	0x3E00 - 0x3FFF	0x3DFF	0x3E00
0	1	1024 words	16	0x0000 - 0x3BFF	0x3C00 - 0x3FFF	0x3BFF	0x3C00
0	0	2048 words	32	0x0000 - 0x37FF	0x3800 - 0x3FFF	0x37FF	0x3800

Note: The different BOOTSZ Fuse configurations are shown in Figure 26-2 on page 280.

Table 26-14. Read-While-Write Limit, ATmega328P

Section	Pages	Address
Read-While-Write section (RWW)	224	0x0000 - 0x37FF
No Read-While-Write section (NRWW)	32	0x3800 - 0x3FFF

For details about these two section, see "NRWW – No Read-While-Write Section" on page 278 and "RWW – Read-While-Write Section" on page 278.

Table 26-15. Explanation of Different Variables used in Figure 26-3 and the Mapping to the Z-pointer, ATmega328P

Variable		Corresponding Z-value[1]	Description
PCMSB	13		Most significant bit in the Program Counter. (The Program Counter is 14 bits PC[13:0])
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		Z14	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[13:6]	Z14: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: 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 282 for details about the use of Z-pointer during Self-Programming.

26.9 Register Description

26.9.1 SPMCSR – Store Program Memory Control and Status Register

The Store Program Memory Control and Status Register contains the control bits needed to control the Boot Loader operations.

Bit	7	6	5	4	3	2	1	0	
0x37 (0x57)	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SELFPRGEN	SPMCSR
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 SELFPRGEN bit in the SPMCSR 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 ATmega48PA/88PA/168PA/328P 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 (SELFPRGEN will be cleared). Then, if the RWWSRE bit is written to one at the same time as SELFPRGEN, 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 (SELFPRGEN 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 SELFPRGEN, the next SPM instruction within four clock cycles sets Boot Lock bits and Memory 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 SELFPRGEN are set in the SPMCSR 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 284 for details.

- **Bit 2 – PGWRT: Page Write**

If this bit is written to one at the same time as SELFPRGEN, 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 SELFPRGEN, 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 – SELFPRGEN: Self Programming 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 SELFPRGEN 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 SELFPRGEN 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 SELFPRGEN 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.

27. Memory Programming

27.1 Program And Data Memory Lock Bits

The ATmega88PA/168PA/328P provides six Lock bits which can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 27-2. The Lock bits can only be erased to “1” with the Chip Erase command. The ATmega48PA has no separate Boot Loader section. The SPM instruction is enabled for the whole Flash if the SELFPRGEN fuse is programmed (“0”), otherwise it is disabled.

Table 27-1. Lock Bit Byte(1)

Lock Bit Byte	Bit No	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12(2)	5	Boot Lock bit	1 (unprogrammed)
BLB11(2)	4	Boot Lock bit	1 (unprogrammed)
BLB02(2)	3	Boot Lock bit	1 (unprogrammed)
BLB01(2)	2	Boot Lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Notes: 1. “1” means unprogrammed, “0” means programmed.
2. Only on ATmega88PA/168PA/328P.

Table 27-2. Lock Bit Protection Modes(1)(2)

Memory Lock Bits			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 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 Serial Programming mode. The Boot Lock bits and Fuse bits are locked in both Serial and Parallel Programming mode.(1)

Notes: 1. Program the Fuse bits and Boot Lock bits before programming the LB1 and LB2.
2. “1” means unprogrammed, “0” means programmed

Table 27-3. Lock Bit Protection Modes[1][2]. Only ATmega88PA/168PA/328P.

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.
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 and Boot Lock bits before programming the LB1 and LB2.
2. “1” means unprogrammed, “0” means programmed

27.2 Fuse Bits

The ATmega48PA/88PA/168PA/328P has three Fuse bytes. Table 27-5 - Table 27-9 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 27-4. Extended Fuse Byte for ATmega48PA

Extended Fuse Byte	Bit No	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
–	2	–	1
–	1	–	1
SELFPRGEN	0	Self Programming Enable	1 (unprogrammed)

Table 27-5. Extended Fuse Byte for ATmega88PA/168PA

Extended Fuse Byte	Bit No	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
BOOTSZ1	2	Select Boot Size (see Table 26-7 on page 289 and Table 26-10 on page 290 for details)	0 (programmed)(1)
BOOTSZ0	1	Select Boot Size (see Table 26-7 on page 289 and Table 26-10 on page 290 for details)	0 (programmed)(1)
BOOTRST	0	Select Reset Vector	1 (unprogrammed)

Note: 1. The default value of BOOTSZ[1:0] results in maximum Boot Size. See "Pin Name Mapping" on page 300.

Table 27-6. Extended Fuse Byte for ATmega328P

Extended Fuse Byte	Bit No	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
BODLEVEL2(1)	2	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL1(1)	1	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL0(1)	0	Brown-out Detector trigger level	1 (unprogrammed)

Note: 1. See Table 28-4 on page 318 for BODLEVEL Fuse decoding.

Table 27-7. Fuse High Byte for ATmega48PA/88PA/168PA

High Fuse Byte	Bit No	Description	Default Value
RSTDISBL(1)	7	External Reset Disable	1 (unprogrammed)
DWEN	6	debugWIRE Enable	1 (unprogrammed)
SPIEN(2)	5	Enable Serial Program and Data Downloading	0 (programmed, SPI programming enabled)

Table 27-7. Fuse High Byte for ATmega48PA/88PA/168PA (Continued)

High Fuse Byte	Bit No	Description	Default Value
WDTON(3)	4	Watchdog Timer Always On	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed), EEPROM not reserved
BODLEVEL2(4)	2	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL1(4)	1	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL0(4)	0	Brown-out Detector trigger level	1 (unprogrammed)

Notes: 1. See "Alternate Functions of Port C" on page 85 for description of RSTDISBL Fuse.
2. The SPIEN Fuse is not accessible in serial programming mode.
3. See "WDTCSR – Watchdog Timer Control Register" on page 54 for details.
4. See Table 28-4 on page 318 for BODLEVEL Fuse decoding.

Table 27-8. Fuse High Byte for ATmega328P

High Fuse Byte	Bit No	Description	Default Value
RSTDISBL(1)	7	External Reset Disable	1 (unprogrammed)
DWEN	6	debugWIRE Enable	1 (unprogrammed)
SPIEN(2)	5	Enable Serial Program and Data Downloading	0 (programmed, SPI programming enabled)
WDTON(3)	4	Watchdog Timer Always On	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed), EEPROM not reserved
BOOTSZ1	2	Select Boot Size (see Table 26-7 on page 289, Table 26-10 on page 290 and Table 26-13 on page 291 for details)	0 (programmed)(4)
BOOTSZ0	1	Select Boot Size (see Table 26-7 on page 289, Table 26-10 on page 290 and Table 26-13 on page 291 for details)	0 (programmed)(4)
BOOTRST	0	Select Reset Vector	1 (unprogrammed)

Notes: 1. See "Alternate Functions of Port C" on page 85 for description of RSTDISBL Fuse.
2. The SPIEN Fuse is not accessible in serial programming mode.
3. See "WDTCSR – Watchdog Timer Control Register" on page 54 for details.
4. The default value of BOOTSZ[1:0] results in maximum Boot Size. See "Pin Name Mapping" on page 300.

Table 27-9. Fuse Low Byte

Low Fuse Byte	Bit No	Description	Default Value
CKDIV8[4]	7	Divide clock by 8	0 (programmed)
CKOUT[3]	6	Clock output	1 (unprogrammed)
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	1 (unprogrammed)[2]
CKSEL0	0	Select Clock source	0 (programmed)[2]

Note:
1. The default value of SUT1..0 results in maximum start-up time for the default clock source. See Table 8-12 on page 33 for details.
2. The default setting of CKSEL3..0 results in internal RC Oscillator @ 8 MHz. See Table 8-11 on page 33 for details.
3. The CKOUT Fuse allows the system clock to be output on PORTB0. See "Clock Output Buffer" on page 35 for details.
4. See "System Clock Prescaler" on page 35 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.

27.2.1 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.

27.3 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 ATmega48PA/88PA/168PA/328P the signature bytes are given in Table 27-10.

Table 27-10. Device ID

	Signature Bytes Address		
Part	**0x000**	**0x001**	**0x002**
ATmega48PA	0x1E	0x92	0x0A
ATmega88PA	0x1E	0x93	0x0F
ATmega168PA	0x1E	0x94	0x0B
ATmega328P	0x1E	0x95	0x0F

27.4 Calibration Byte

The ATmega48PA/88PA/168PA/328P has a byte calibration value for the Internal RC Oscillator. This byte resides in the high byte of address 0x000 in the signature address space. During reset,

this byte is automatically written into the OSCCAL Register to ensure correct frequency of the calibrated RC Oscillator.

27.5 Page Size

Table 27-11. No. of Words in a Page and No. of Pages in the Flash

Device	Flash Size	Page Size	PCWORD	No. of Pages	PCPAGE	PCMSB
ATmega48PA	2K words (4K bytes)	32 words	PC[4:0]	64	PC[10:5]	10
ATmega88PA	4K words (8K bytes)	32 words	PC[4:0]	128	PC[11:5]	11
ATmega168PA	8K words (16K bytes)	64 words	PC[5:0]	128	PC[12:6]	12
ATmega328P	16K words (32K bytes)	64 words	PC[5:0]	256	PC[13:6]	13

Table 27-12. No. of Words in a Page and No. of Pages in the EEPROM

Device	EEPROM Size	Page Size	PCWORD	No. of Pages	PCPAGE	EEAMSB
ATmega48PA	256 bytes	4 bytes	EEA[1:0]	64	EEA[7:2]	7
ATmega88PA	512 bytes	4 bytes	EEA[1:0]	128	EEA[8:2]	8
ATmega168PA	512 bytes	4 bytes	EEA[1:0]	128	EEA[8:2]	8
ATmega328P	1K bytes	4 bytes	EEA[1:0]	256	EEA[9:2]	9

27.6 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 ATmega48PA/88PA/168PA/328P. Pulses are assumed to be at least 250 ns unless otherwise noted.

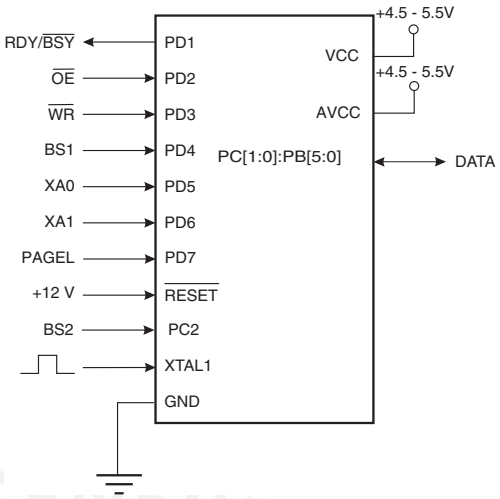
27.6.1 Signal Names

In this section, some pins of the ATmega48PA/88PA/168PA/328P are referenced by signal names describing their functionality during parallel programming, see Figure 27-1 and Table 27-13. 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 27-15.

When pulsing $\overline{WR}$ or $\overline{OE}$, the command loaded determines the action executed. The different Commands are shown in Table 27-16.

Figure 27-1. Parallel Programming



Note: V_{CC} - 0.3V < AV_{CC} < V_{CC} + 0.3V, however, AV_{CC} should always be within 4.5 - 5.5V

Table 27-13. 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	PC2	I	Byte Select 2 (“0” selects Low byte, “1” selects 2’nd High byte)
DATA	{PC[1:0]: PB[5:0]}	I/O	Bi-directional Data bus (Output when $\overline{OE}$ is low)

Table 27-14. 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 27-15. 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 27-16. 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
0000 1000	Read Signature Bytes and Calibration byte
0000 0100	Read Fuse and Lock bits
0000 0010	Read Flash
0000 0011	Read EEPROM

27.7 Parallel Programming

27.7.1 Enter Programming Mode

The following algorithm puts the device in Parallel (High-voltage) Programming mode:

1. Set Prog_enable pins listed in Table 27-14 on page 300 to “0000”, RESET pin to 0V and V_{CC} to 0V.
2. Apply 4.5 - 5.5V between V_{CC} and GND.

Ensure that V_{CC} reaches at least 1.8V within the next 20 µs.

3. Wait 20 - 60 µs, and apply 11.5 - 12.5V to RESET.
4. Keep the Prog_enable pins unchanged for at least 10µs after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
5. Wait at least 300 µs before giving any parallel programming commands.
6. Exit Programming mode by power the device down or by bringing RESET pin to 0V.

If the rise time of the V_{CC} is unable to fulfill the requirements listed above, the following alternative algorithm can be used.

1. Set Prog_enable pins listed in Table 27-14 on page 300 to “0000”, RESET pin to 0V and V_{CC} to 0V.
2. Apply 4.5 - 5.5V between V_{CC} and GND.
3. Monitor V_{CC}, and as soon as V_{CC} reaches 0.9 - 1.1V, apply 11.5 - 12.5V to RESET.

4. Keep the Prog_enable pins unchanged for at least 10µs after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
5. Wait until V_{CC} actually reaches 4.5 -5.5V before giving any parallel programming commands.
6. Exit Programming mode by power the device down or by bringing RESET pin to 0V.

27.7.2 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 0xFF, 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.

27.7.3 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 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 $\overline{WR}$ a negative pulse. This starts the Chip Erase. RDY/$\overline{BSY}$ goes low.
6. Wait until RDY/$\overline{BSY}$ goes high before loading a new command.

27.7.4 Programming the Flash

The Flash is organized in pages, see Table 27-11 on page 299. 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 (0x00 - 0xFF).

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 (0x00 - 0xFF).
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 (0x00 - 0xFF).
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 27-3 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 27-2 on page 304. Note that if less than eight 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 (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the address high byte.

H. Program Page

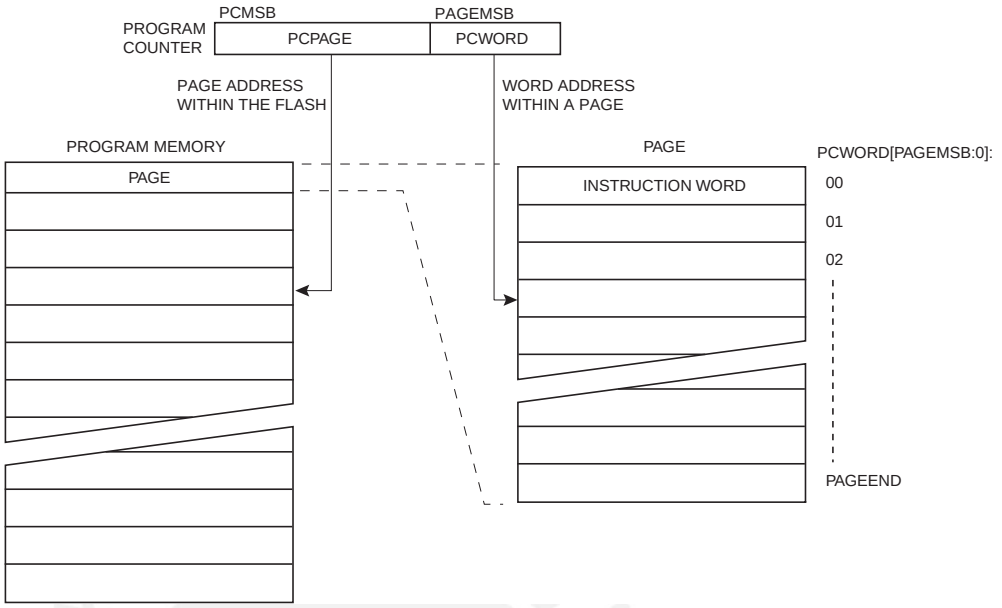
1. Give $\overline{WR}$ a negative pulse. This starts programming of the entire page of data. RDY/$\overline{BSY}$ goes low.
2. Wait until RDY/$\overline{BSY}$ goes high (See Figure 27-3 for signal waveforms).

I. Repeat B through H until the entire Flash is programmed or until all data has been programmed.

J. End Page Programming

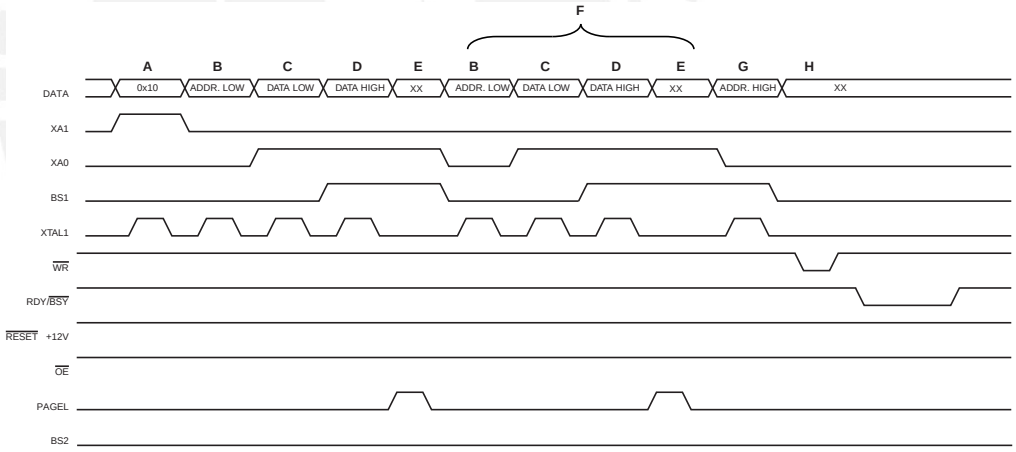
1. 1. Set XA1, XA0 to “10”. This enables command loading.
2. Set DATA to “0000 0000”. This is the command for No Operation.
3. Give XTAL1 a positive pulse. This loads the command, and the internal write signals are reset.

Figure 27-2. Addressing the Flash Which is Organized in Pages[1]



Note: 1. PCPAGE and PCWORD are listed in Table 27-11 on page 299.

Figure 27-3. Programming the Flash Waveforms[1]



Note: 1. “XX” is don’t care. The letters refer to the programming description above.

27.7.5 Programming the EEPROM

The EEPROM is organized in pages, see Table 27-12 on page 299. 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 302 for details on Command, Address and Data loading):

1. A: Load Command “0001 0001”.
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. C: Load Data (0x00 - 0xFF).

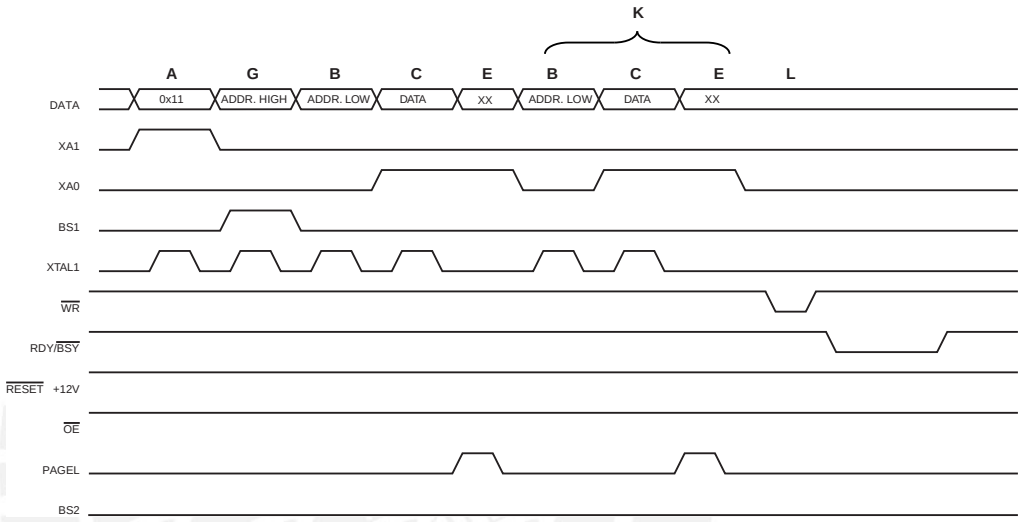
5. E: Latch data (give PAGEL a positive pulse).

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 27-4 for signal waveforms).

Figure 27-4. Programming the EEPROM Waveforms



27.7.6 Reading the Flash

The algorithm for reading the Flash memory is as follows (refer to ”Programming the Flash” on page 302 for details on Command and Address loading):

1. A: Load Command “0000 0010”.
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
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”.

27.7.7 Reading the EEPROM

The algorithm for reading the EEPROM memory is as follows (refer to ”Programming the Flash” on page 302 for details on Command and Address loading):

1. A: Load Command “0000 0011”.
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
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”.

27.7.8 Programming the Fuse Low Bits

The algorithm for programming the Fuse Low bits is as follows (refer to ”Programming the Flash” on page 302 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. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.

27.7.9 Programming the Fuse High Bits

The algorithm for programming the Fuse High bits is as follows (refer to ”Programming the Flash” on page 302 for details on Command and Data loading):

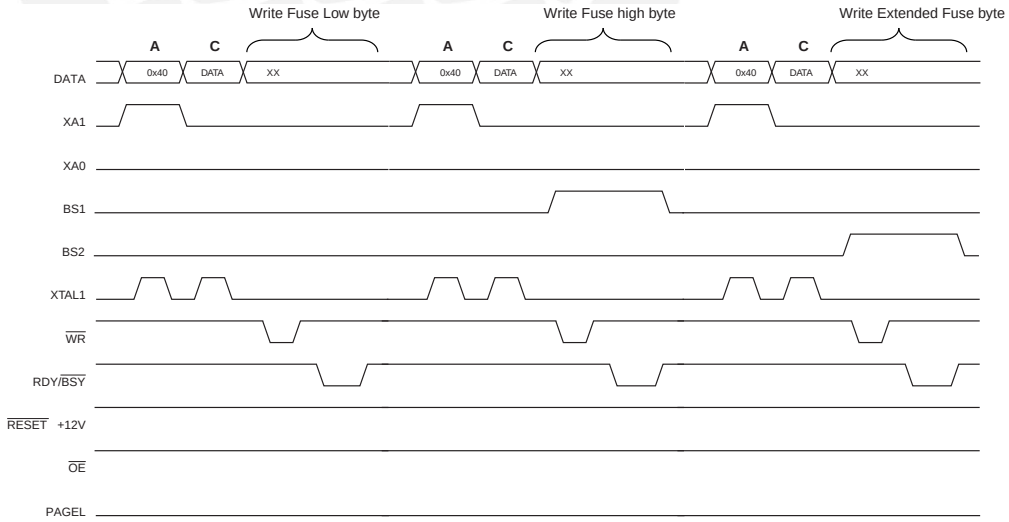
1. A: Load Command “0100 0000”.
2. C: Load Data Low Byte. Bit n = “0” programs and bit n = “1” erases the Fuse bit.
3. Set BS1 to “1” and BS2 to “0”. This selects high data byte.
4. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.
5. Set BS1 to “0”. This selects low data byte.

27.7.10 Programming the Extended Fuse Bits

The algorithm for programming the Extended Fuse bits is as follows (refer to ”Programming the Flash” on page 302 for details on Command and Data loading):

1. 1. A: Load Command “0100 0000”.
2. 2. C: Load Data Low Byte. Bit n = “0” programs and bit n = “1” erases the Fuse bit.
3. 3. Set BS1 to “0” and BS2 to “1”. This selects extended data byte.
4. 4. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.
5. 5. Set BS2 to “0”. This selects low data byte.

Figure 27-5. Programming the FUSES Waveforms



27.7.11 Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (refer to "Programming the Flash" on page 302 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. If LB mode 3 is programmed (LB1 and LB2 is programmed), it is not possible to program the Boot Lock bits by any External Programming mode.
3. Give $\overline{WR}$ a negative pulse and wait for RDY/$\overline{BSY}$ to go high.

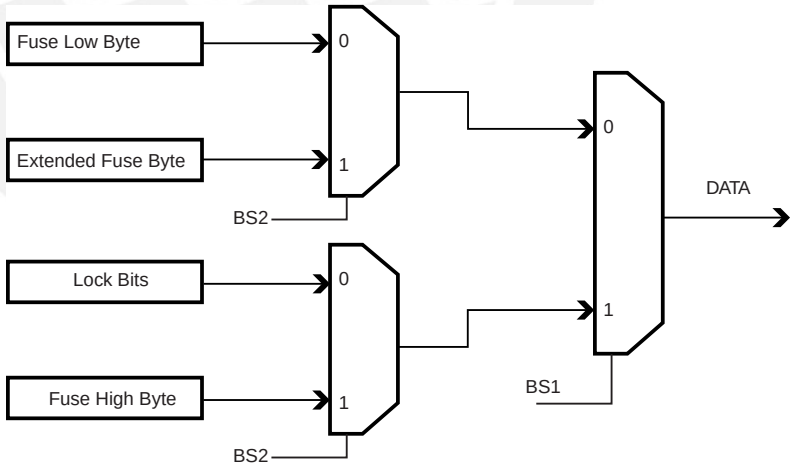
The Lock bits can only be cleared by executing Chip Erase.

27.7.12 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 302 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 OE to "0", BS2 to "1", and BS1 to "0". The status of the Extended Fuse bits can now be read at DATA ("0" means programmed).
5. 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).
6. Set $\overline{OE}$ to "1".

Figure 27-6. Mapping Between BS1, BS2 and the Fuse and Lock Bits During Read



27.7.13 Reading the Signature Bytes

The algorithm for reading the Signature bytes is as follows (refer to "Programming the Flash" on page 302 for details on Command and Address loading):

1. A: Load Command "0000 1000".
2. B: Load Address Low Byte (0x00 - 0x02).
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".

27.7.14 Reading the Calibration Byte

The algorithm for reading the Calibration byte is as follows (refer to ”Programming the Flash” on page 302 for details on Command and Address loading):

1. A: Load Command “0000 1000”.
2. B: Load Address Low Byte, 0x00.
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”.

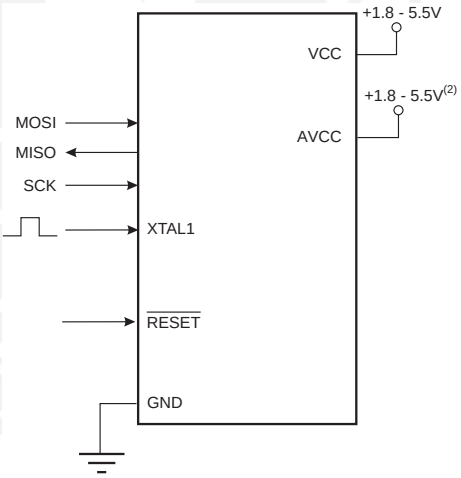
27.7.15 Parallel Programming Characteristics

For chracteristics of the Parallel Programming, see ”Parallel Programming Characteristics” on page 324.

27.8 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 27-17 on page 309, the pin mapping for SPI programming is listed. Not all parts use the SPI pins dedicated for the internal SPI interface.

Figure 27-7. 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 < AV_{CC} < V_{CC} + 0.3V, however, AV_{CC} should always be within 1.8 - 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 0xFF.

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

27.8.1 Serial Programming Pin Mapping

Table 27-17. Pin Mapping Serial Programming

Symbol	Pins	I/O	Description
MOSI	PB3	I	Serial Data in
MISO	PB4	O	Serial Data out
SCK	PB5	I	Serial Clock

27.8.2 Serial Programming Algorithm

When writing serial data to the ATmega48PA/88PA/168PA/328P, data is clocked on the rising edge of SCK.

When reading data from the ATmega48PA/88PA/168PA/328P, data is clocked on the falling edge of SCK. See Figure 27-9 for timing details.

To program and verify the ATmega48PA/88PA/168PA/328P in the serial programming mode, the following sequence is recommended (See Serial Programming Instruction set in Table 27-19 on page 310):

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 serial programming by sending the Programming Enable serial instruction to pin MOSI.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync. the second byte (0x53), 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 0x53 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 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 (RDY/$\overline{\text{BSY}}$) is not used, the user must wait at least t_{WD_FLASH} before issuing the next page (See Table 27-18). Accessing the serial programming interface before the Flash write operation completes can result in incorrect programming.
5. **A**: 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 (RDY/$\overline{\text{BSY}}$) is not used, the user must wait at least t_{WD_EEPROM} before issuing the next byte (See Table 27-18). In a chip erased device, no 0xFFs in the data file(s) need to be programmed.
   B: The EEPROM array is programmed one page at a time. The Memory page is loaded one byte at a time by supplying the 6 LSB of the address and data together with the Load EEPROM Memory Page instruction. The EEPROM Memory Page is stored by loading the Write EEPROM Memory Page Instruction with the 7 MSB of the address. When using EEPROM page access only byte locations loaded with the Load EEPROM Memory Page instruction is altered. The remaining locations remain unchanged. If polling (RDY/$\overline{\text{BSY}}$) is

not used, the used must wait at least t_{WD_EEPROM} before issuing the next byte (See Table 27-18). In a chip erased device, no 0xFF 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.

Table 27-18. Typical Wait Delay Before Writing the Next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FLASH}	4.5 ms
t_{WD_EEPROM}	3.6 ms
t_{WD_ERASE}	9.0 ms

27.8.3 Serial Programming Instruction set

Table 27-19 on page 310 and Figure 27-8 on page 312 describes the Instruction set.

Table 27-19. Serial Programming Instruction Set (Hexadecimal values)

	Instruction Format			
Instruction/Operation	**Byte 1**	**Byte 2**	**Byte 3**	**Byte4**
Programming Enable	$AC	$53	$00	$00
Chip Erase (Program Memory/EEPROM)	$AC	$80	$00	$00
Poll RDY/$\overline{\text{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	$00	adr LSB	high data byte in
Load Program Memory Page, Low byte	$40	$00	adr LSB	low data byte in
Load EEPROM Memory Page (page access)	$C1	$00	0000 000aa	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	0000 00aa	aaaa aaaa	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	$00	data byte out

Table 27-19. Serial Programming Instruction Set (Hexadecimal values) (Continued)

Instruction/Operation	Instruction Format			
	Byte 1	**Byte 2**	**Byte 3**	**Byte4**
Write Instructions[6]				
Write Program Memory Page	$4C	adr MSB	adr LSB	$00
Write EEPROM Memory	$C0	0000 00aa	aaaa aaaa	data byte in
Write EEPROM Memory Page (page access)	$C2	0000 00aa	aaaa aa00	$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

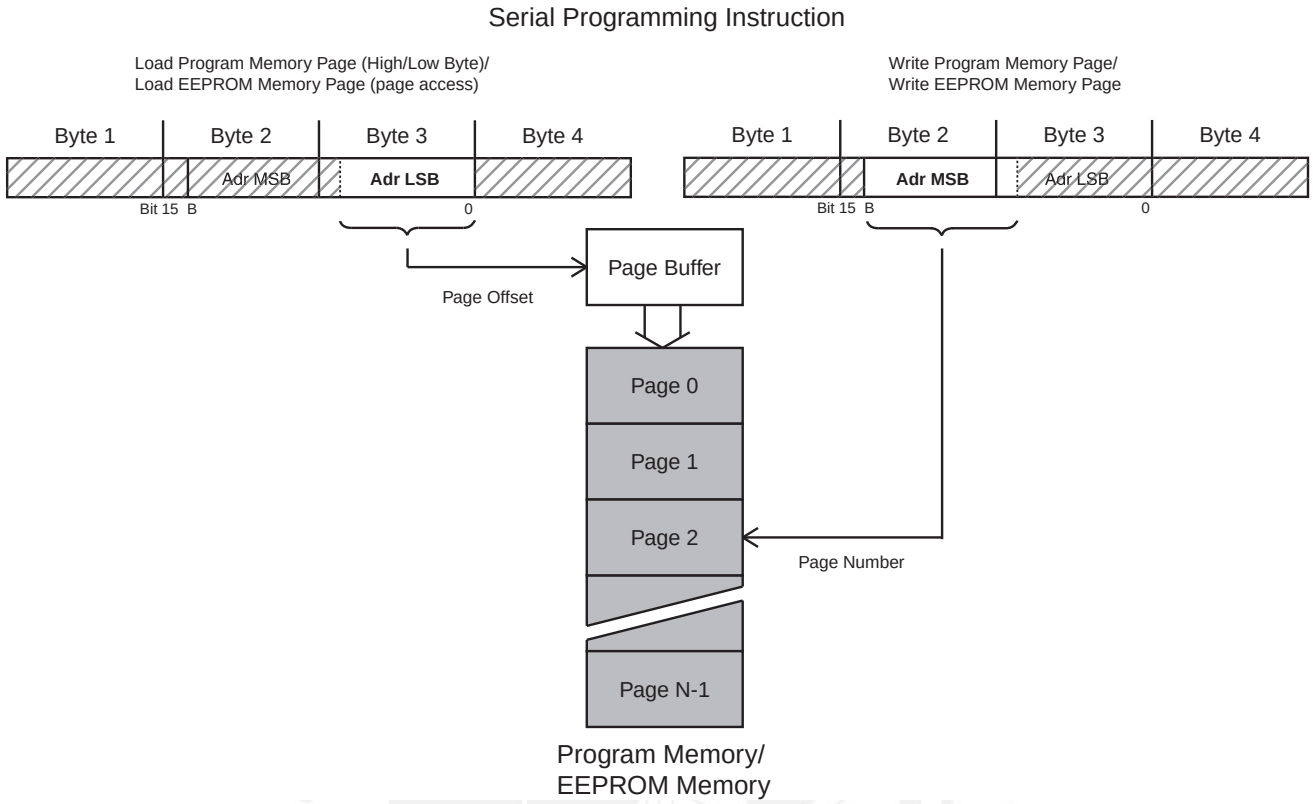
Notes:
1. Not all instructions are applicable for all parts.
2. a = address.
3. Bits are programmed ‘0’, unprogrammed ‘1’.
4. To ensure future compatibility, unused Fuses and Lock bits should be unprogrammed (‘1’) .
5. Refer to the correspondig section for Fuse and Lock bits, Calibration and Signature bytes and Page size.
6. Instructions accessing program memory use a word address. This address may be random within the page range.
7. 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.

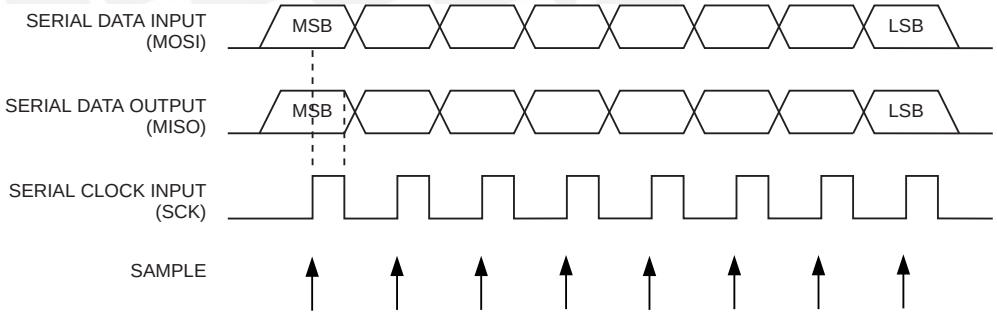
After data is loaded to the page buffer, program the EEPROM page, see Figure 27-8 on page 312.

Figure 27-8. Serial Programming Instruction example



27.8.4 SPI Serial Programming Characteristics

Figure 27-9. Serial Programming Waveforms



For characteristics of the SPI module see “SPI Timing Characteristics” on page 319.

28. Electrical Characteristics

28.1 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

*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.

28.2 DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage, except XTAL1 and $\overline{\text{RESET}}$ pin	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	-0.5 -0.5		$0.2V_{CC}$[1] $0.3V_{CC}$[1]	V
V_{IH}	Input High Voltage, except XTAL1 and $\overline{\text{RESET}}$ pins	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	$0.7V_{CC}$[2] $0.6V_{CC}$[2]		V_{CC} + 0.5 V_{CC} + 0.5	V
V_{IL1}	Input Low Voltage, XTAL1 pin	V_{CC} = 1.8V - 5.5V	-0.5		$0.1V_{CC}$[1]	V
V_{IH1}	Input High Voltage, XTAL1 pin	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	$0.8V_{CC}$[2] $0.7V_{CC}$[2]		V_{CC} + 0.5 V_{CC} + 0.5	V
V_{IL2}	Input Low Voltage, $\overline{\text{RESET}}$ pin	V_{CC} = 1.8V - 5.5V	-0.5		$0.1V_{CC}$[1]	V
V_{IH2}	Input High Voltage, $\overline{\text{RESET}}$ pin	V_{CC} = 1.8V - 5.5V	$0.9V_{CC}$[2]		V_{CC} + 0.5	V
V_{IL3}	Input Low Voltage, $\overline{\text{RESET}}$ pin as I/O	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	-0.5 -0.5		$0.2V_{CC}$[1] $0.3V_{CC}$[1]	V
V_{IH3}	Input High Voltage, $\overline{\text{RESET}}$ pin as I/O	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	$0.7V_{CC}$[2] $0.6V_{CC}$[2]		V_{CC} + 0.5 V_{CC} + 0.5	V
V_{OL}	Output Low Voltage[3] except RESET pin	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.9 0.6	V
V_{OH}	Output High Voltage[4] except Reset pin	I_{OH} = -20 mA, V_{CC} = 5V I_{OH} = -10 mA, V_{CC} = 3V	4.2 2.3			V
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin low (absolute value)			1	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin high (absolute value)			1	µA

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted) (Continued)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
R_{RST}	Reset Pull-up Resistor		30		60	kΩ
R_{PU}	I/O Pin Pull-up Resistor		20		50	kΩ
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V V_{in} = V_{CC}/2		<10	40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = V_{CC}/2	-50		50	nA
t_{ACID}	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 V_{CC} = 5V, 10 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   ATmega48PA/88PA/168PA/328P:
   1] The sum of all I_{OL}, for ports C0 - C5, ADC7, ADC6 should not exceed 100 mA.
   2] The sum of all I_{OL}, for ports B0 - B5, D5 - D7, XTAL1, XTAL2 should not exceed 100 mA.
   3] The sum of all I_{OL}, for ports D0 - D4, $\overline{\text{RESET}}$ should not exceed 100 mA.
   If I_{OL} exceeds the test condition, V_{OL} 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 V_{CC} = 5V, 10 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   ATmega48PA/88PA/168PA/328P:
   1] The sum of all I_{OH}, for ports C0 - C5, D0- D4, ADC7, $\overline{\text{RESET}}$ should not exceed 150 mA.
   2] The sum of all I_{OH}, for ports B0 - B5, D5 - D7, ADC6, XTAL1, XTAL2 should not exceed 150 mA.
   If II_{OH} exceeds the test condition, V_{OH} may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

28.2.1 ATmega48PA DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.[2]	Max.	Units
I_{CC}	Power Supply Current[1]	Active 1 MHz, V_{CC} = 2V		0.2	0.5	mA
		Active 4 MHz, V_{CC} = 3V		1.2	2.5	mA
		Active 8 MHz, V_{CC} = 5V		4.0	9	mA
		Idle 1 MHz, V_{CC} = 2V		0.03	0.15	mA
		Idle 4 MHz, V_{CC} = 3V		0.21	0.7	mA
		Idle 8 MHz, V_{CC} = 5V		0.9	2.7	mA
	Power-save mode[3]	32 kHz TOSC enabled, V_{CC} = 1.8V		0.75		µA
		32 kHz TOSC enabled, V_{CC} = 3V		0.9		µA
	Power-down mode[3]	WDT enabled, V_{CC} = 3V		3.9	8	µA
		WDT disabled, V_{CC} = 3V		0.1	2	µA

Notes:
1. Values with "Minimizing Power Consumption" enabled (0xFF).
2. Typical values at 25°C. Maximum values are characterized values and not test limits in production.
3. The current consumption values include input leakage current.

28.2.2 ATmega88PA DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.[2]	Max.	Units
I_{CC}	Power Supply Current[1]	Active 1 MHz, V_{CC} = 2V		0.2	0.5	mA
		Active 4 MHz, V_{CC} = 3V		1.2	2.5	mA
		Active 8 MHz, V_{CC} = 5V		4.1	9	mA
		Idle 1 MHz, V_{CC} = 2V		0.03	0.15	mA
		Idle 4 MHz, V_{CC} = 3V		0.18	0.7	mA
		Idle 8 MHz, V_{CC} = 5V		0.8	2.7	mA
	Power-save mode[3]	32 kHz TOSC enabled, V_{CC} = 1.8V		0.8		µA
		32 kHz TOSC enabled, V_{CC} = 3V		0.9		µA
	Power-down mode[3]	WDT enabled, V_{CC} = 3V		3.9	8	µA
		WDT disabled, V_{CC} = 3V		0.1	2	µA

Notes: 1. Values with “Minimizing Power Consumption” enabled (0xFF).
2. Typical values at 25°C. Maximum values are test limits in production.
3. The current consumption values include input leakage current.

28.2.3 ATmega168PA DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.[2]	Max.	Units
I_{CC}	Power Supply Current[1]	Active 1 MHz, V_{CC} = 2V		0.2	0.5	mA
		Active 4 MHz, V_{CC} = 3V		1.2	2.5	mA
		Active 8 MHz, V_{CC} = 5V		4.2	9	mA
		Idle 1 MHz, V_{CC} = 2V		0.03	0.15	mA
		Idle 4 MHz, V_{CC} = 3V		0.2	0.7	mA
		Idle 8 MHz, V_{CC} = 5V		0.9	2.7	
	Power-save mode[3]	32 kHz TOSC enabled, V_{CC} = 1.8V		0.75		µA
		32 kHz TOSC enabled, V_{CC} = 3V		0.83		µA
	Power-down mode[3]	WDT enabled, V_{CC} = 3V		4.1	8	µA
		WDT disabled, V_{CC} = 3V		0.1	2	µA

Notes: 1. Values with “Minimizing Power Consumption” enabled (0xFF).
2. Typical values at 25°C. Maximum values are test limits in production.
3. The current consumption values include input leakage current.

28.2.4 ATmega328P DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

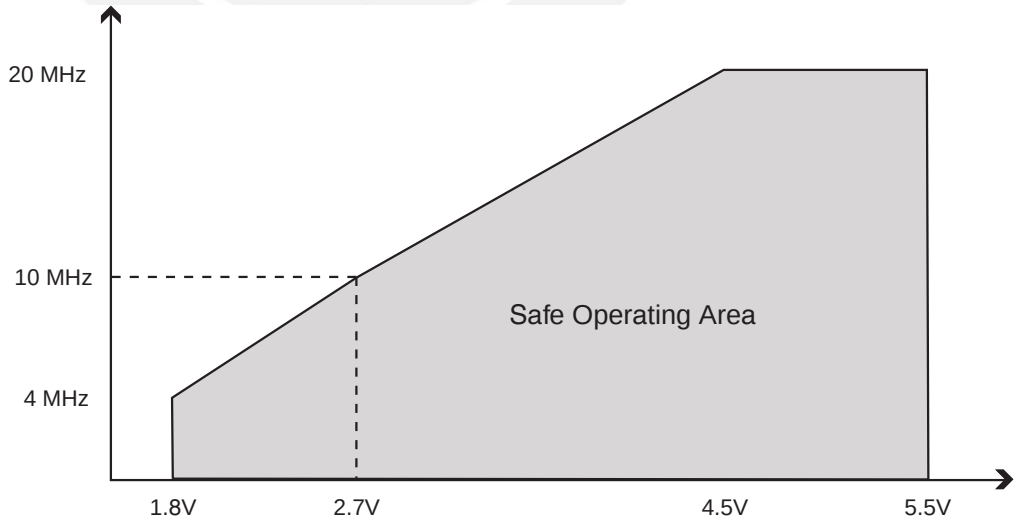
Symbol	Parameter	Condition	Min.	Typ.[2]	Max.	Units
I_{CC}	Power Supply Current[1]	Active 1 MHz, V_{CC} = 2V		0.3	0.5	mA
		Active 4 MHz, V_{CC} = 3V		1.7	2.5	mA
		Active 8 MHz, V_{CC} = 5V		5.2	9	mA
		Idle 1 MHz, V_{CC} = 2V		0.04	0.15	mA
		Idle 4 MHz, V_{CC} = 3V		0.3	0.7	mA
		Idle 8 MHz, V_{CC} = 5V		1.2	2.7	mA
	Power-save mode[3][4]	32 kHz TOSC enabled, V_{CC} = 1.8V		0.8	1.6	µA
		32 kHz TOSC enabled, V_{CC} = 3V		0.9	2.6	µA
	Power-down mode[3]	WDT enabled, V_{CC} = 3V		4.2	8	µA
		WDT disabled, V_{CC} = 3V		0.1	2	µA

Notes:
1. Values with “Minimizing Power Consumption” enabled (0xFF).
2. Typical values at 25°C. Maximum values are test limits in production.
3. The current consumption values include input leakage current.
4. Maximum values are characterized values and not test limits in production.

28.3 Speed Grades

Maximum frequency is dependent on V_{CC}. As shown in Figure 28-1, the Maximum Frequency vs. V_{CC} curve is linear between 1.8V < V_{CC} < 2.7V and between 2.7V < V_{CC} < 4.5V.

Figure 28-1. Maximum Frequency vs. V_{CC}



28.4 Clock Characteristics

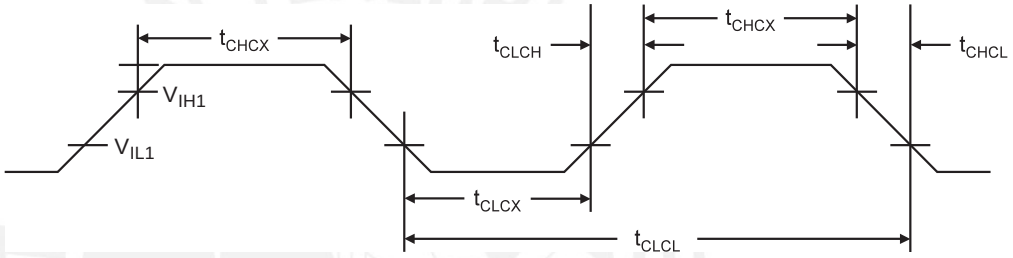
28.4.1 Calibrated Internal RC Oscillator Accuracy

Table 28-1. Calibration Accuracy of Internal RC Oscillator

	Frequency	V_{CC}	Temperature	Calibration Accuracy
Factory Calibration	8.0 MHz	3V	25°C	±10%
User Calibration	7.3 - 8.1 MHz	1.8V - 5.5V	-40°C - 85°C	±1%

28.4.2 External Clock Drive Waveforms

Figure 28-2. External Clock Drive Waveforms



28.4.3 External Clock Drive

Table 28-2. External Clock Drive

Symbol	Parameter	V_{CC}= 1.8 - 5.5V		V_{CC}= 2.7 - 5.5V		V_{CC}= 4.5 - 5.5V		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
$1/t_{CLCL}$	Oscillator Frequency	0	4	0	10	0	20	MHz
t_{CLCL}	Clock Period	250		100		50		ns
t_{CHCX}	High Time	100		40		20		ns
t_{CLCX}	Low Time	100		40		20		ns
t_{CLCH}	Rise Time		2.0		1.6		0.5	µs
t_{CHCL}	Fall Time		2.0		1.6		0.5	µs
Δt_{CLCL}	Change in period from one clock cycle to the next		2		2		2	%

Note: All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.

28.5 System and Reset Characteristics

Table 28-3. Reset, Brown-out and Internal Voltage Characteristics[1]

Symbol	Parameter		Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)		1.1	1.4	1.6	V
	Power-on Reset Threshold Voltage (falling)[2]		0.6	1.3	1.6	V
SR_{ON}	Power-on Slope Rate		0.01		10	V/ms
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.2 V_{CC}		0.9 V_{CC}	V
t_{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin				2.5	µs
V_{HYST}	Brown-out Detector Hysteresis			50		mV
t_{BOD}	Min Pulse Width on Brown-out Reset			2		µs
V_{BG}	Bandgap reference voltage	V_{CC}=2.7 T_A=25°C	1.0	1.1	1.2	V
t_{BG}	Bandgap reference start-up time	V_{CC}=2.7 T_A=25°C		40	70	µs
I_{BG}	Bandgap reference current consumption	V_{CC}=2.7 T_A=25°C		10		µA

Notes: 1. Values are guidelines only.
2. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling)

Table 28-4. BODLEVEL Fuse Coding[1]

BODLEVEL 2:0 Fuses	Min V_{BOT}	Typ V_{BOT}	Max V_{BOT}	Units
111	BOD Disabled			
110	1.7	1.8	2.0	V
101	2.5	2.7	2.9	
100	4.1	4.3	4.5	
011	Reserved			
010				
001				
000				

Notes: 1. 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 = 110, 101 and 100.

28.6 SPI Timing Characteristics

See Figure 28-3 and Figure 28-4 for details.

Table 28-5. SPI Timing Parameters

	Description	Mode	Min	Typ	Max	
1	SCK period	Master		See Table 18-5		ns
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}		
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_{ck}			
11	SCK high/low(1)	Slave	2 • t_{ck}			
12	Rise/Fall time	Slave			1600	
13	Setup	Slave	10			
14	Hold	Slave	t_{ck}			
15	SCK to out	Slave		15		
16	SCK to $\overline{SS}$ high	Slave	20			
17	$\overline{SS}$ high to tri-state	Slave		10		
18	$\overline{SS}$ low to SCK	Slave	20			

Note:
1. In SPI Programming mode the minimum SCK high/low period is:
   - 2 t_{CLCL} for f_{CK} < 12 MHz
   - 3 t_{CLCL} for f_{CK} > 12 MHz
2. All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.

Figure 28-3. SPI Interface Timing Requirements (Master Mode)

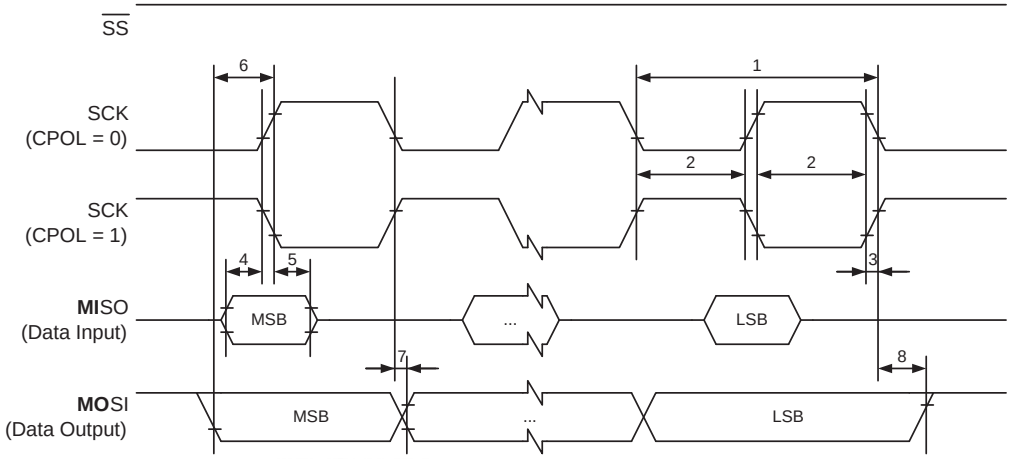
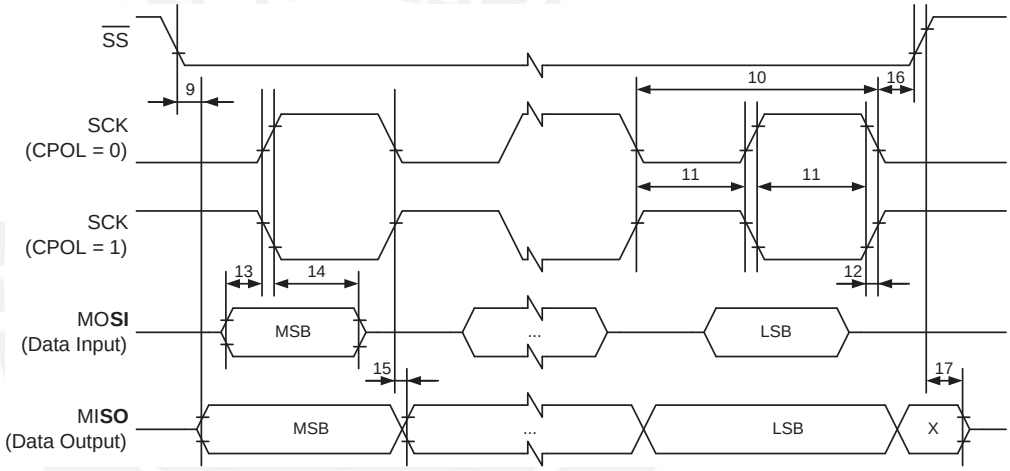


Figure 28-4. SPI Interface Timing Requirements (Slave Mode)



28.7 2-wire Serial Interface Characteristics

Table 28-6 describes the requirements for devices connected to the 2-wire Serial Bus. The ATmega48PA/88PA/168PA/328P 2-wire Serial Interface meets or exceeds these requirements under the noted conditions.

Timing symbols refer to Figure 28-5.

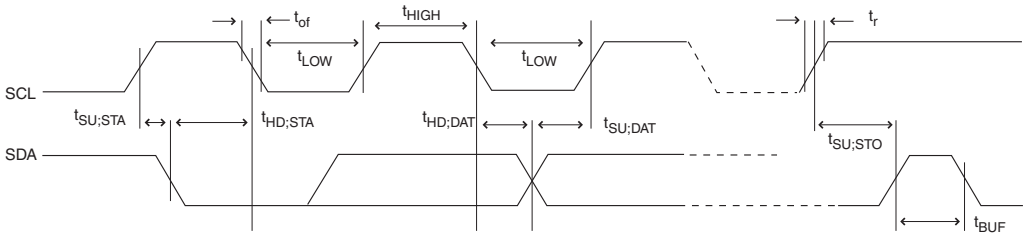
Table 28-6. 2-wire Serial Bus Requirements

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
V_{hys} (1)	Hysteresis of Schmitt Trigger Inputs		0.05 V_{CC} (2)	–	V
V_{OL} (1)	Output Low-voltage	3 mA sink current	0	0.4	V
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	ns
t_{SP} (1)	Spikes Suppressed by Input Filter		0	50 (2)	ns
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	$\frac{V_{CC}-0{,}4V}{3mA}$	$\frac{1000ns}{C_b}$	Ω
		f_{SCL} > 100 kHz	$\frac{V_{CC}-0{,}4V}{3mA}$	$\frac{300ns}{C_b}$	Ω
$t_{HD;STA}$	Hold Time (repeated) START Condition	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
t_{LOW}	Low Period of the SCL Clock	f_{SCL} ≤ 100 kHz	4.7	–	µs
		f_{SCL} > 100 kHz	1.3	–	µs
t_{HIGH}	High period of the SCL clock	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
$t_{SU;STA}$	Set-up time for a repeated START condition	f_{SCL} ≤ 100 kHz	4.7	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
$t_{HD;DAT}$	Data hold time	f_{SCL} ≤ 100 kHz	0	3.45	µs
		f_{SCL} > 100 kHz	0	0.9	µs
$t_{SU;DAT}$	Data setup time	f_{SCL} ≤ 100 kHz	250	–	ns
		f_{SCL} > 100 kHz	100	–	ns
$t_{SU;STO}$	Setup time for STOP condition	f_{SCL} ≤ 100 kHz	4.0	–	µs
		f_{SCL} > 100 kHz	0.6	–	µs
t_{BUF}	Bus free time between a STOP and START condition	f_{SCL} ≤ 100 kHz	4.7	–	µs
		f_{SCL} > 100 kHz	1.3	–	µs

Notes: 1. In ATmega48PA/88PA/168PA/328P, 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 ATmega48PA/88PA/168PA/328P 2-wire Serial Interface operation. Other devices connected to the 2-wire Serial Bus need only obey the general f_{SCL} requirement.

Figure 28-5. 2-wire Serial Bus Timing



28.8 ADC Characteristics

Table 28-7. ADC Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Resolution			10		Bits
	Absolute accuracy (Including INL, DNL, quantization error, gain and offset error)	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		2		LSB
		V_{REF} = 4V, V_{CC} = 4V, ADC clock = 1 MHz		4.5		LSB
		V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz Noise Reduction Mode		2		LSB
		V_{REF} = 4V, V_{CC} = 4V, ADC clock = 1 MHz Noise Reduction Mode		4.5		LSB
	Integral Non-Linearity (INL)	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		0.5		LSB
	Differential Non-Linearity (DNL)	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		0.25		LSB
	Gain Error	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		2		LSB
	Offset Error	V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz		2		LSB
	Conversion Time	Free Running Conversion	13		260	µs
	Clock Frequency		50		1000	kHz
AV_{CC} (1)	Analog Supply Voltage		V_{CC} - 0.3		V_{CC} + 0.3	V
V_{REF}	Reference Voltage		1.0		AV_{CC}	V
V_{IN}	Input Voltage		GND		V_{REF}	V
	Input Bandwidth			38.5		kHz
V_{INT}	Internal Voltage Reference		1.0	1.1	1.2	V
R_{REF}	Reference Input Resistance			32		kΩ
R_{AIN}	Analog Input Resistance			100		MΩ

Note: 1. AV_{CC} absolute min/max: 1.8V/5.5V

28.9 Parallel Programming Characteristics

Table 28-8. 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
t_{DVXH}	Data and Control Valid before XTAL1 High	67			ns
t_{XLXH}	XTAL1 Low to XTAL1 High	200			ns
t_{XHXL}	XTAL1 Pulse Width High	150			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67			ns
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	0			ns
t_{XLPH}	XTAL1 Low to PAGEL high	0			ns
t_{PLXH}	PAGEL low to XTAL1 high	150			ns
t_{BVPH}	BS1 Valid before PAGEL High	67			ns
t_{PHPL}	PAGEL Pulse Width High	150			ns
t_{PLBX}	BS1 Hold after PAGEL Low	67			ns
t_{WLBX}	BS2/1 Hold after $\overline{WR}$ Low	67			ns
t_{PLWL}	PAGEL Low to $\overline{WR}$ Low	67			ns
t_{BVWL}	BS1 Valid to $\overline{WR}$ Low	67			ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low	150			ns
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	ms
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	0			ns
t_{BVDV}	BS1 Valid to DATA valid	0		250	ns
t_{OLDV}	$\overline{OE}$ Low to DATA Valid			250	ns
t_{OHDZ}	$\overline{OE}$ High to DATA Tri-stated			250	ns

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.

Figure 28-6. Parallel Programming Timing, Including some General Timing Requirements

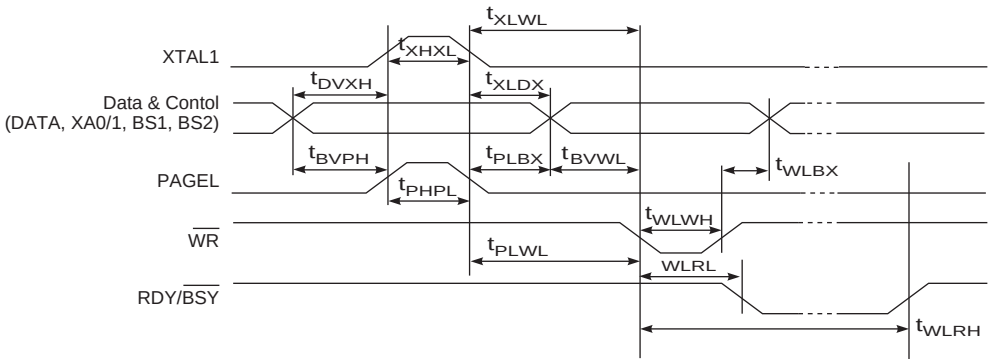
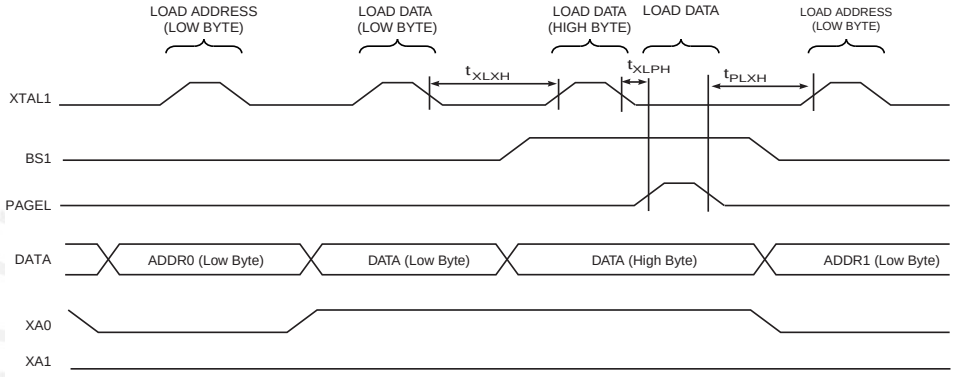
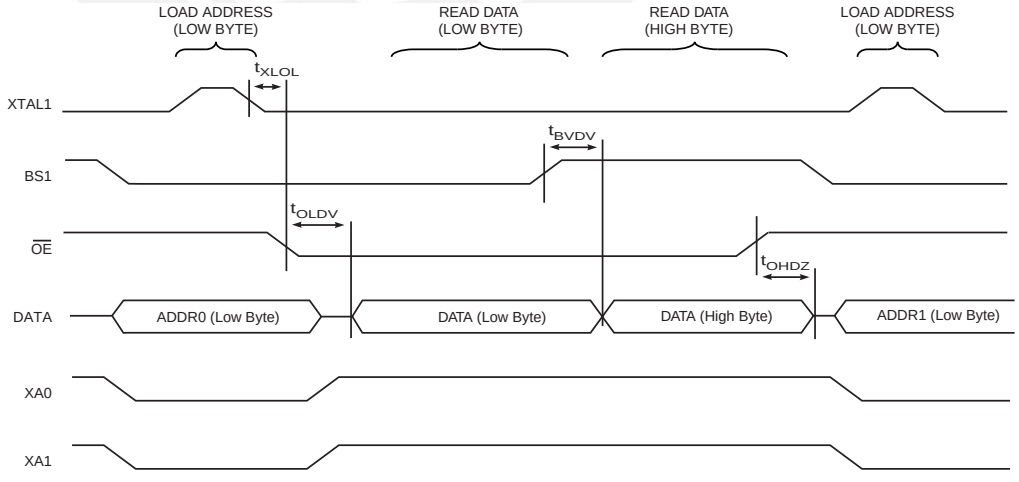


Figure 28-7. Parallel Programming Timing, Loading Sequence with Timing Requirements[1]



Note: 1. The timing requirements shown in Figure 28-6 (i.e., t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to loading operation.

Figure 28-8. Parallel Programming Timing, Reading Sequence (within the Same Page) with Timing Requirements[1]



Note: 1. The timing requirements shown in Figure 28-6 (i.e., t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to reading operation.

29. 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 square wave generator with rail-to-rail output is used as clock source.

All Active- and Idle current consumption measurements are done with all bits in the PRR register set and thus, the corresponding I/O modules are turned off. Also the Analog Comparator is disabled during these measurements. The ”ATmega88PA: Supply Current of IO Modules” on page 356 and page 380 shows the additional current consumption compared to I_{CC} Active and I_{CC} Idle for every I/O module controlled by the Power Reduction Register. See ”Power Reduction Register” on page 42 for details.

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.

29.1 ATmega48PA Typical Characteristics

29.1.1 Active Supply Current

Figure 29-1. ATmega48PA: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

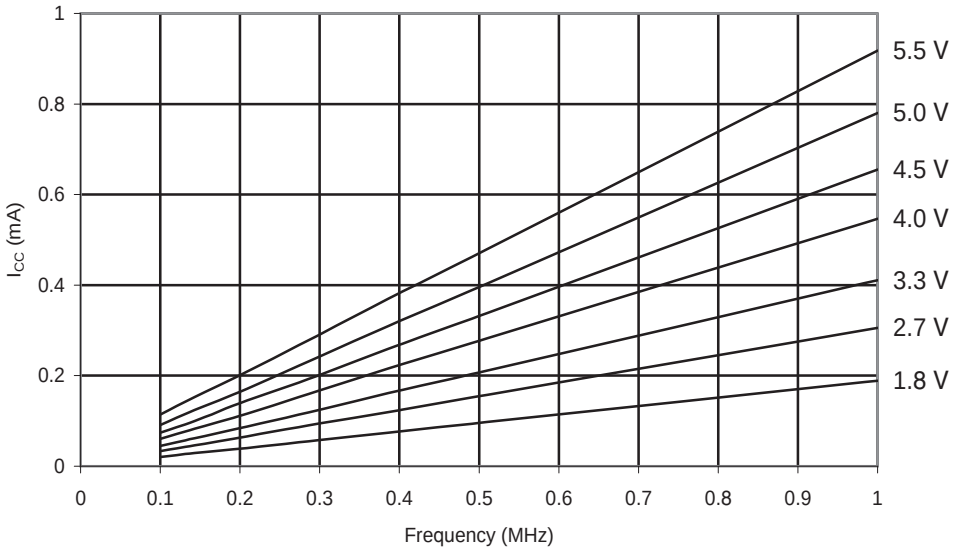


Figure 29-2. ATmega48PA: Active Supply Current vs. Frequency (1-20 MHz)

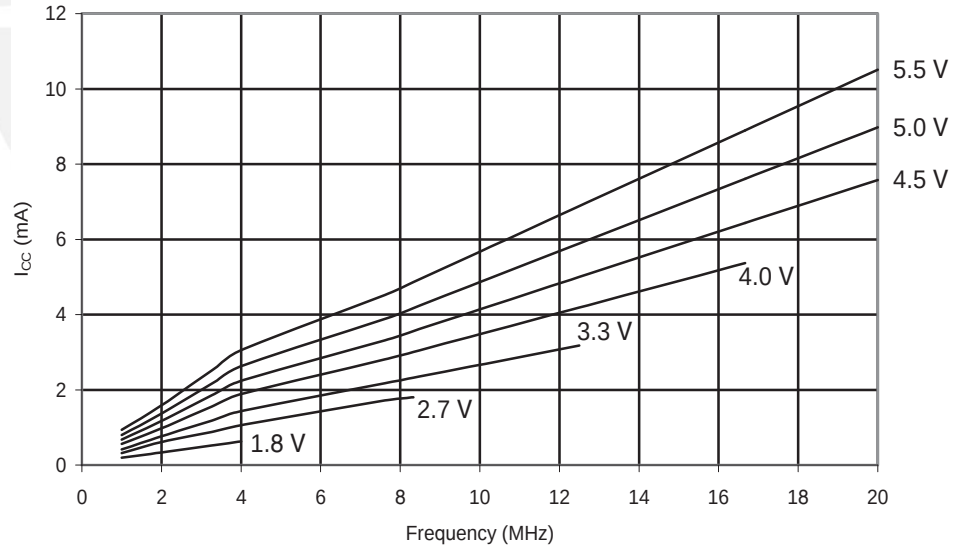


Figure 29-3. ATmega48PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

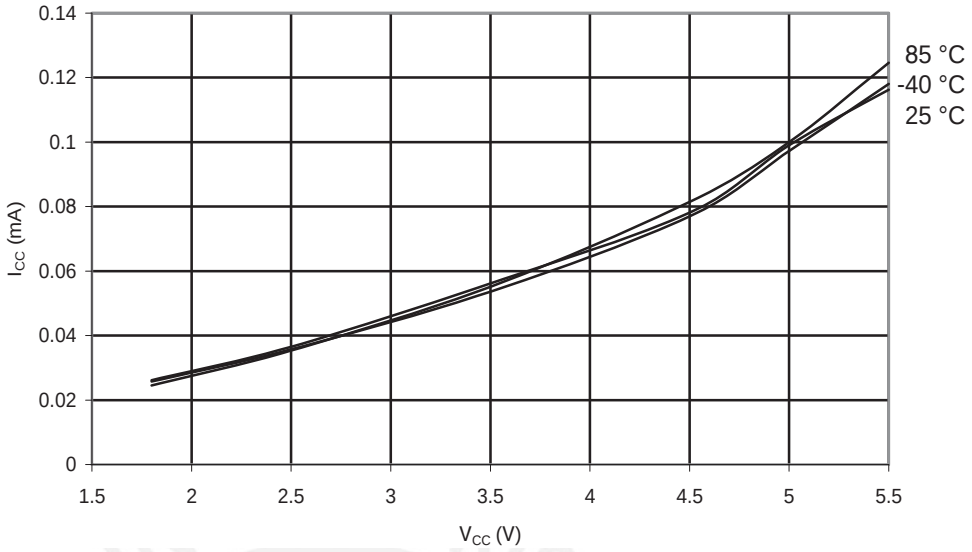


Figure 29-4. ATmega48PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

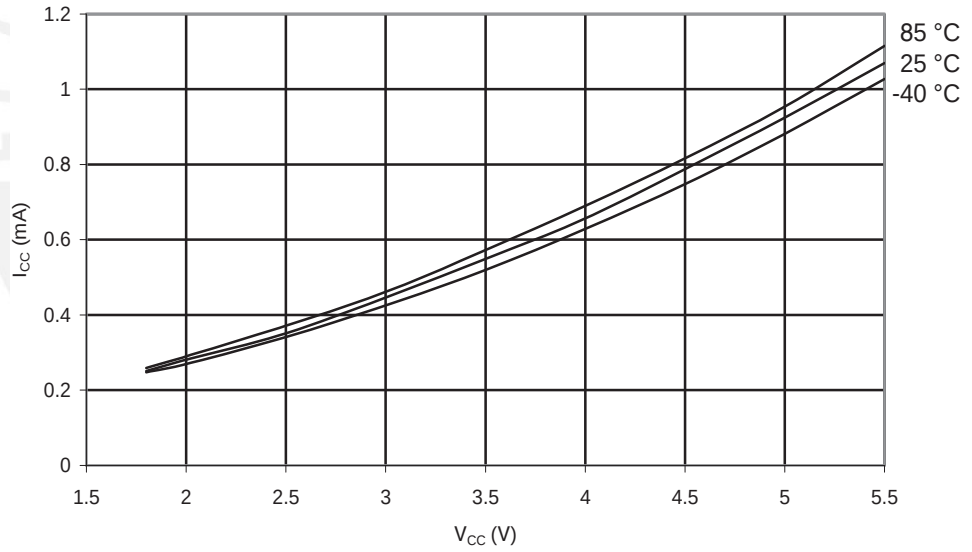
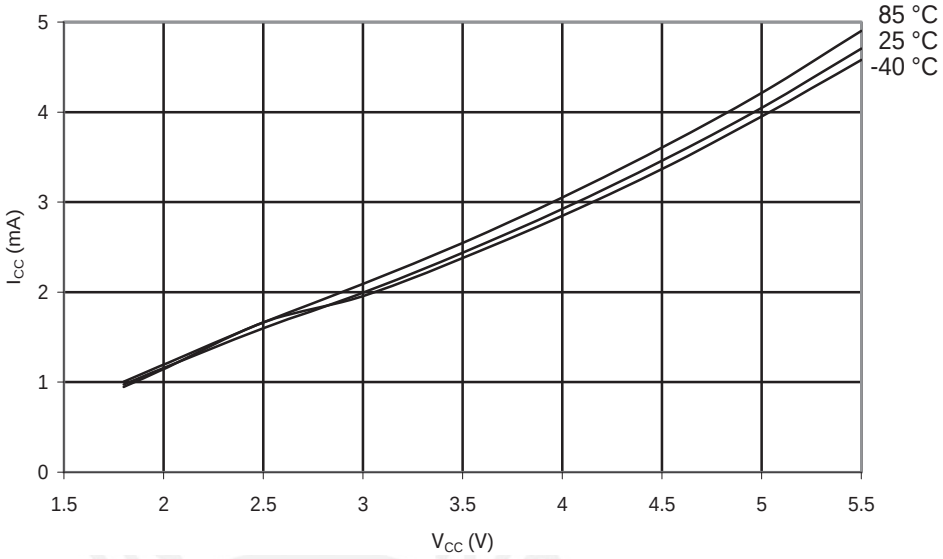


Figure 29-5. ATmega48PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.1.2 Idle Supply Current

Figure 29-6. ATmega48PA: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

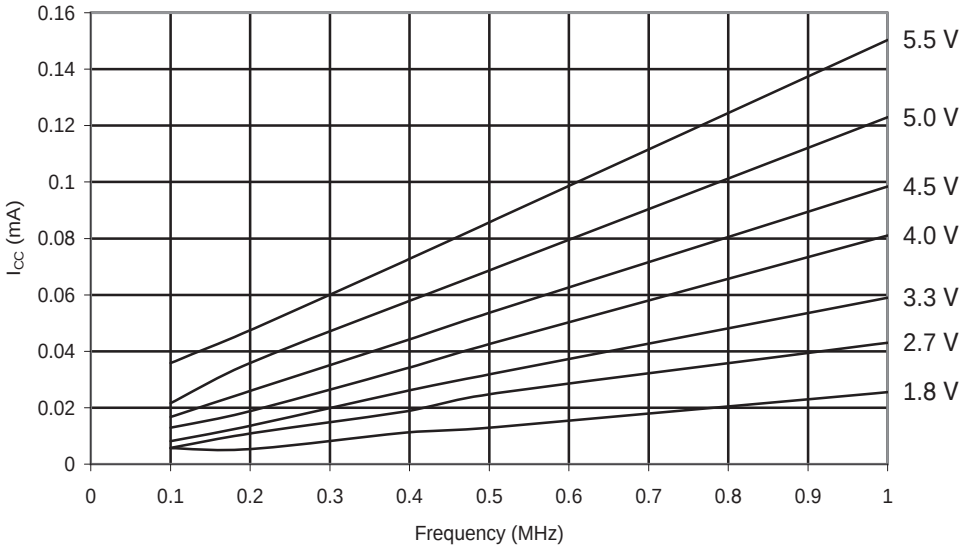


Figure 29-7. ATmega48PA: Idle Supply Current vs. Frequency (1-20 MHz)

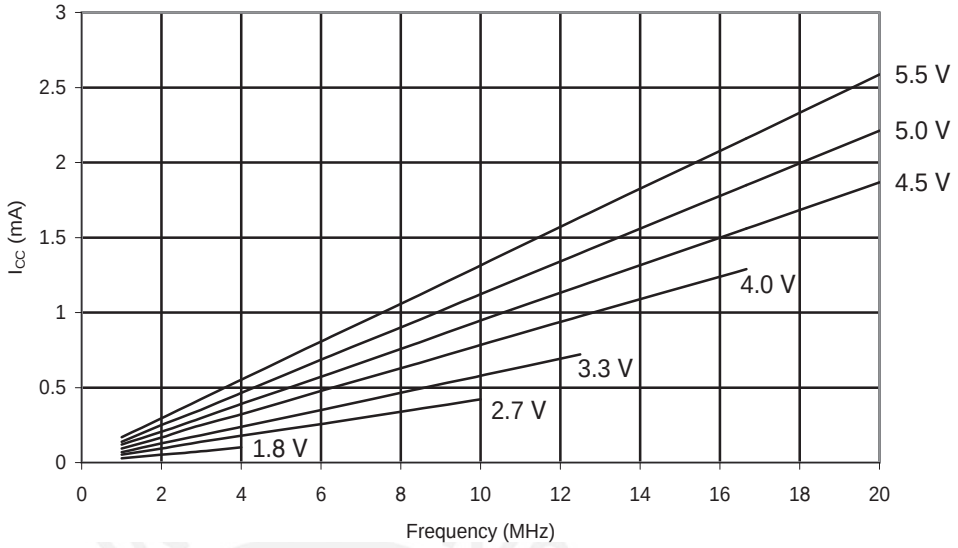


Figure 29-8. ATmega48PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

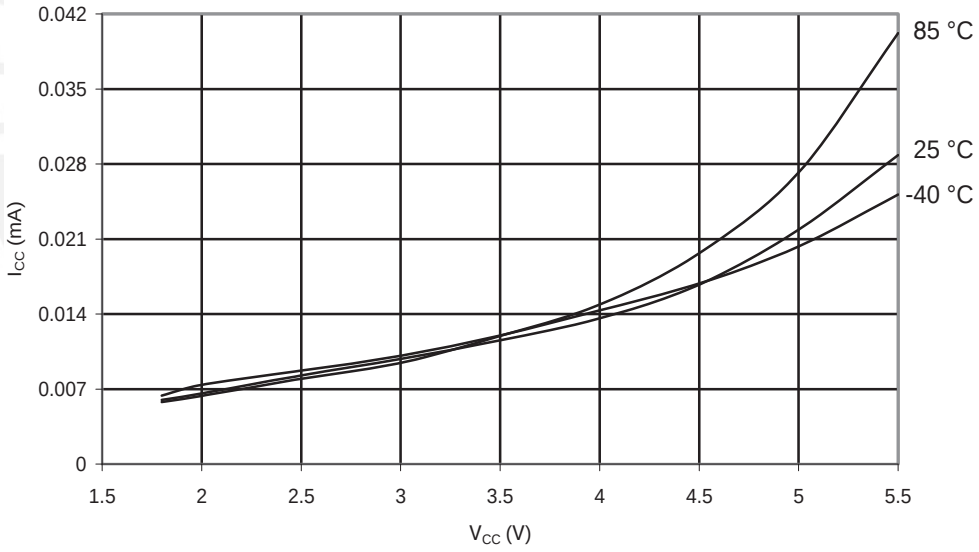


Figure 29-9. ATmega48PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

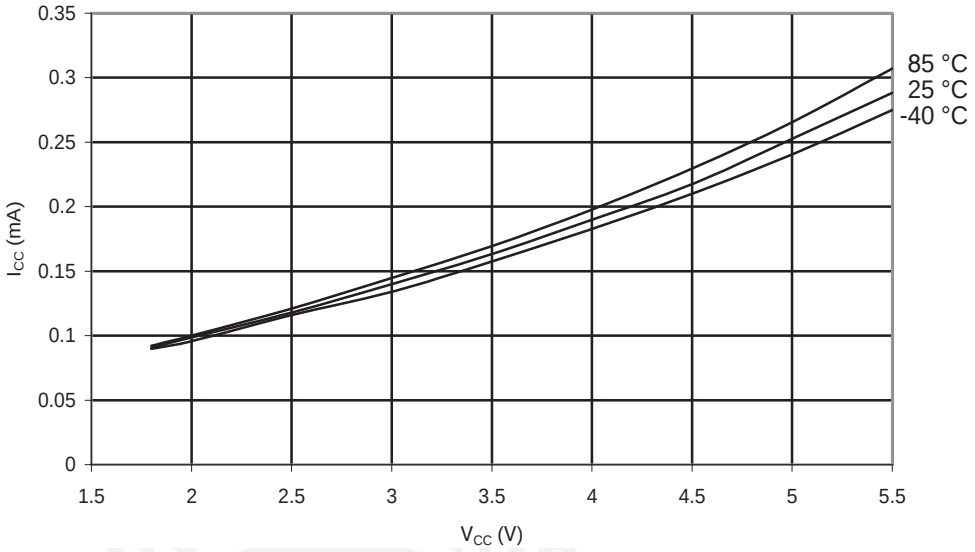
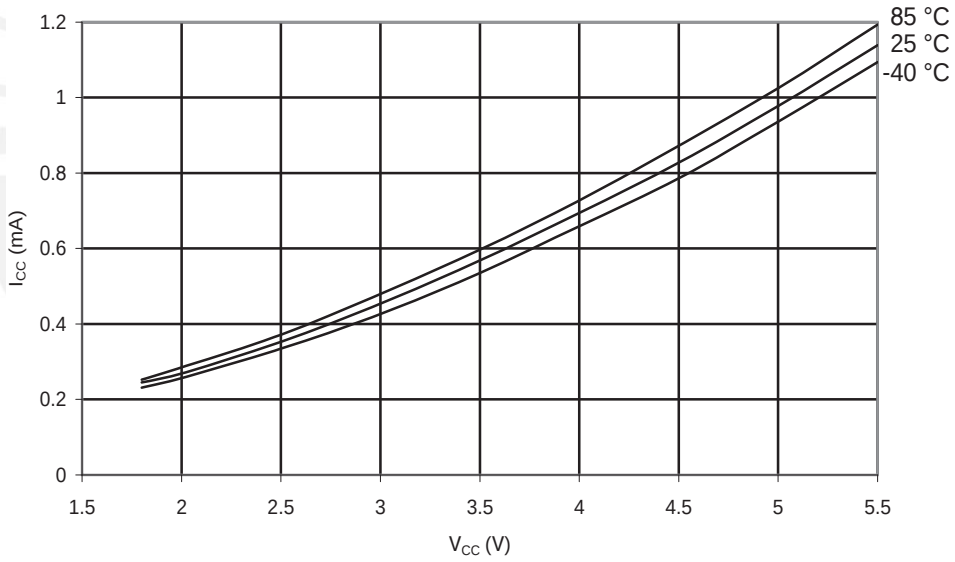


Figure 29-10. ATmega48PA: Idle Supply Current vs. Vcc (Internal RC Oscillator, 8 MHz)



29.1.3 ATmega48PA: **Supply Current of IO Modules**

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See ”Power Reduction Register” on page 42 for details.

Table 29-1. ATmega48PA: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V_{CC} = 2V, F = 1 MHz	V_{CC} = 3V, F = 4 MHz	V_{CC} = 5V, F = 8 MHz
PRUSART0	2.9 uA	20.7 uA	97.4 uA
PRTWI	6.0 uA	44.8 uA	219.7 uA
PRTIM2	5.0 uA	34.5 uA	141.3 uA
PRTIM1	3.6 uA	24.4 uA	107.7 uA
PRTIM0	1.4 uA	9.5 uA	38.4 uA
PRSPI	5.0 uA	38.0 uA	190.4 uA
PRADC	6.1 uA	47.4 uA	244.7 uA

Table 29-2. ATmega48PA: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-1 on page 327 and Figure 29-2 on page 327)	Additional Current consumption compared to Idle with external clock (see Figure 29-6 on page 329 and Figure 29-7 on page 330)
PRUSART0	1.8%	11.4%
PRTWI	3.9%	20.6%
PRTIM2	2.9%	15.7%
PRTIM1	2.1%	11.2%
PRTIM0	0.8%	4.2%
PRSPI	3.3%	17.6%
PRADC	4.2%	22.1%

It is possible to calculate the typical current consumption based on the numbers from Table 29-2 on page 332 for other V_{CC} and frequency settings than listed in Table 29-1 on page 332.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-2 on page 332, third column, we see that we need to add 11.2% for the TIMER1, 22.1% for the ADC, and 17.6% for the SPI module. Reading from **Figure 29-6 on page 329**, we find that the idle current consumption is ~0.028 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CCtotal} \approx 0.028\ mA \cdot (1 + 0.112 + 0.221 + 0.176) \approx 0.042\ mA$$

29.1.4 Power-down Supply Current

Figure 29-11. ATmega48PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

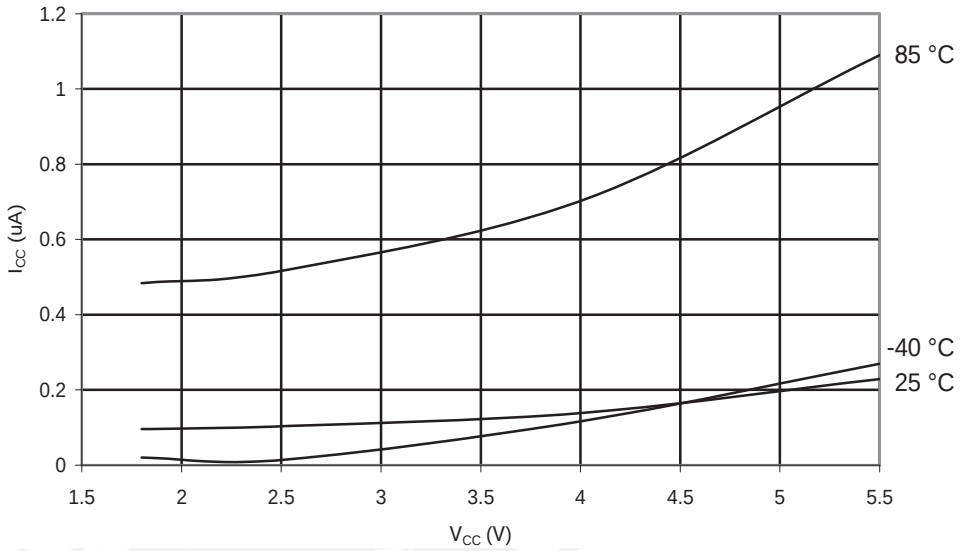
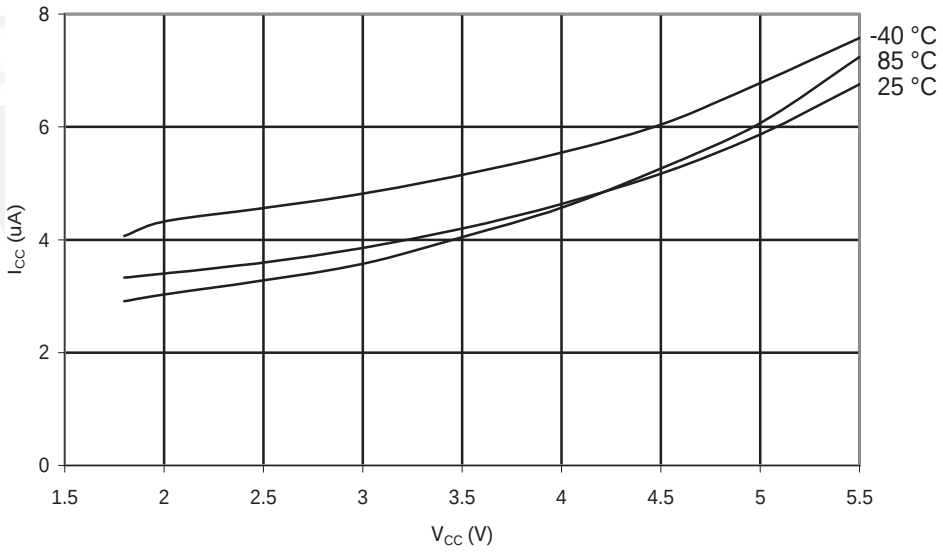
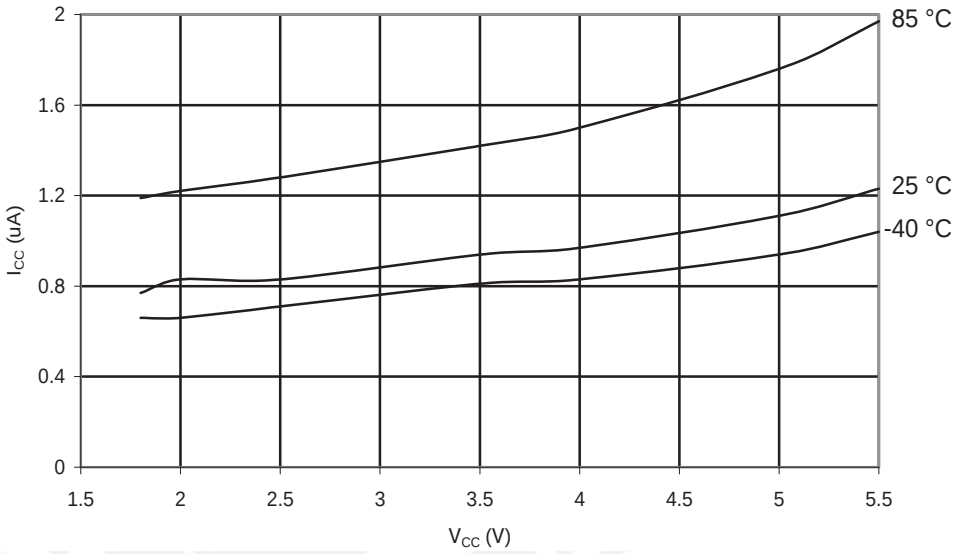


Figure 29-12. ATmega48PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



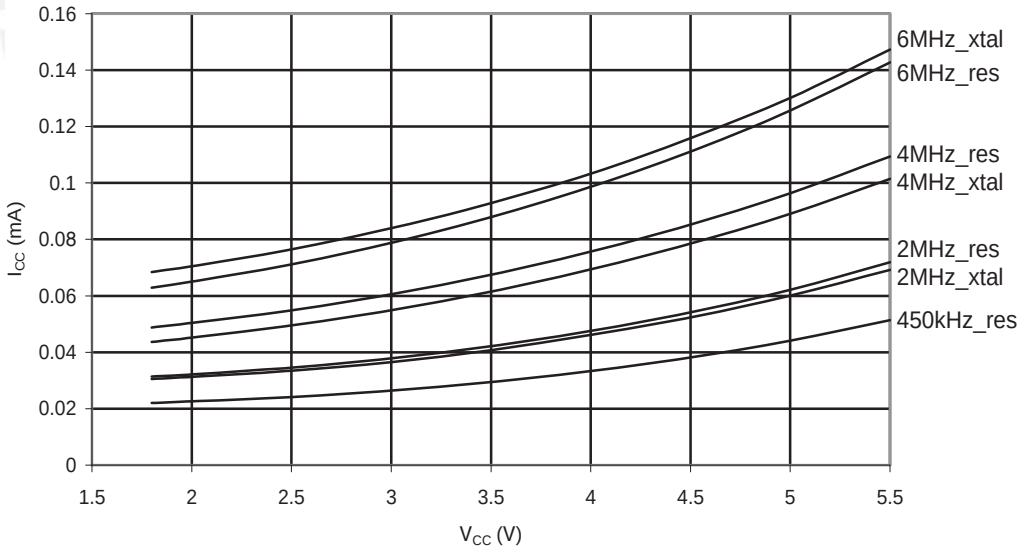
29.1.5 Power-save Supply Current

Figure 29-13. ATmega48PA: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.1.6 Standby Supply Current

Figure 29-14. ATmega48PA: Standby Supply Current vs. Vcc (Watchdog Timer Disabled)



29.1.7 Pin Pull-Up

Figure 29-15. ATmega48PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 1.8 V)

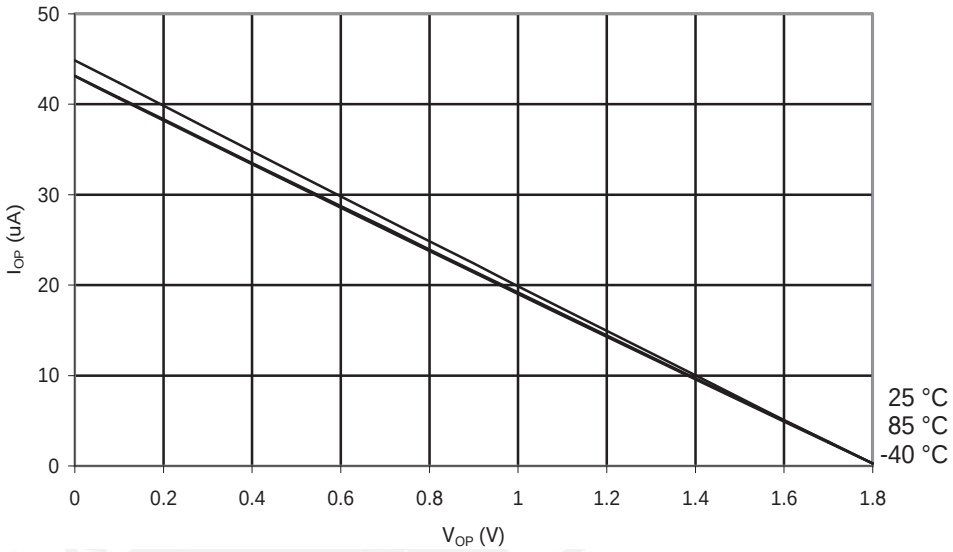


Figure 29-16. ATmega48PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 2.7 V)

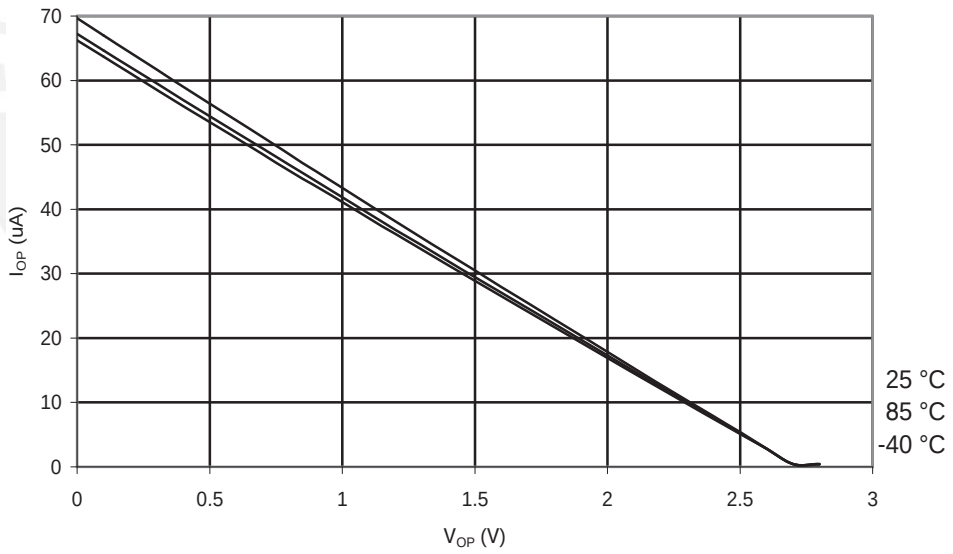


Figure 29-17. ATmega48PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 5 V)

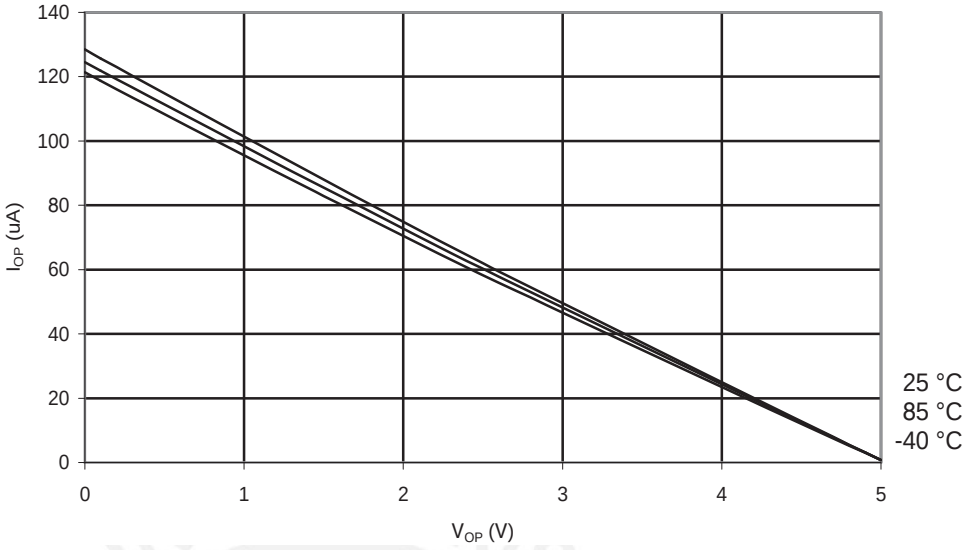


Figure 29-18. ATmega48PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 1.8 V)

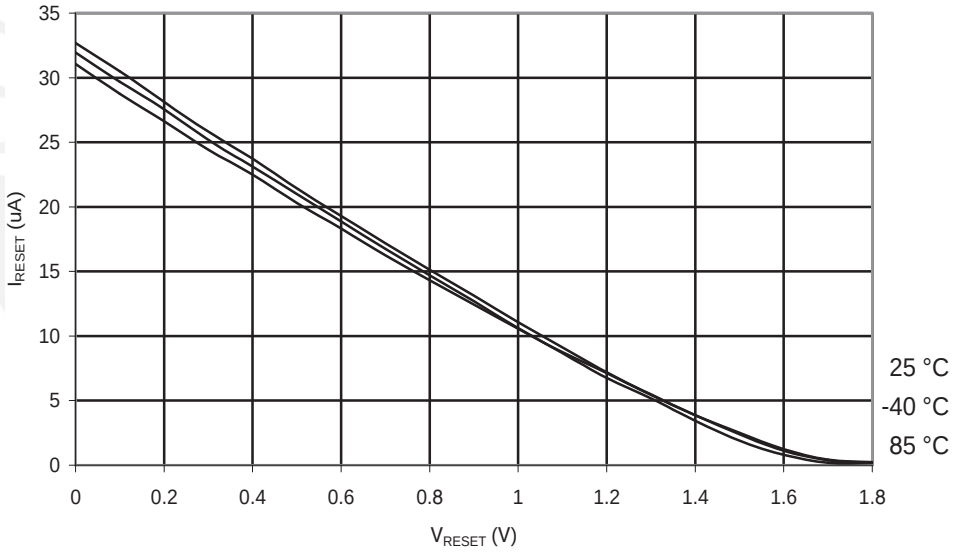


Figure 29-19. ATmega48PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 2.7 V)

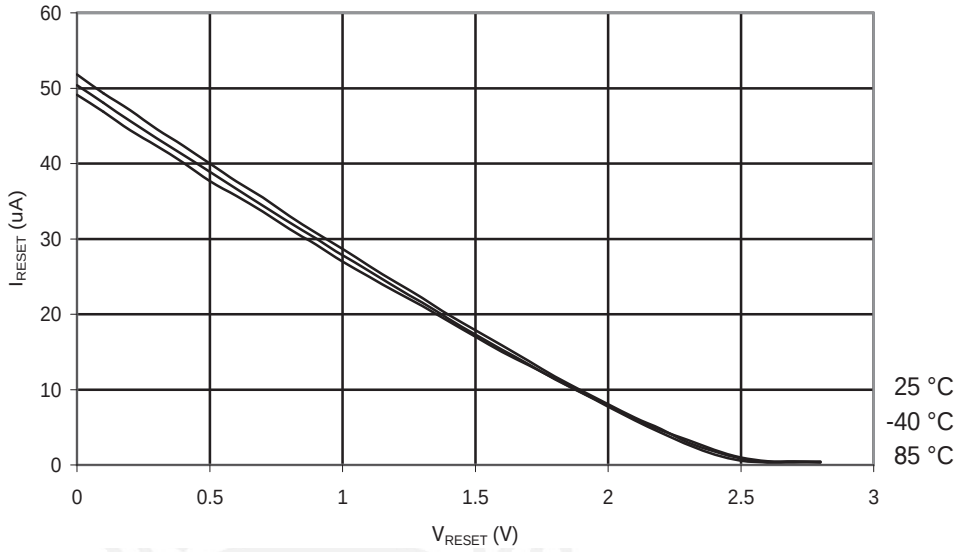
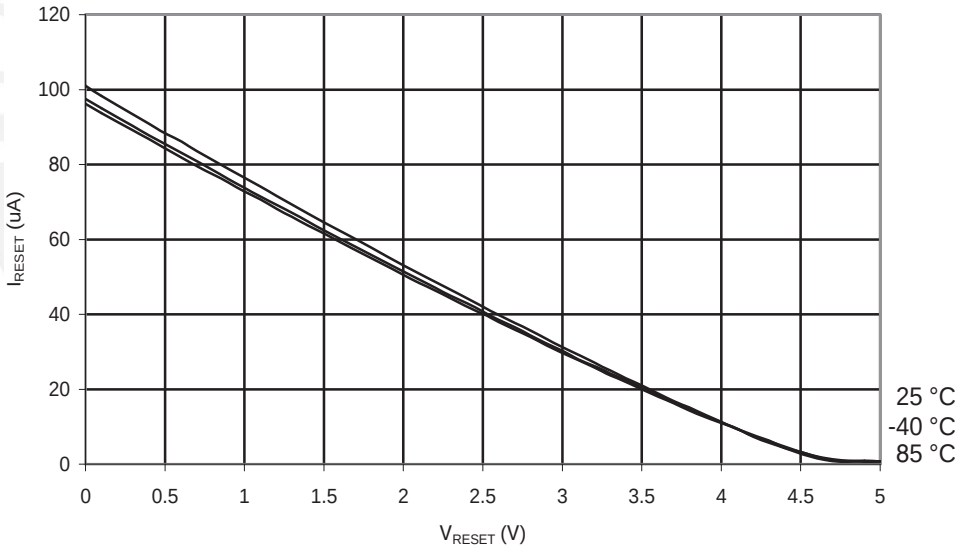


Figure 29-20. ATmega48PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 5 V)



29.1.8 Pin Driver Strength

Figure 29-21. ATmega48PA: I/O Pin Output Voltage vs. Sink Current(V_{CC} = 3 V)

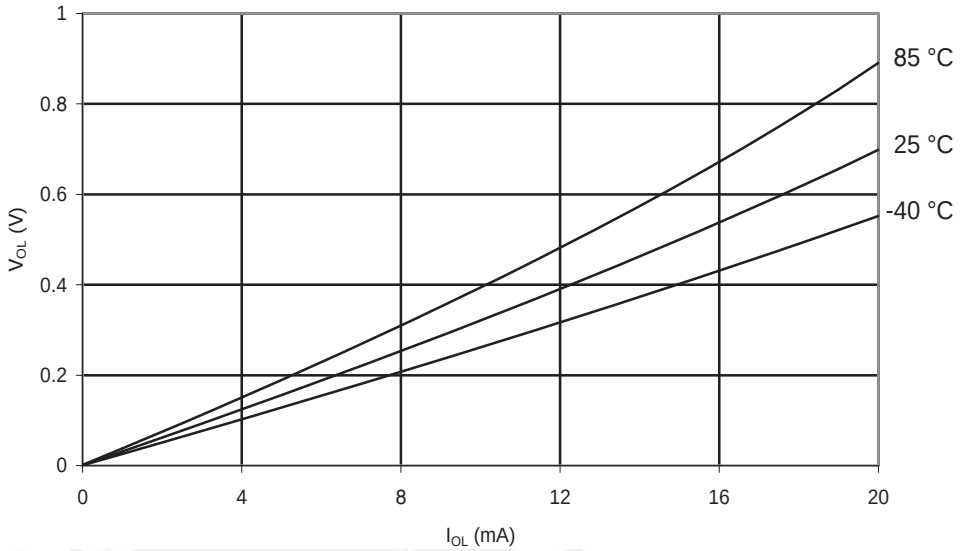


Figure 29-22. ATmega48PA: I/O Pin Output Voltage vs. Sink Current(V_{CC} = 5 V)

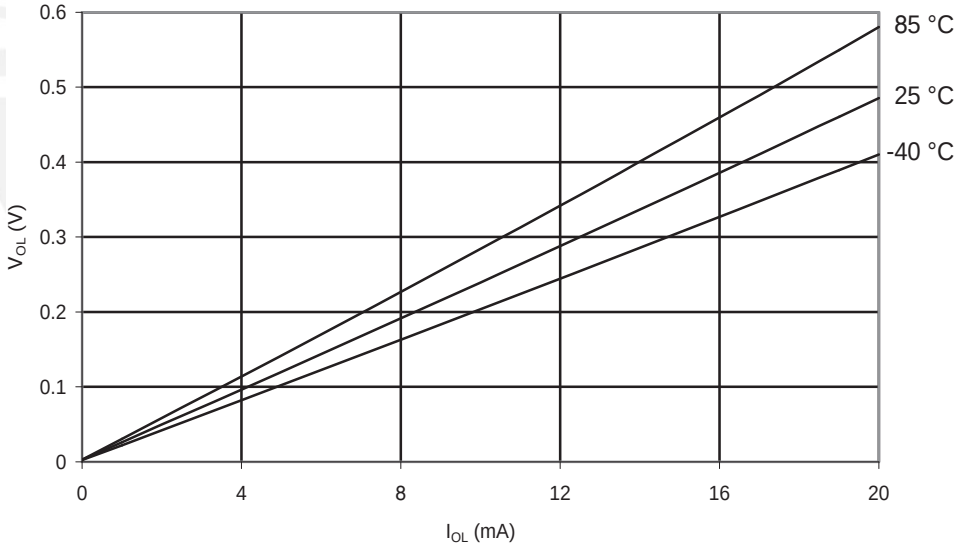


Figure 29-23. ATmega48PA: I/O Pin Output Voltage vs. Source Current(Vcc = 3 V)

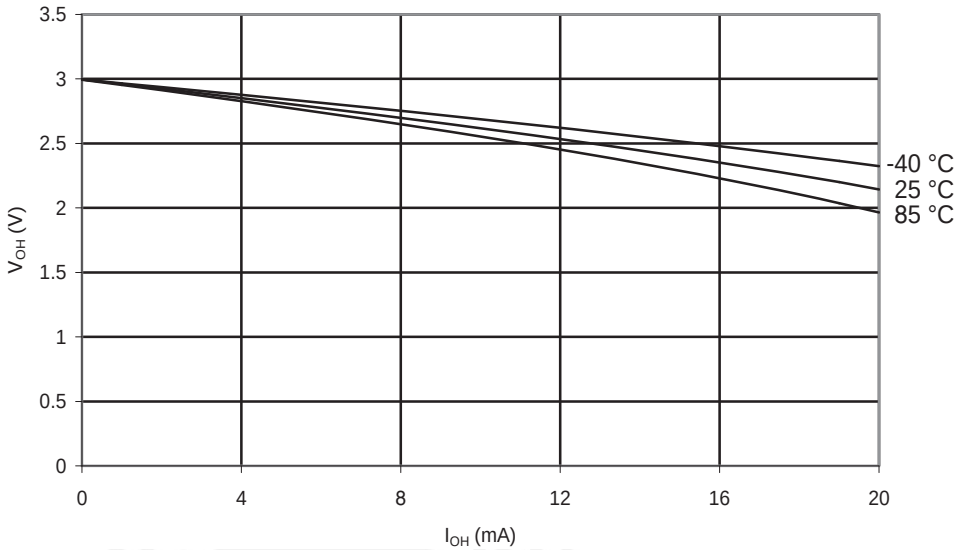
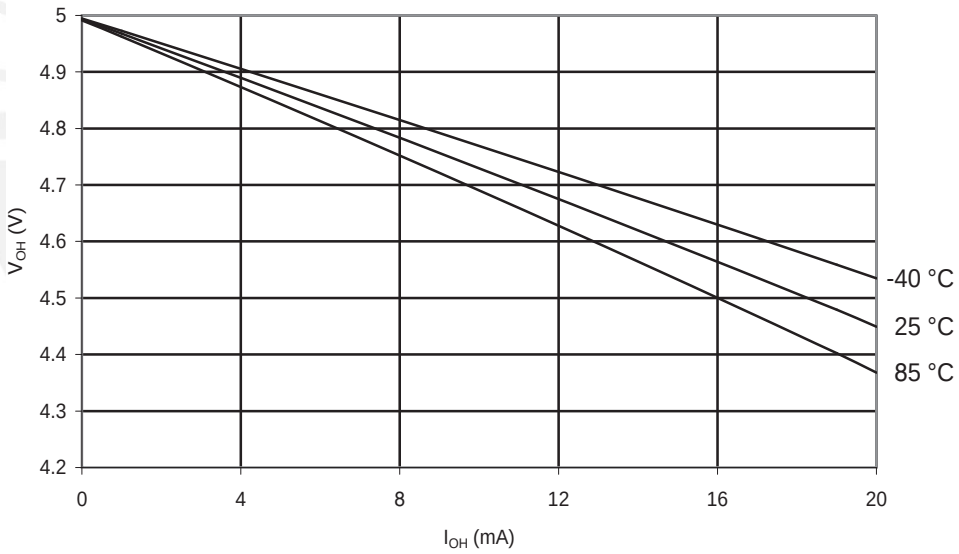


Figure 29-24. ATmega48PA: I/O Pin Output Voltage vs. Source Current(V_{CC} = 5 V)



29.1.9 Pin Threshold and Hysteresis

Figure 29-25. ATmega48PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

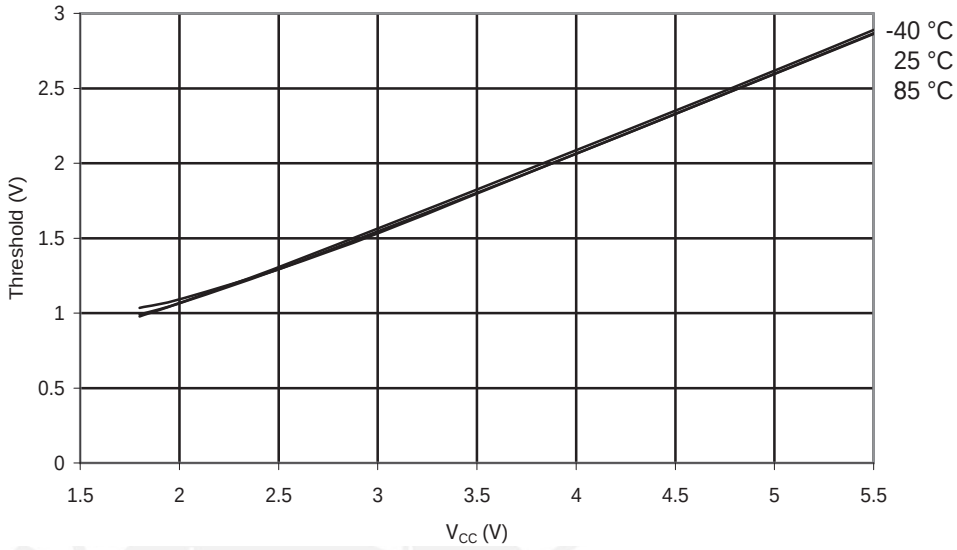


Figure 29-26. ATmega48PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

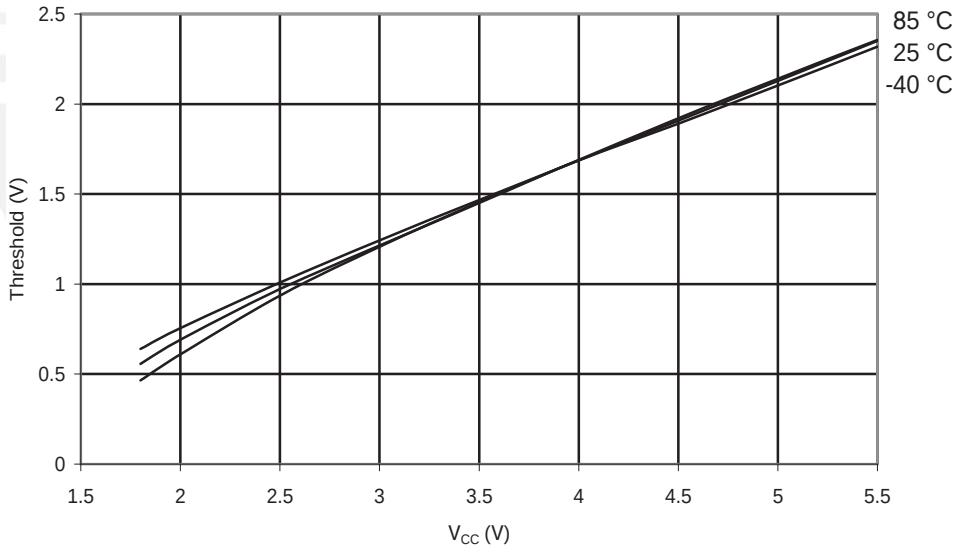


Figure 29-27. ATmega48PA: I/O Pin Input Hysteresis vs. V_{CC}

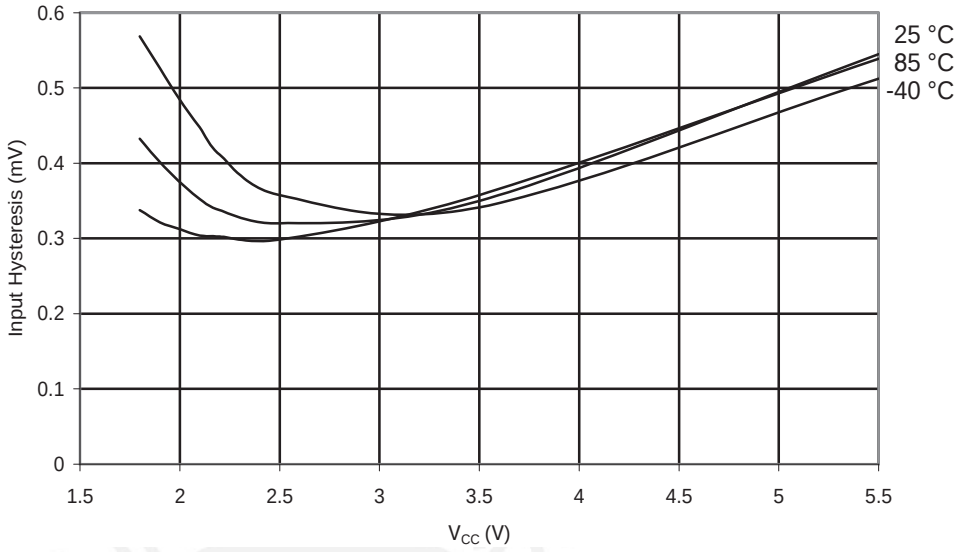


Figure 29-28. ATmega48PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

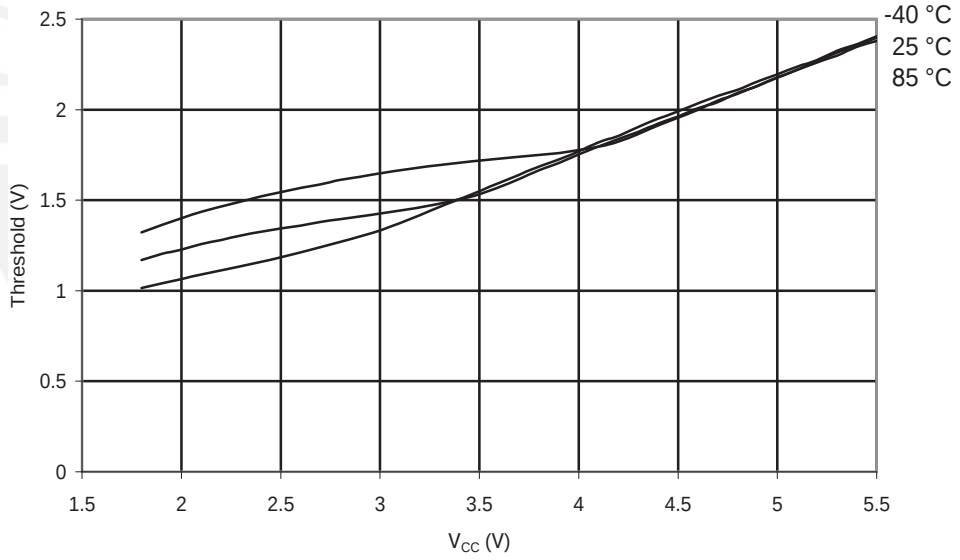


Figure 29-29. ATmega48PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

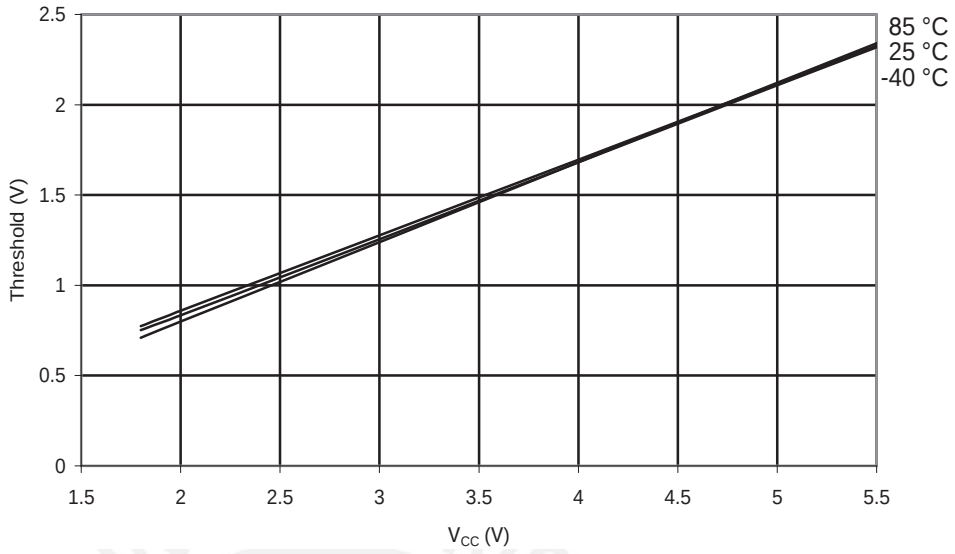
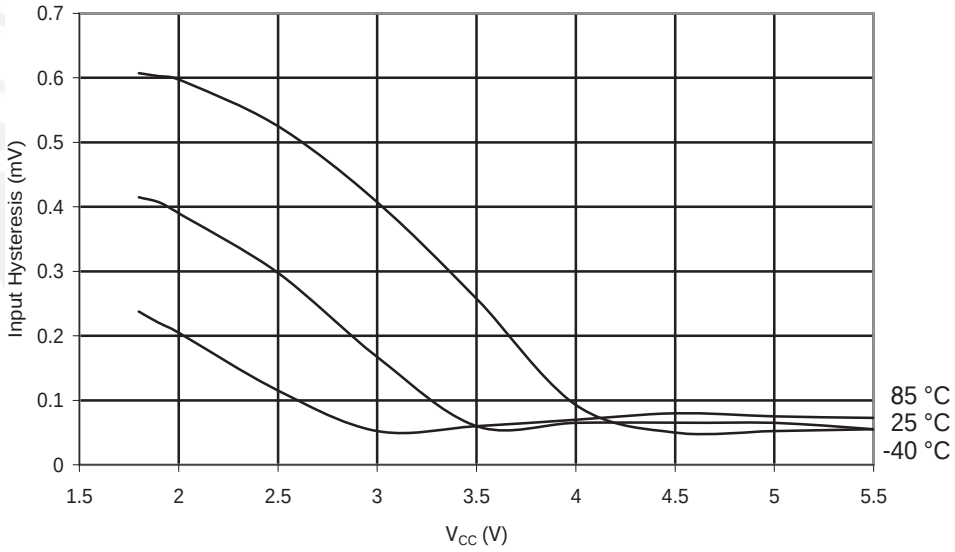


Figure 29-30. ATmega48PA: Reset Pin Input Hysteresis vs. V_{CC}



29.1.10 BOD Threshold

Figure 29-31. ATmega48PA: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

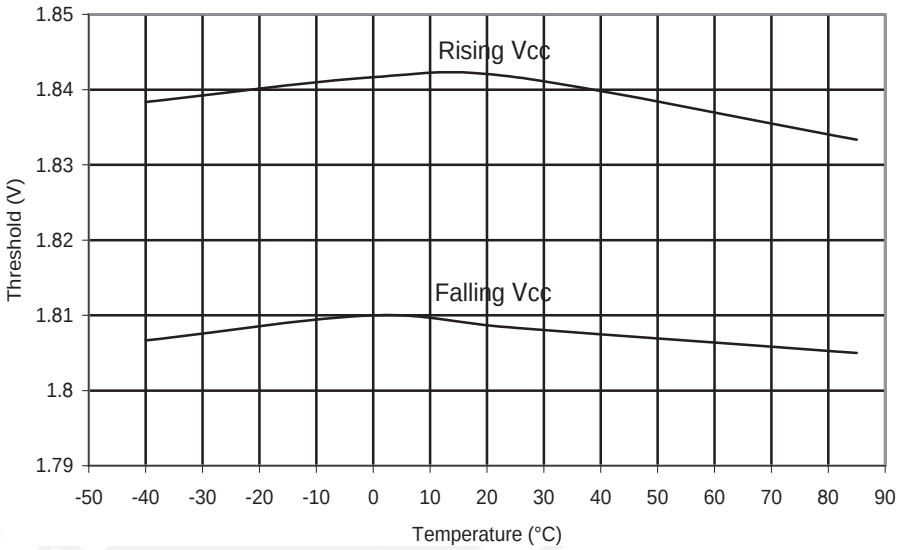


Figure 29-32. ATmega48PA: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

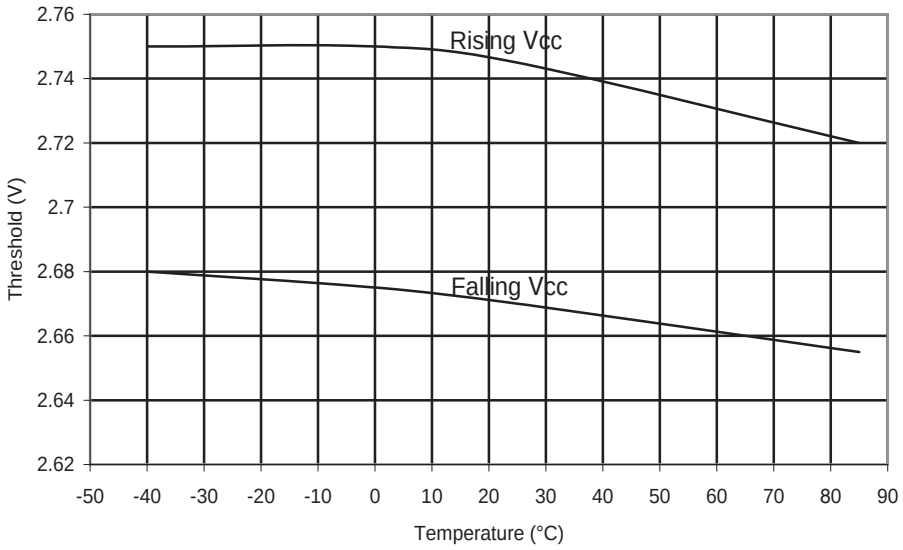
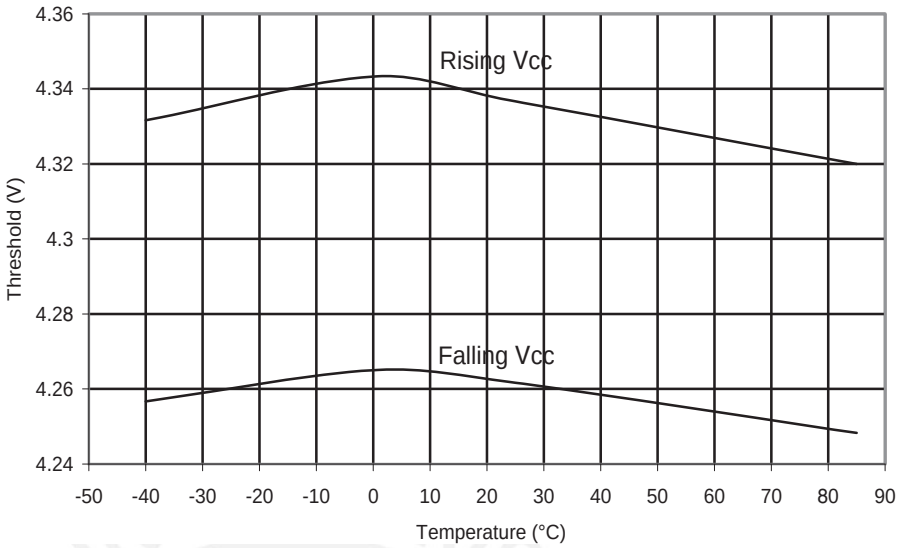


Figure 29-33. ATmega48PA: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.1.11 Internal Oscilllator Speed

Figure 29-34. ATmega48PA: Watchdog Oscillator Frequency vs. Temperature

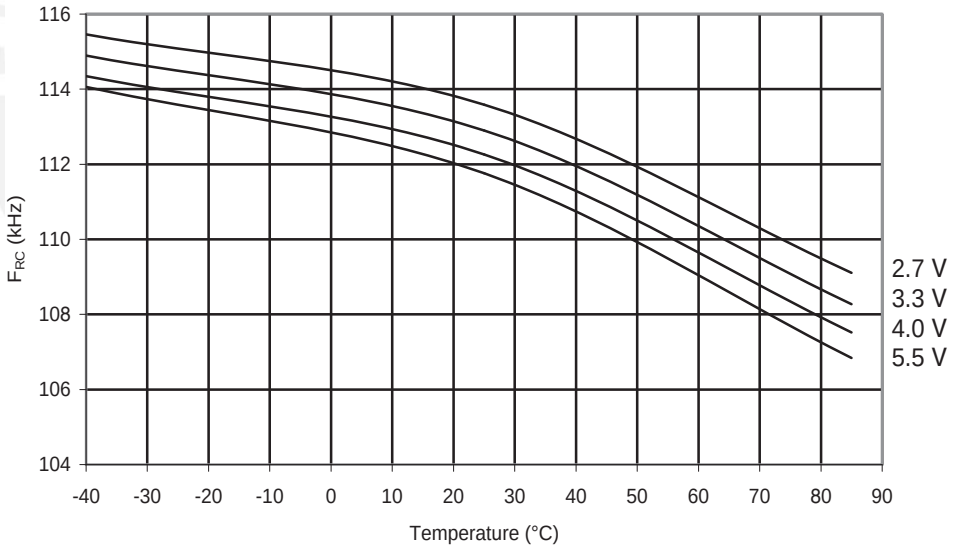


Figure 29-35. ATmega48PA: Watchdog Oscillator Frequency vs. V_{CC}

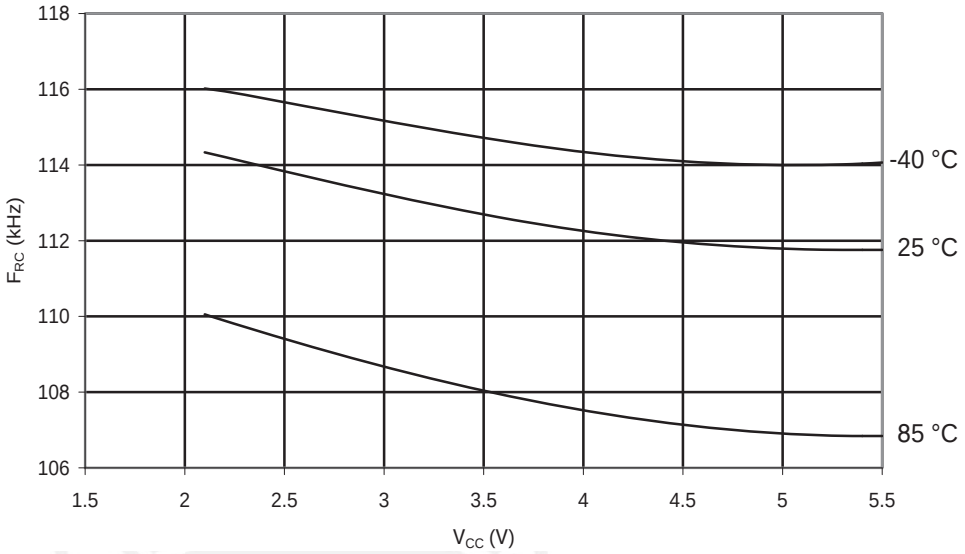


Figure 29-36. ATmega48PA: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

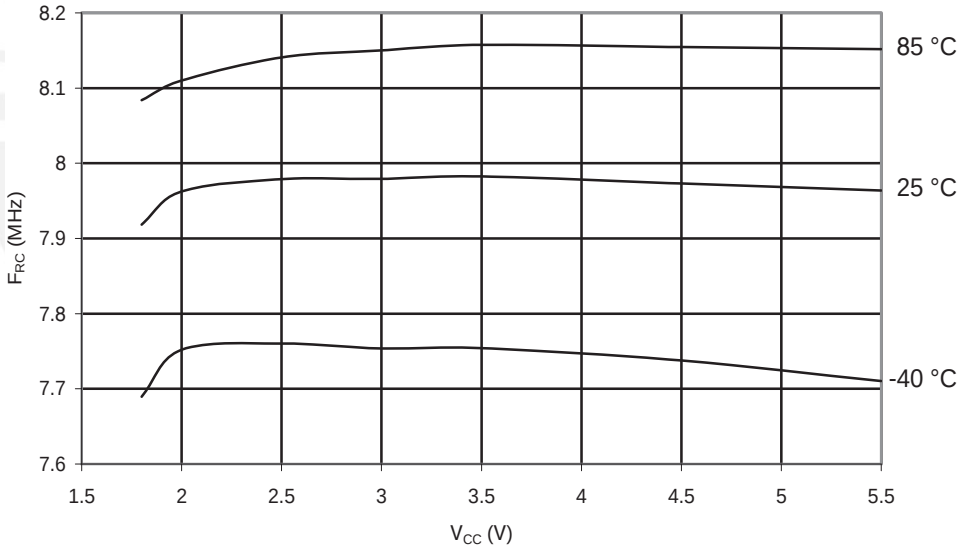


Figure 29-37. ATmega48PA: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

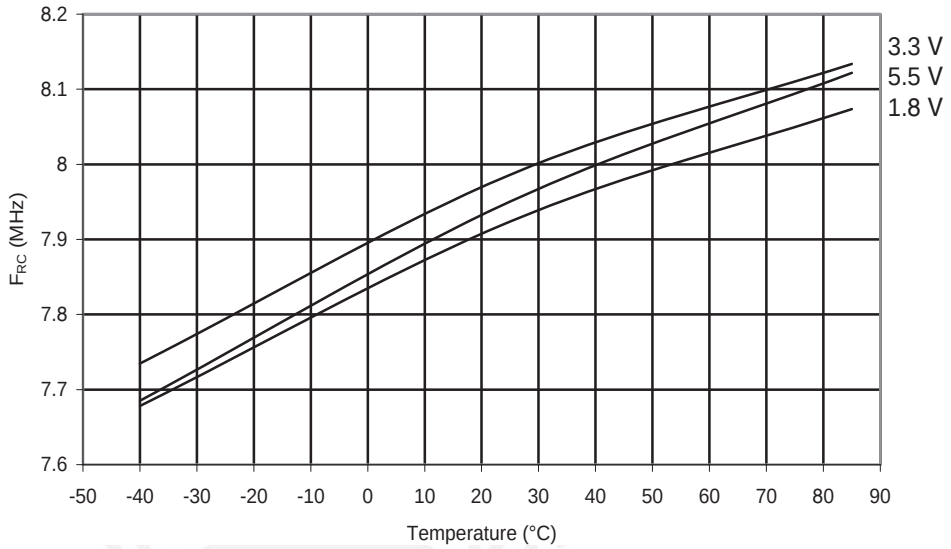
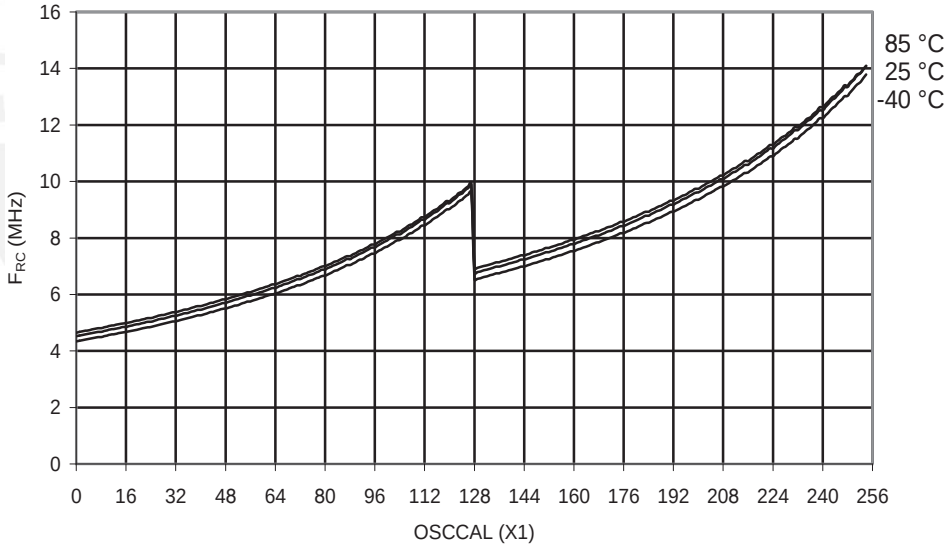


Figure 29-38. ATmega48PA: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.1.12 Current Consumption of Peripheral Units

Figure 29-39. ATmega48PA: ADC Current vs. V_{CC} ($AREF = AV_{CC}$)

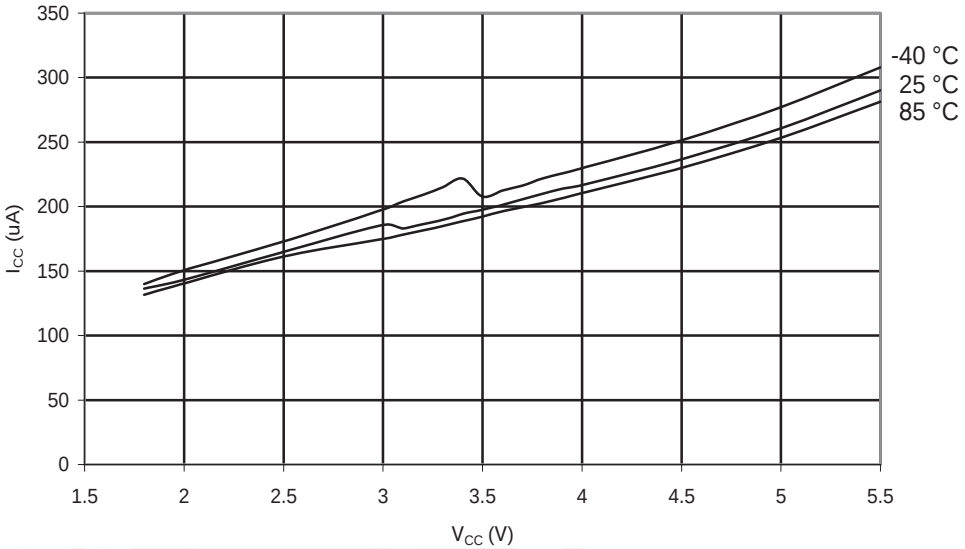


Figure 29-40. ATmega48PA: Analog Comparator Current vs. V_{CC}

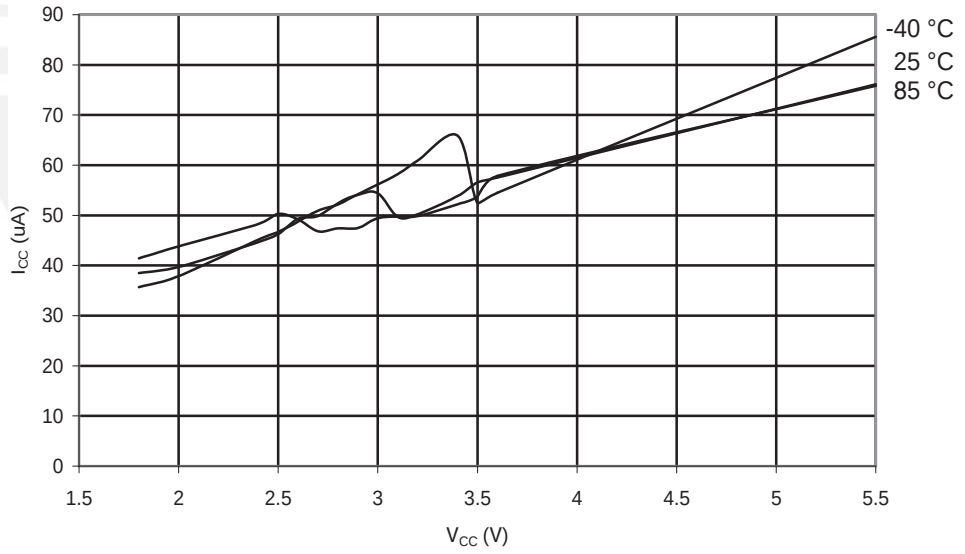


Figure 29-41. ATmega48PA: AREF External Reference Current vs. V_{CC}

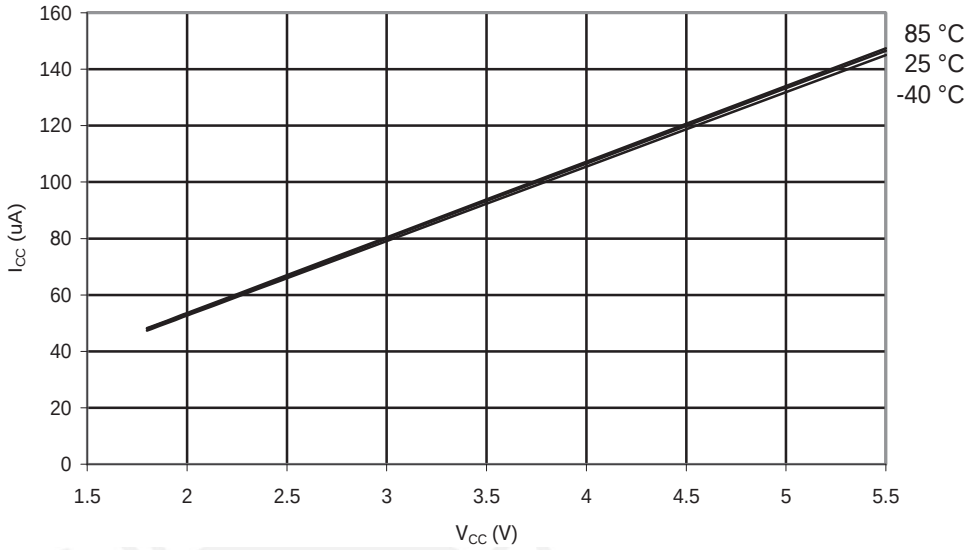


Figure 29-42. ATmega48PA: Brownout Detector Current vs. V_{CC}

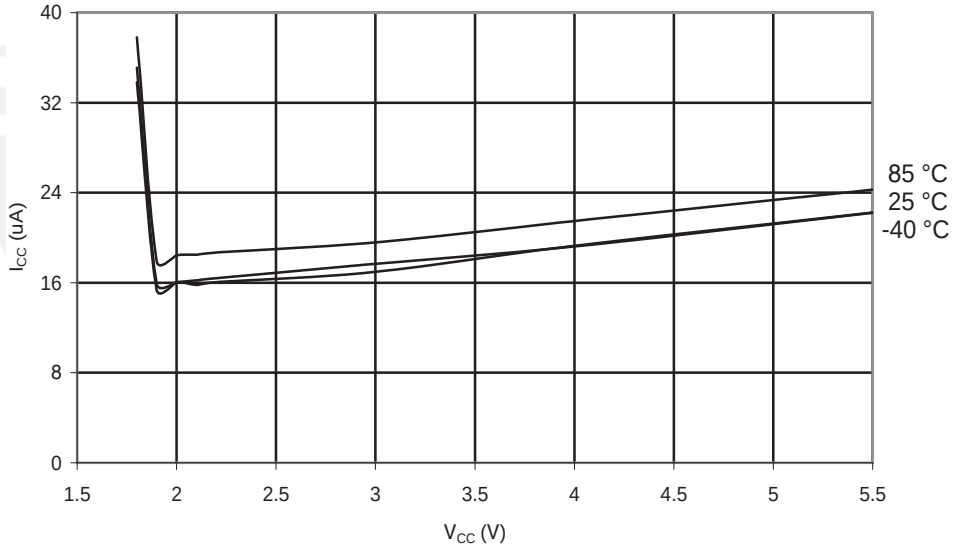
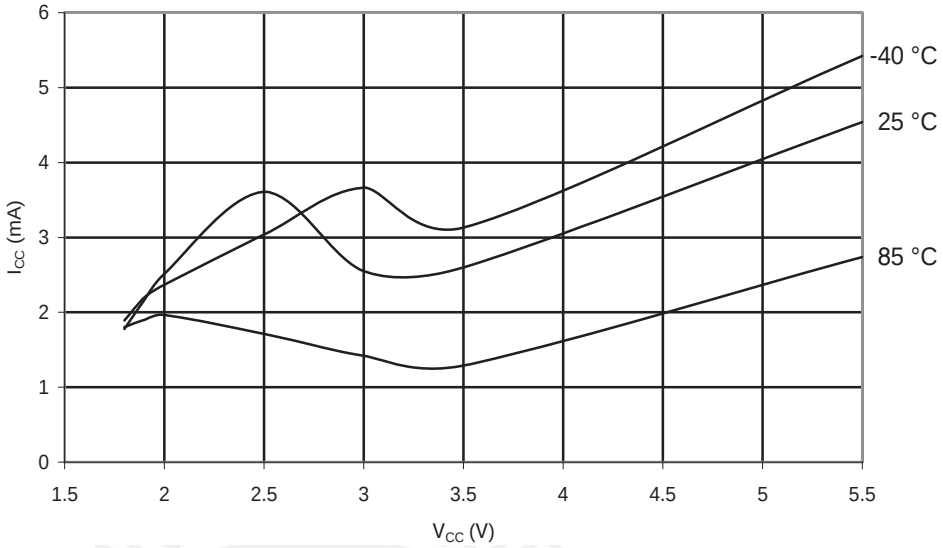


Figure 29-43. ATmega48PA: Programming Current vs. V_{CC}



29.1.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-44. ATmega48PA: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

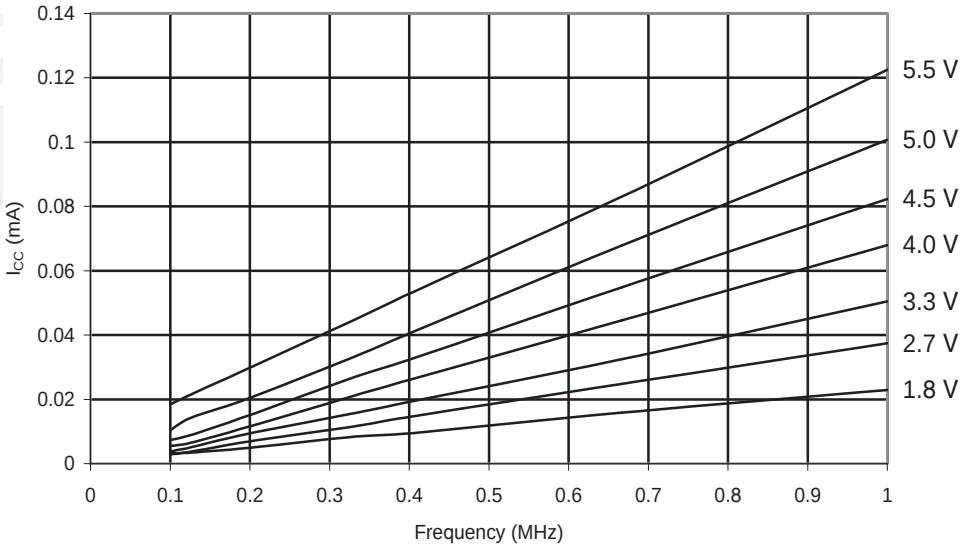


Figure 29-45. ATmega48PA: Reset Supply Current vs. Frequency (1 - 20 MHz)

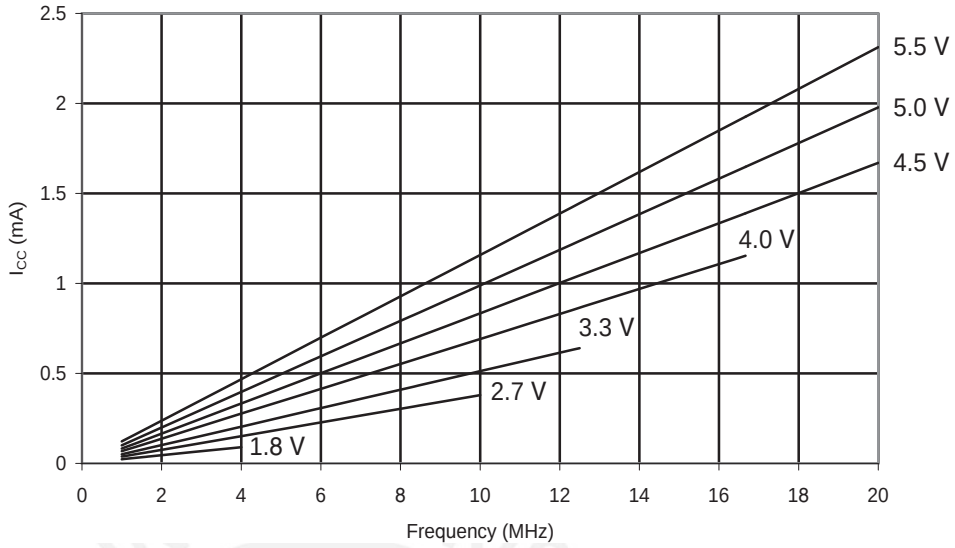
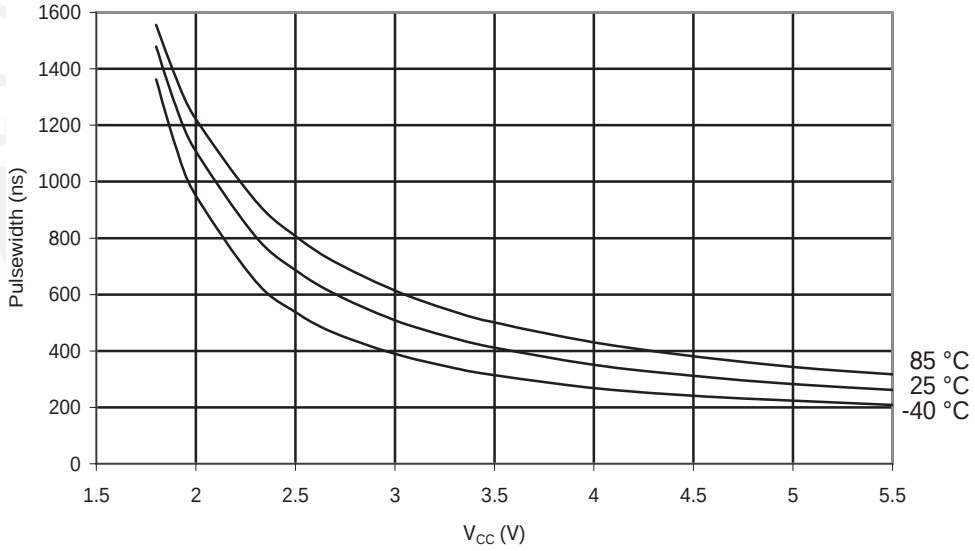


Figure 29-46. ATmega48PA: Minimum Reset Pulse width vs. V_{CC}



29.2 ATmega88PA Typical Characteristics

29.2.1 Active Supply Current

Figure 29-47. ATmega88PA: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

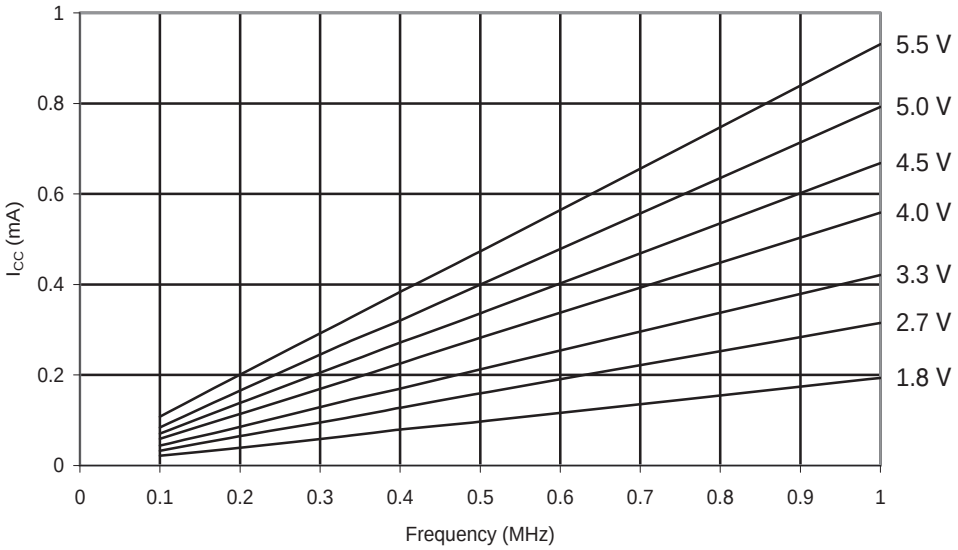


Figure 29-48. ATmega88PA: Active Supply Current vs. Frequency (1 - 20 MHz)

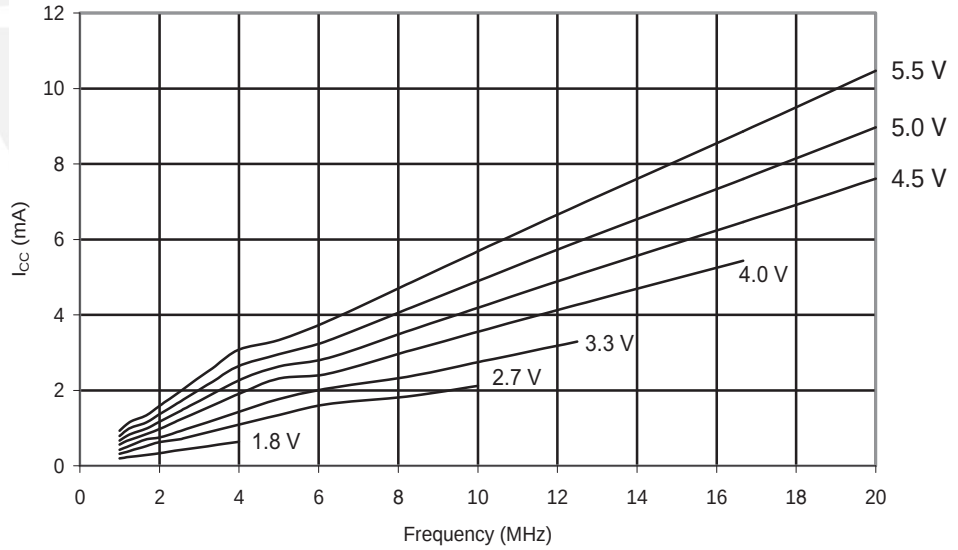


Figure 29-49. ATmega88PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

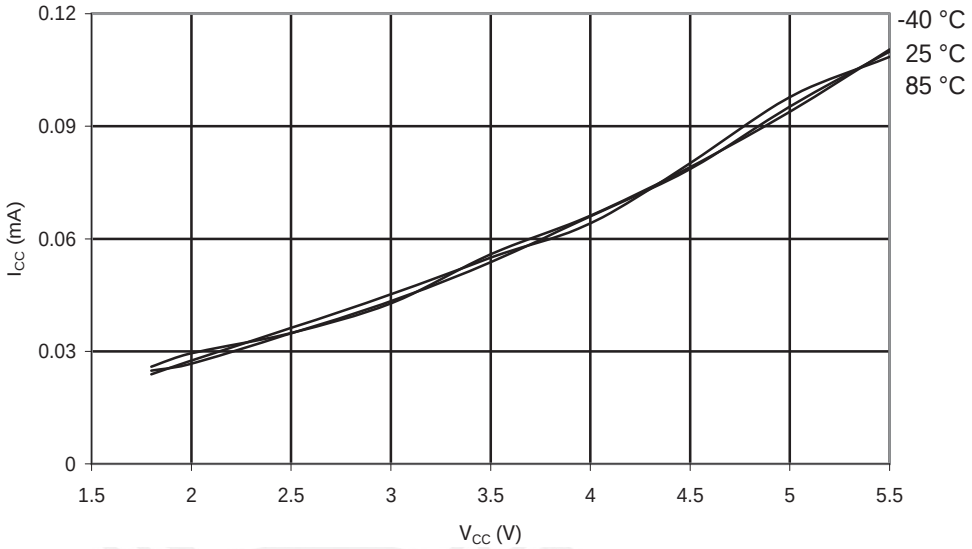


Figure 29-50. ATmega88PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

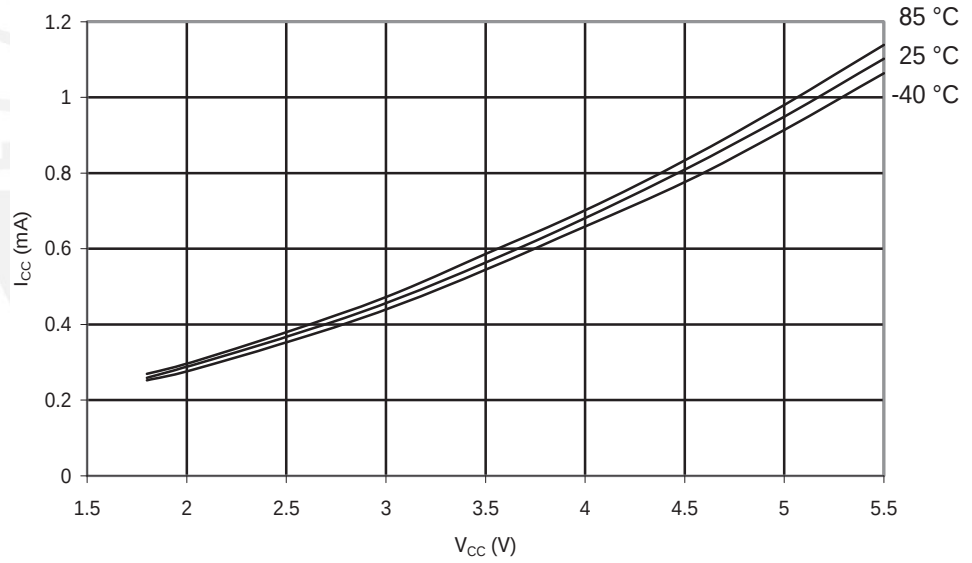
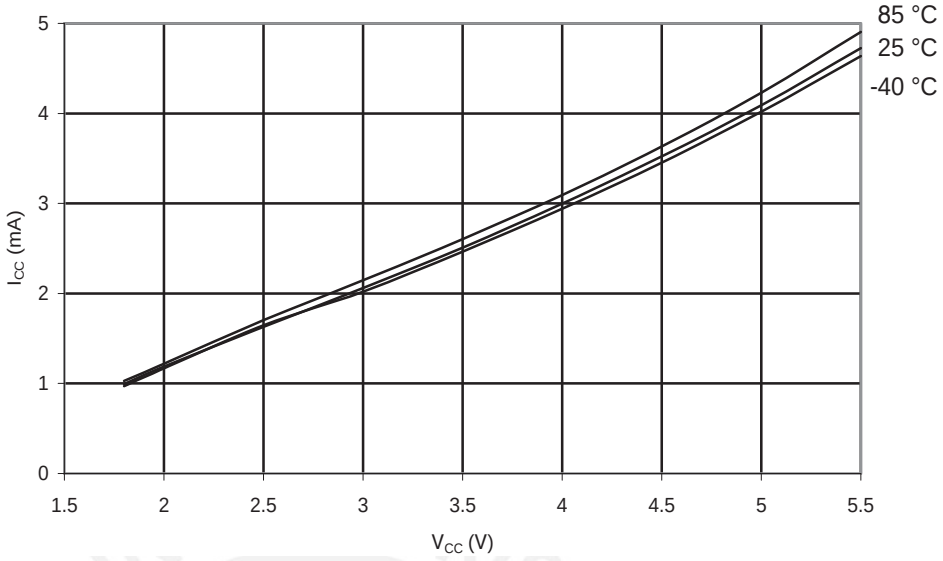


Figure 29-51. ATmega88PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.2.2 Idle Supply Current

Figure 29-52. ATmega88PA: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

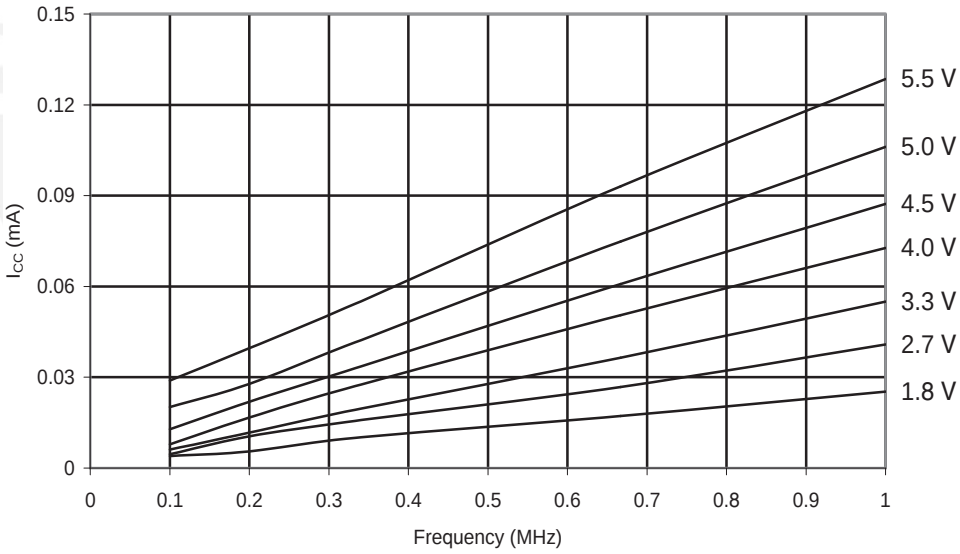


Figure 29-53. ATmega88PA: Idle Supply Current vs. Frequency (1 - 20 MHz)

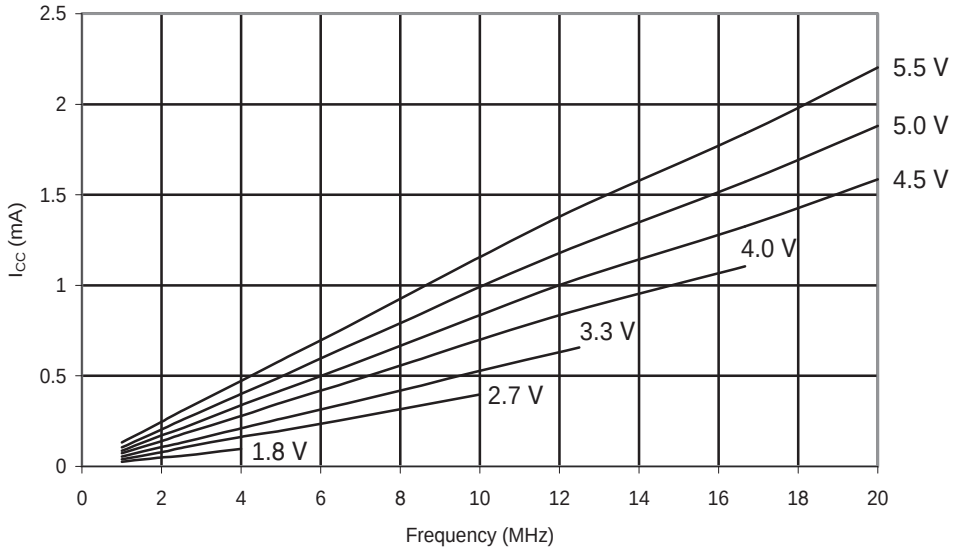


Figure 29-54. ATmega88PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

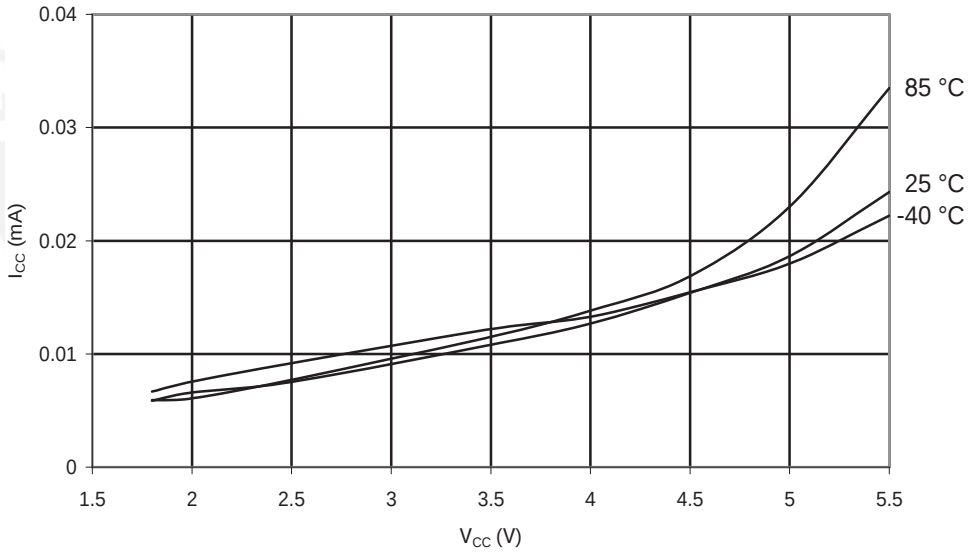


Figure 29-55. ATmega88PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

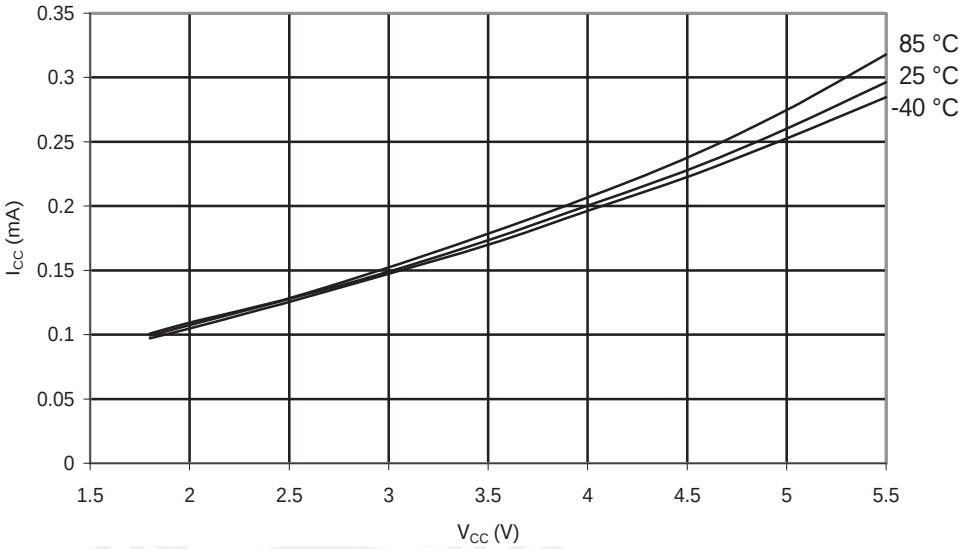
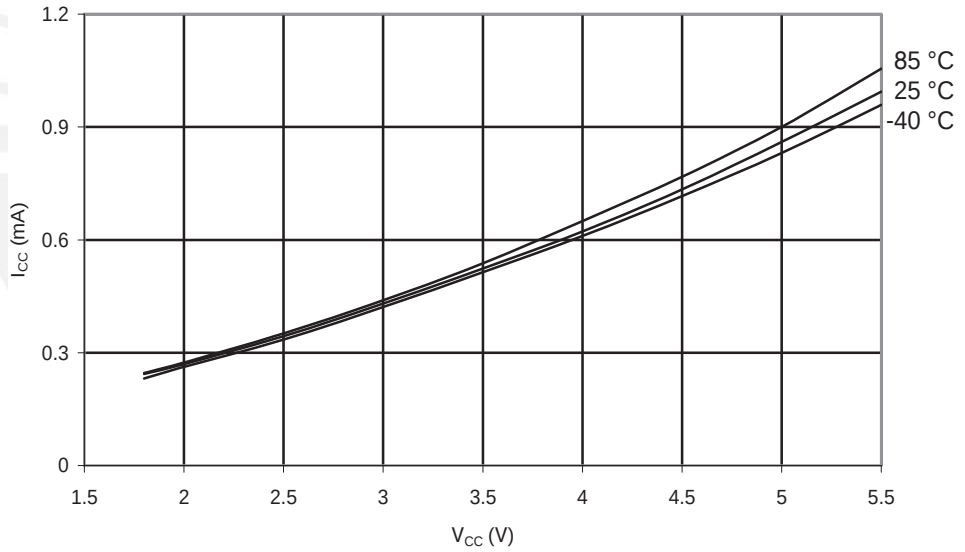


Figure 29-56. ATmega88PA: Idle Supply Current vs. Vcc (Internal RC Oscillator, 8 MHz)



29.2.3 ATmega88PA: **Supply Current of IO Modules**

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See "Power Reduction Register" on page 42 for details.

Table 29-3. ATmega88PA: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V_{CC} = 2V, F = 1 MHz	V_{CC} = 3V, F = 4 MHz	V_{CC} = 5V, F = 8 MHz
PRUSART0	3.0 uA	21.3 uA	97.9 uA
PRTWI	6.1 uA	45.4 uA	219.0 uA
PRTIM2	5.2 uA	35.2 uA	149.5 uA
PRTIM1	3.8 uA	25.6 uA	110.0 uA
PRTIM0	1.5 uA	9.8 uA	39.6 uA
PRSPI	5.2 uA	40.0 uA	199.6 uA
PRADC	6.3 uA	48.7 uA	247.0 uA

Table 29-4. ATmega88PA: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-47 on page 351 and Figure 29-48 on page 351)	Additional Current consumption compared to Idle with external clock (see Figure 29-52 on page 353 and Figure 29-53 on page 354)
PRUSART0	1.8%	11.4%
PRTWI	3.9%	24.4%
PRTIM2	2.9%	18.6%
PRTIM1	2.1%	13.6%
PRTIM0	0.8%	5.2%
PRSPI	3.5%	21.5%
PRADC	4.2%	26.3%

It is possible to calculate the typical current consumption based on the numbers from Table 29-4 on page 356 for other V_{CC} and frequency settings than listed in Table 29-3 on page 356.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-4 on page 356, third column, we see that we need to add 13.6% for the TIMER1, 26.3% for the ADC, and 21.5% for the SPI module. Reading from **Figure 29-52 on page 353**, we find that the idle current consumption is ~0.027 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CCtotal} \approx 0.027\ mA \cdot (1 + 0.136 + 0.263 + 0.215) \approx 0.043\ mA$$

29.2.4 Power-down Supply Current

Figure 29-57. ATmega88PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

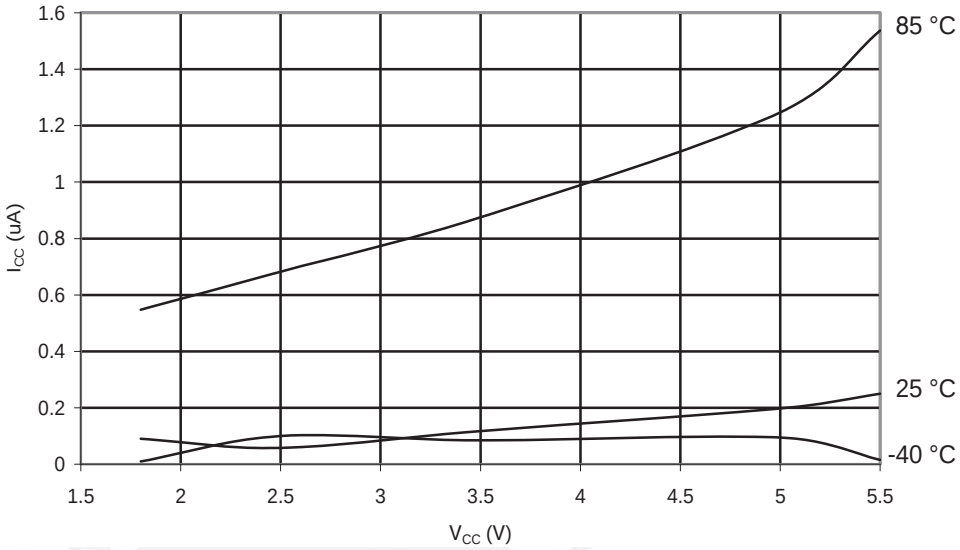
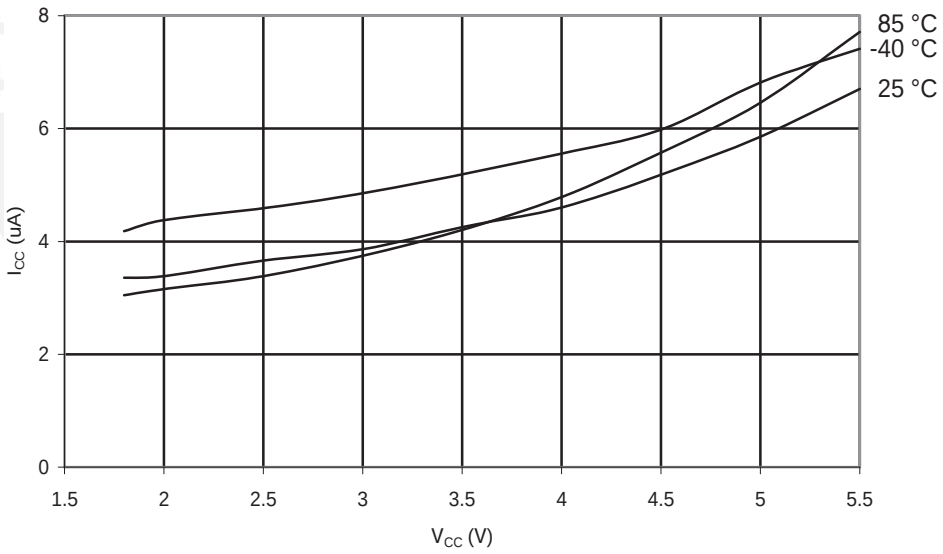
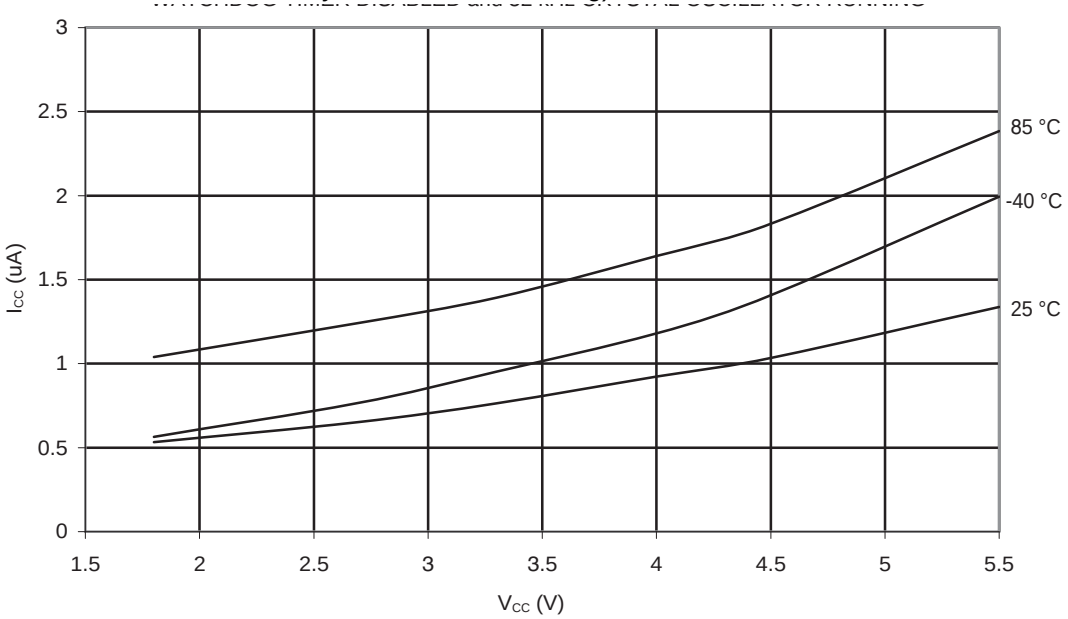


Figure 29-58. ATmega88PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



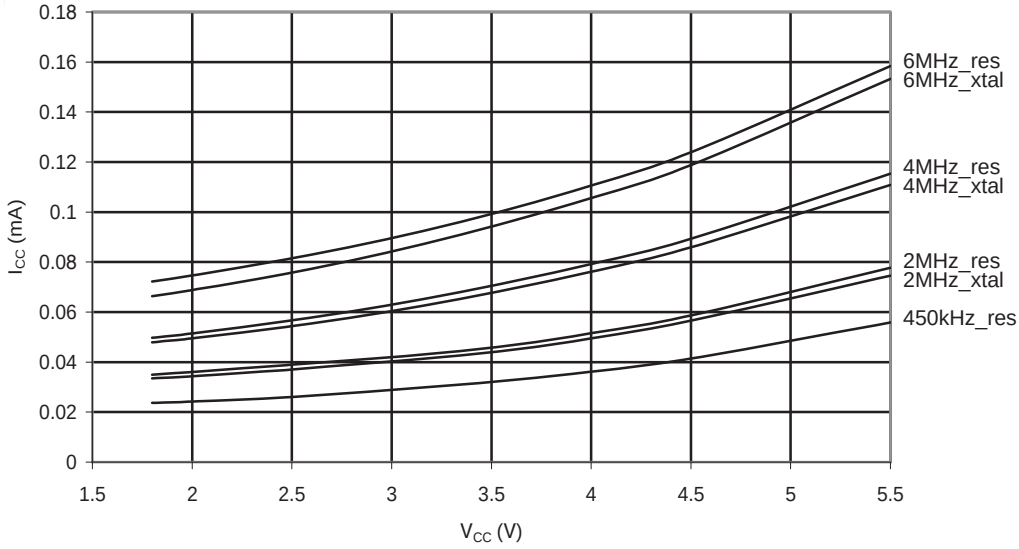
29.2.5 Power-save Supply Current

Figure 29-59. ATmega88PA: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.2.6 Standby Supply Current

Figure 29-60. ATmega88PA: Standby Supply Current vs. Vcc (Watchdog Timer Disabled)



29.2.7 Pin Pull-Up

Figure 29-61. ATmega88PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 1.8 V)

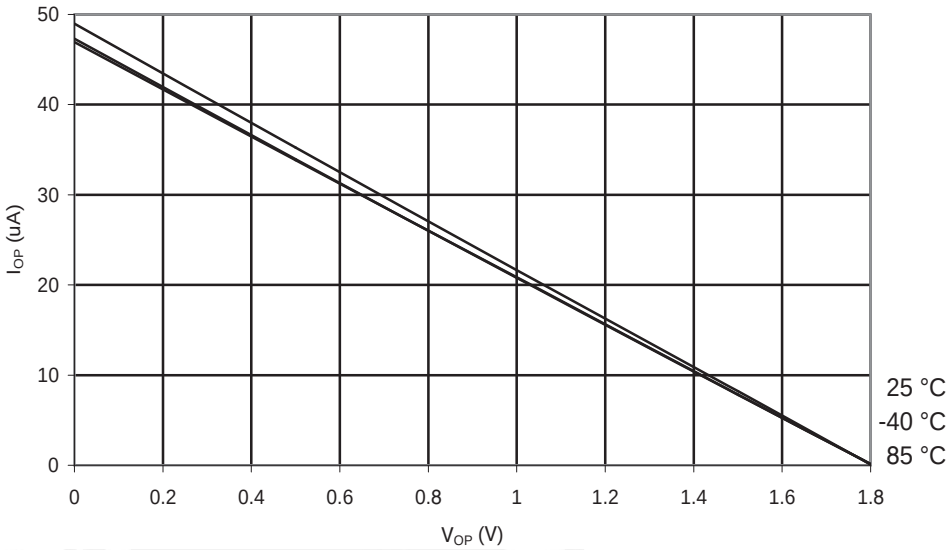


Figure 29-62. ATmega88PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 2.7 V)

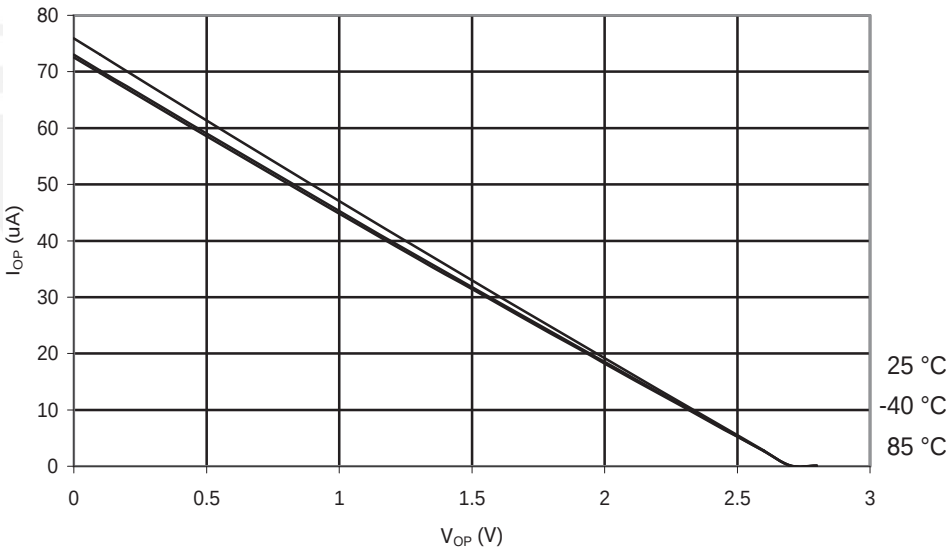


Figure 29-63. ATmega88PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 5 V)

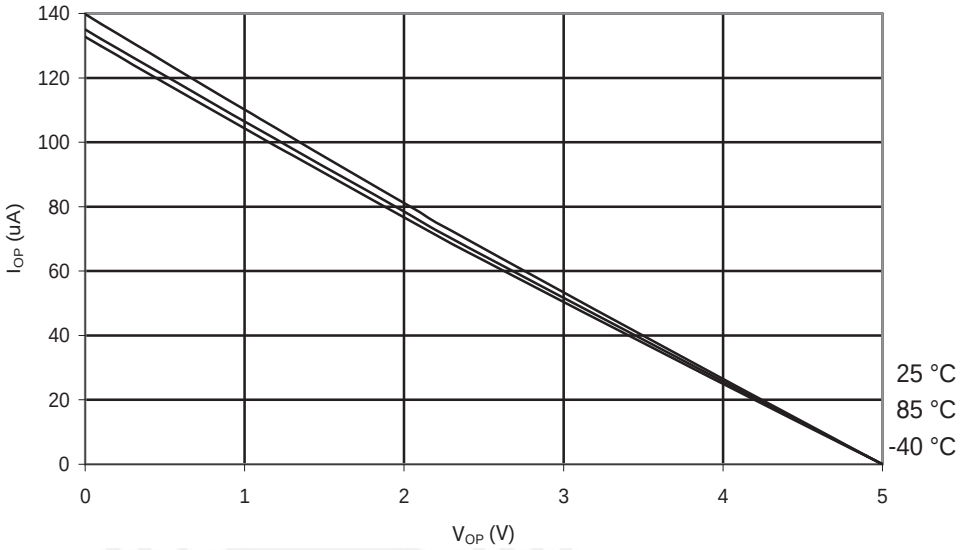


Figure 29-64. ATmega88PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 1.8 V)

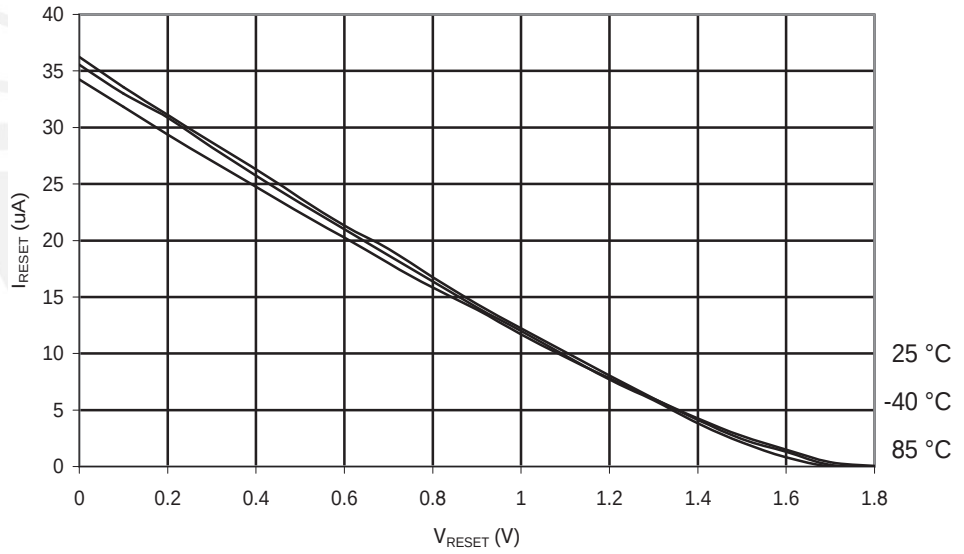


Figure 29-65. ATmega88PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 2.7 V)

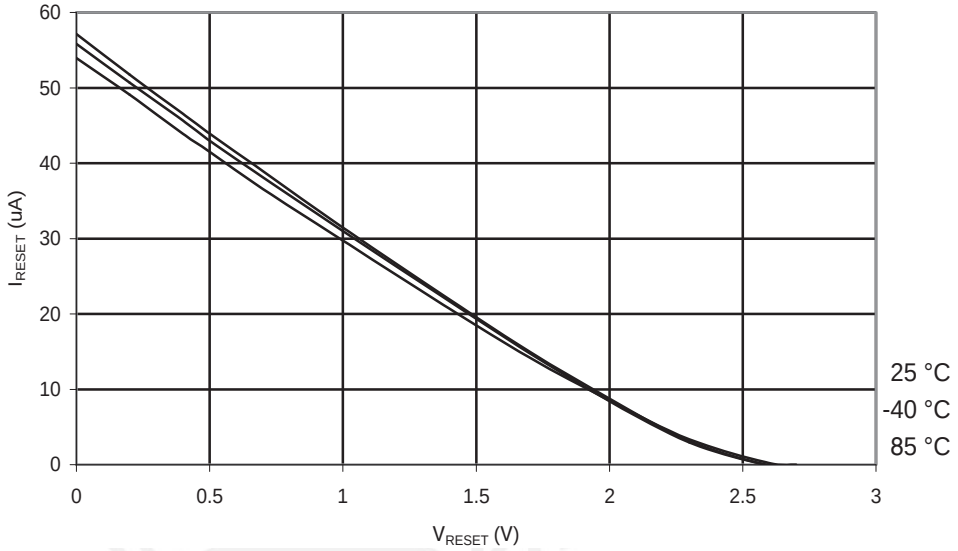
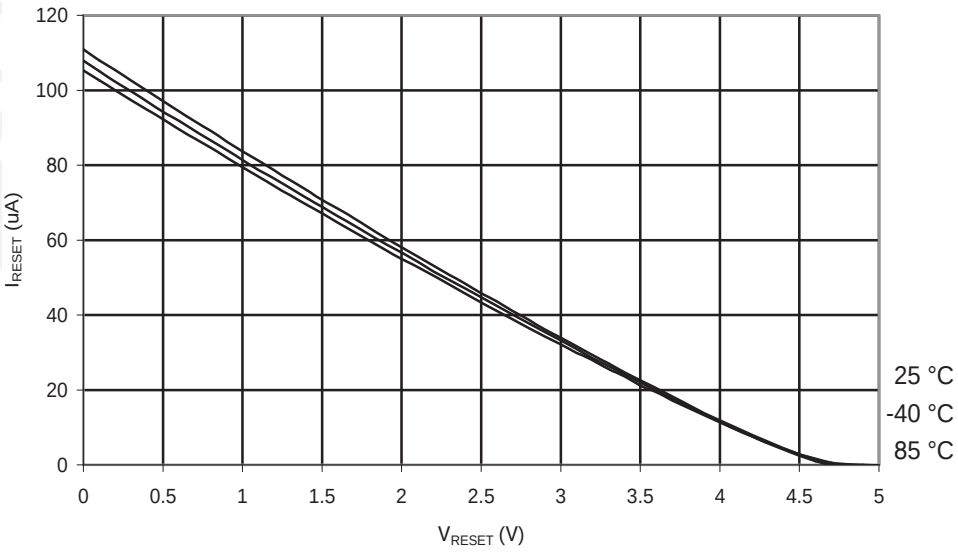


Figure 29-66. ATmega88PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 5 V)



29.2.8 Pin Driver Strength

Figure 29-67. ATmega88PA: I/O Pin Output Voltage vs. Sink Current (V_{CC} = 3 V)

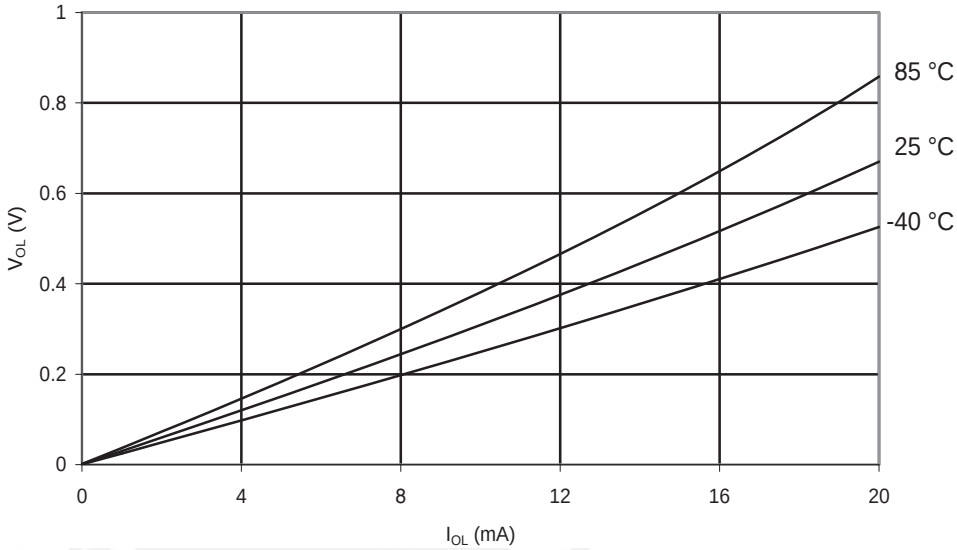


Figure 29-68. ATmega88PA: I/O Pin Output Voltage vs. Sink Current (V_{CC} = 5 V)

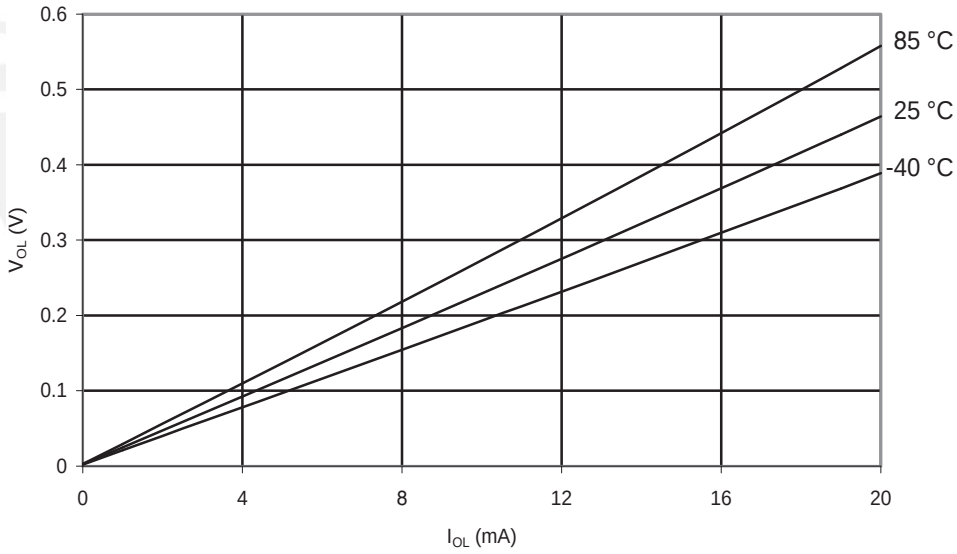


Figure 29-69. ATmega88PA: I/O Pin Output Voltage vs. Source Current (Vcc = 3 V)

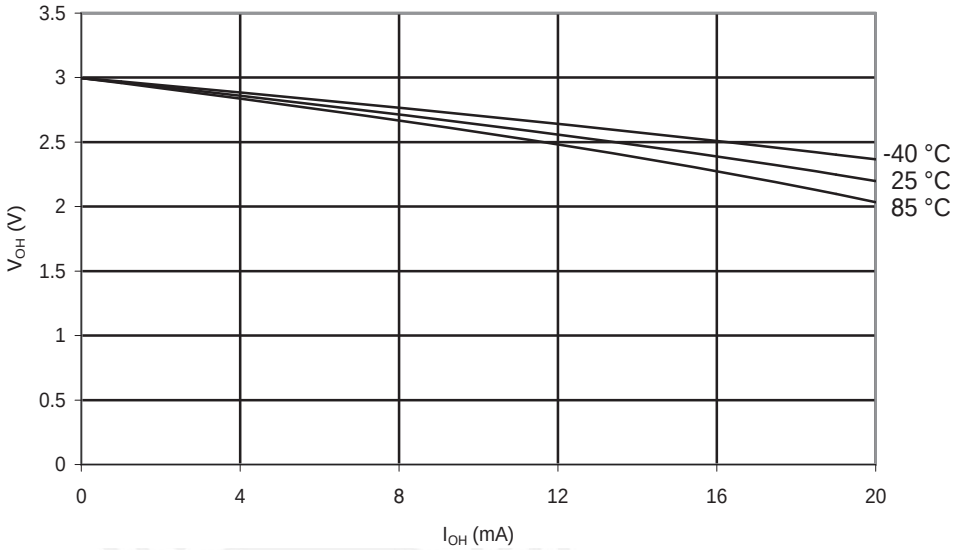
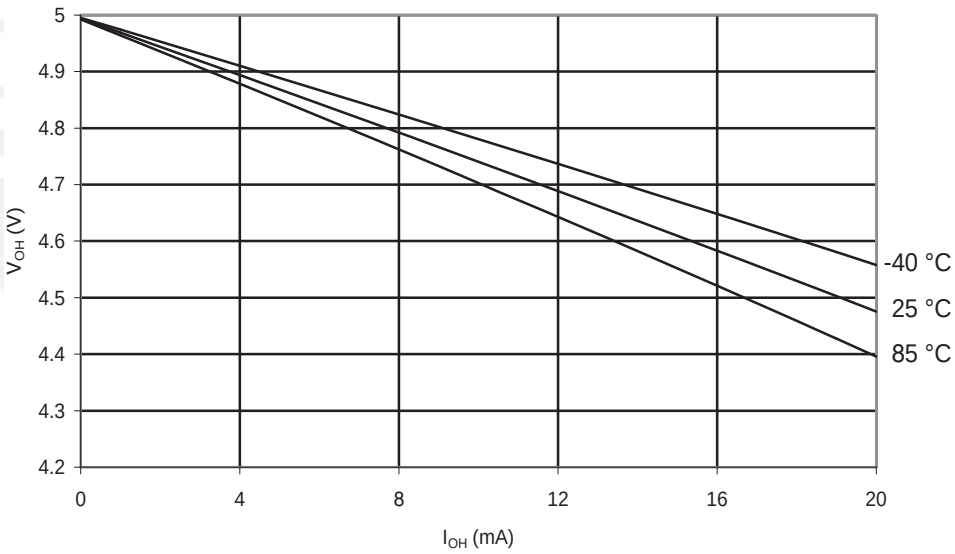


Figure 29-70. ATmega88PA: I/O Pin Output Voltage vs. Source Current (V_{CC} = 5 V)



29.2.9 Pin Threshold and Hysteresis

Figure 29-71. ATmega88PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

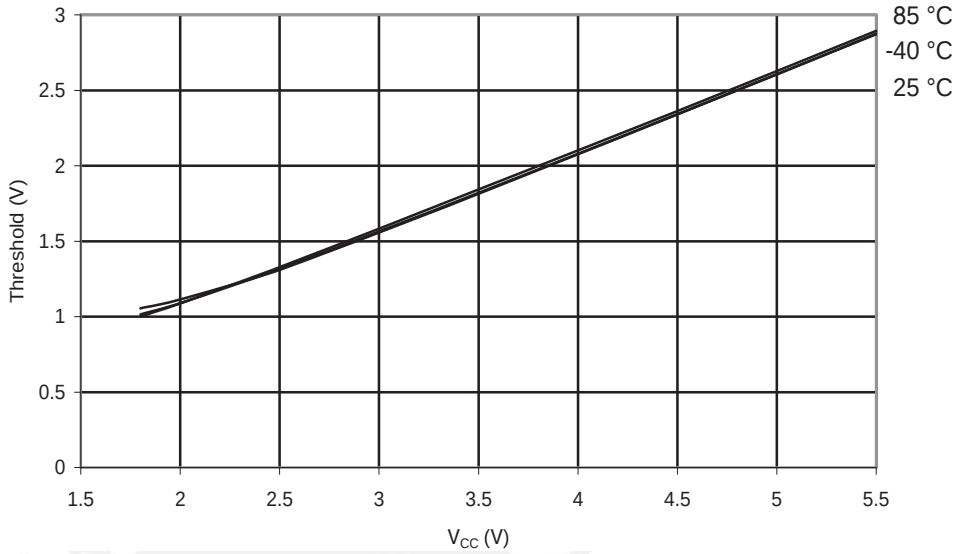


Figure 29-72. ATmega88PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

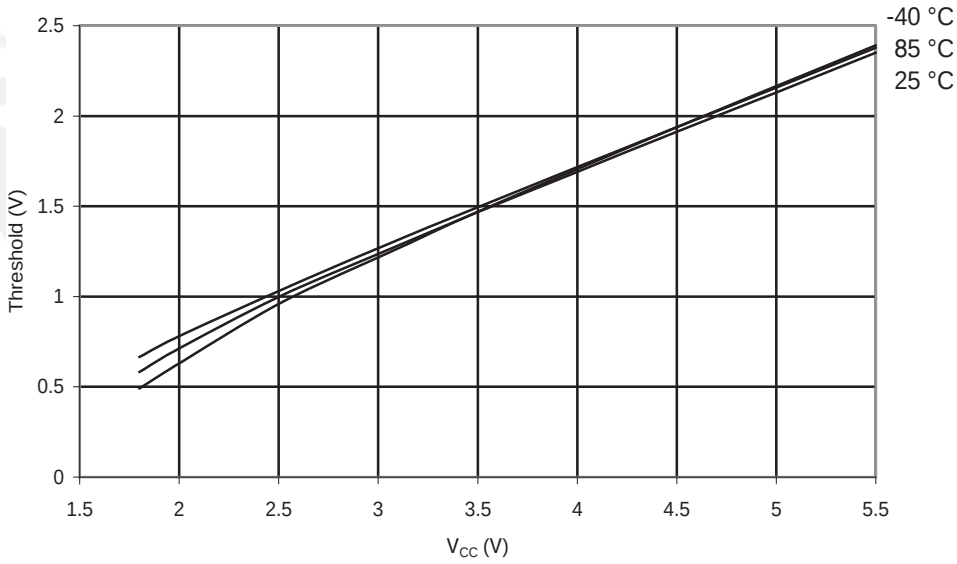


Figure 29-73. ATmega88PA: I/O Pin Input Hysteresis vs. V_{CC}

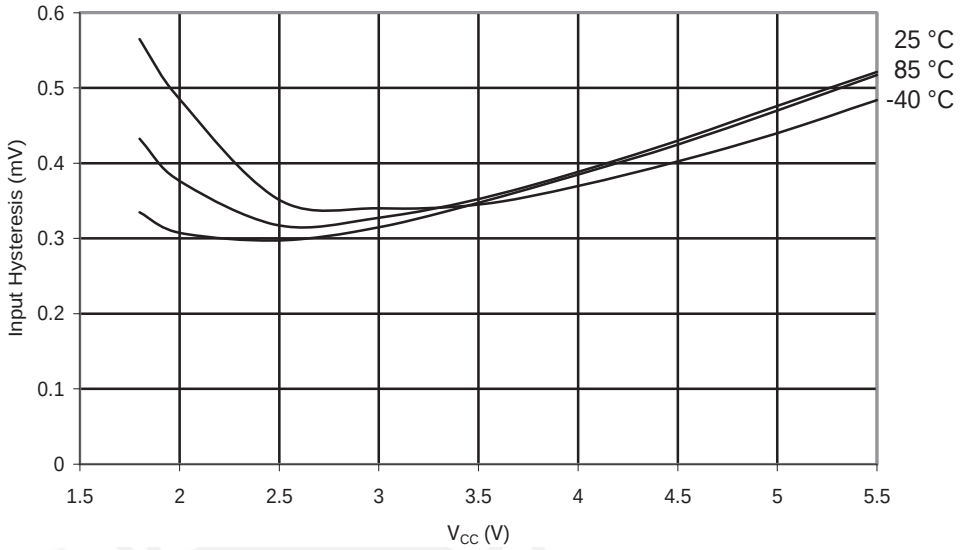


Figure 29-74. ATmega88PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

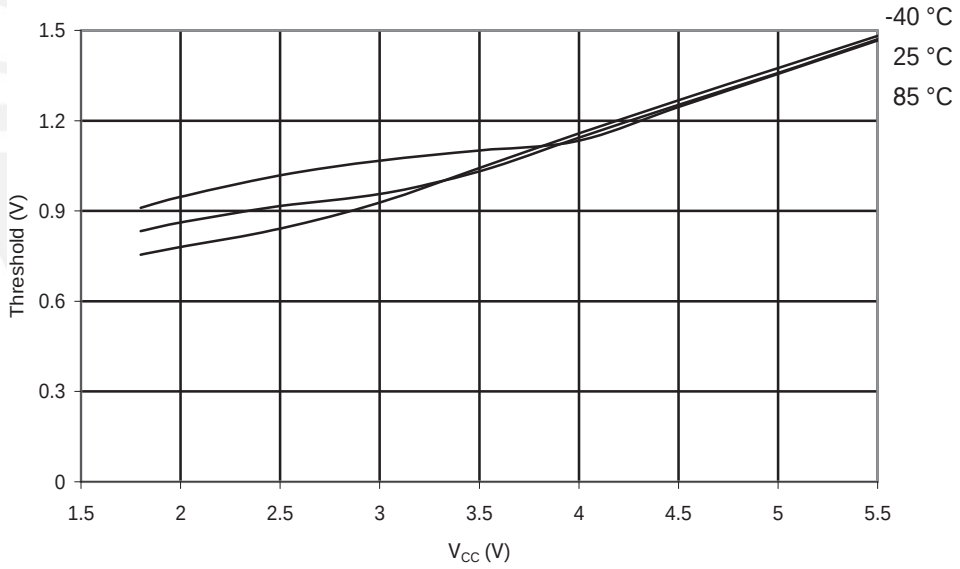


Figure 29-75. ATmega88PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

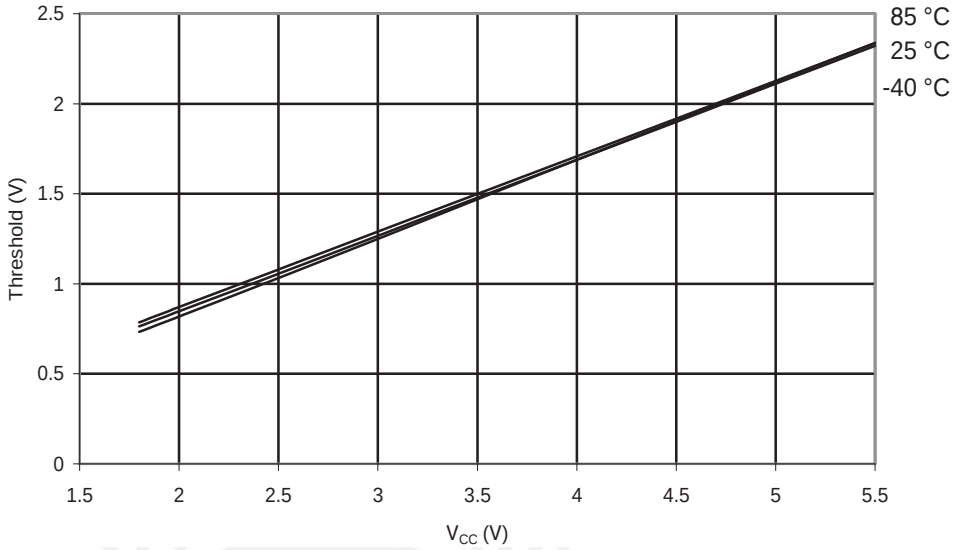
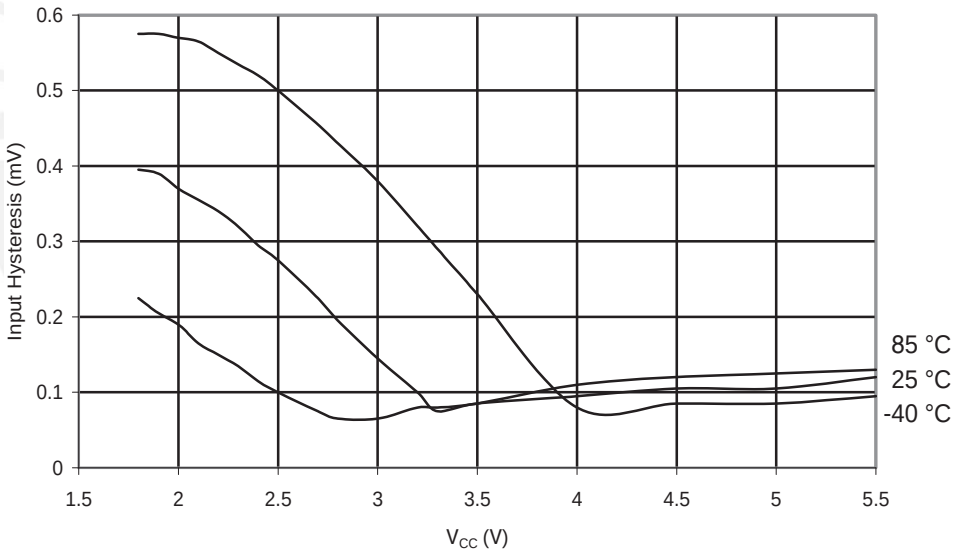


Figure 29-76. ATmega88PA: Reset Pin Input Hysteresis vs. V_{CC}



29.2.10 BOD Threshold

Figure 29-77. ATmega88PA: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

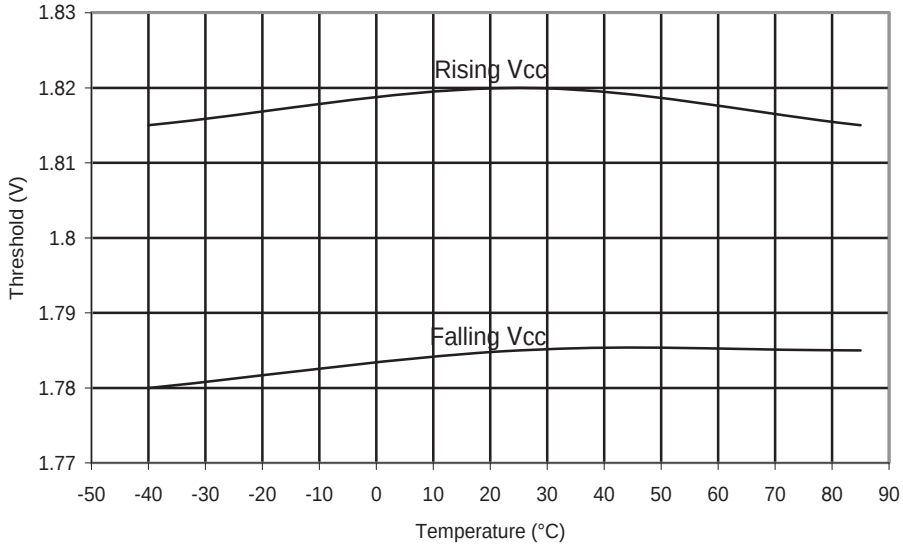


Figure 29-78. ATmega88PA: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

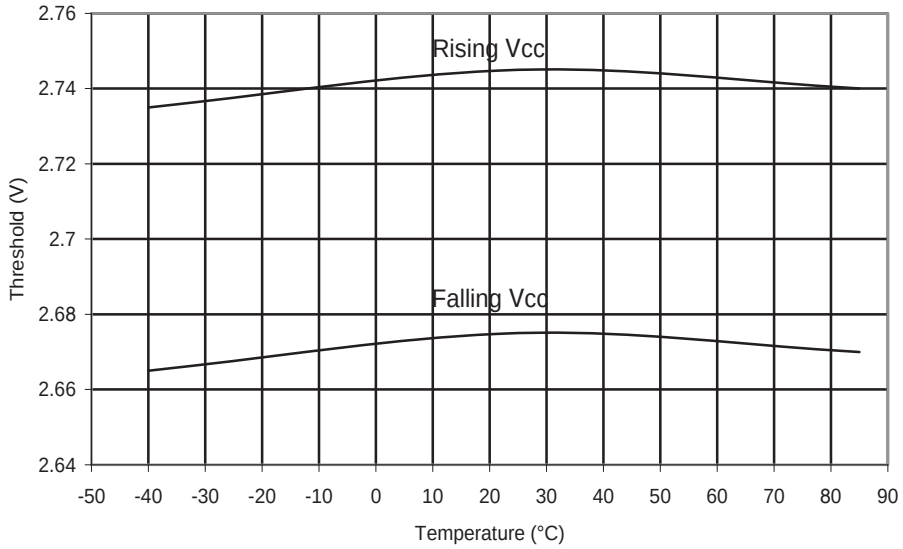
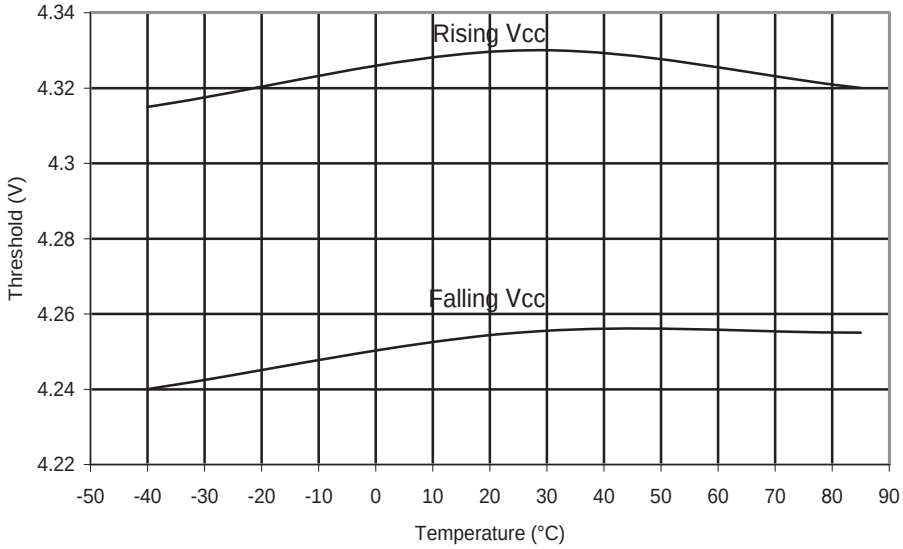


Figure 29-79. ATmega88PA: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.2.11 Internal Oscilllator Speed

Figure 29-80. ATmega88PA: Watchdog Oscillator Frequency vs. Temperature

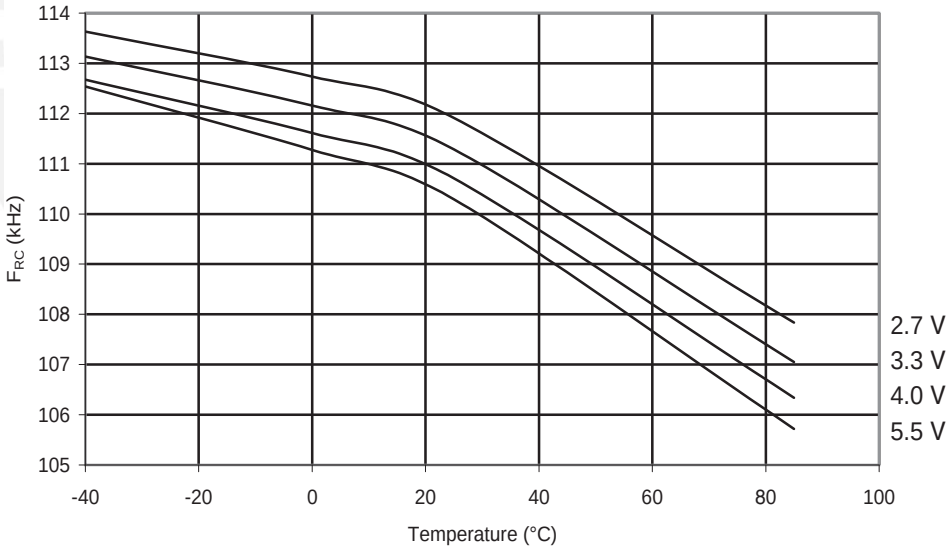


Figure 29-81. ATmega88PA: Watchdog Oscillator Frequency vs. V_{CC}

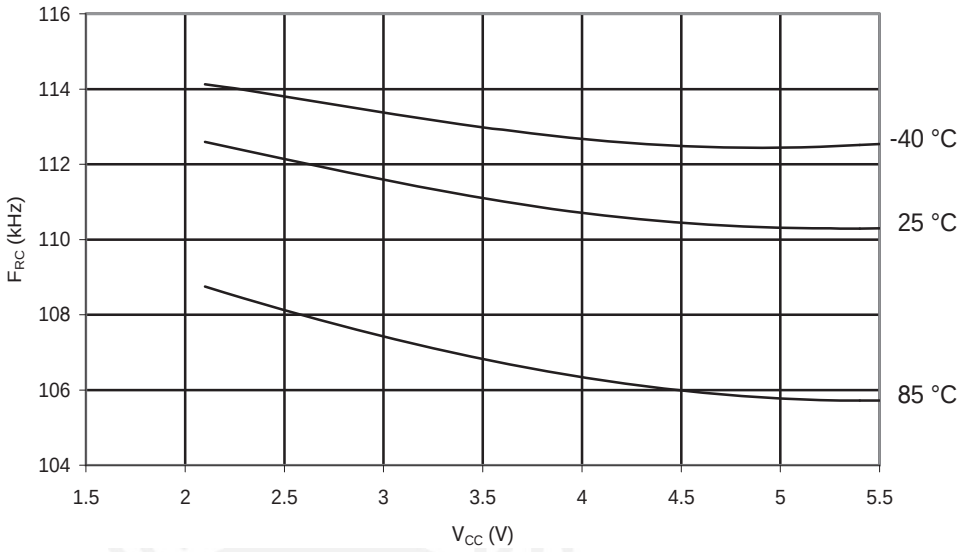


Figure 29-82. ATmega88PA: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

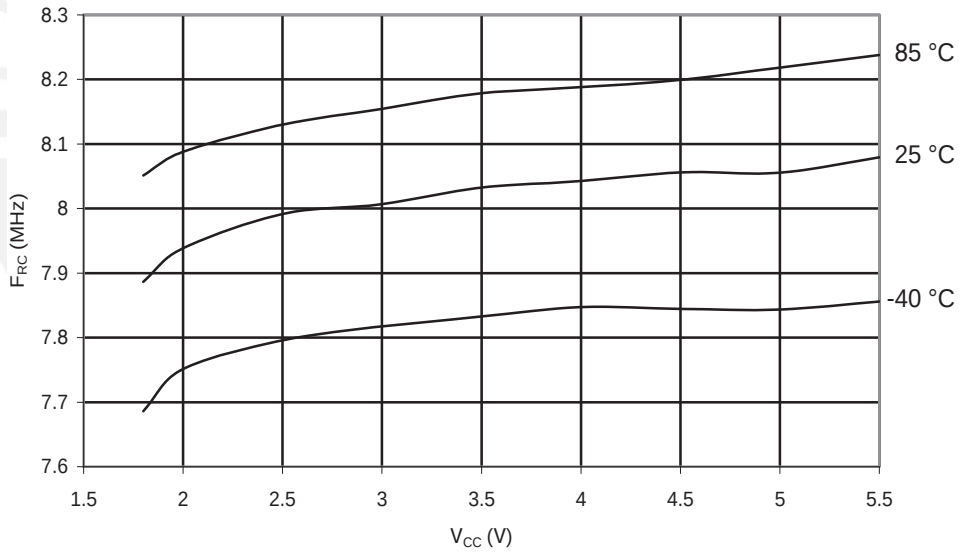


Figure 29-83. ATmega88PA: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

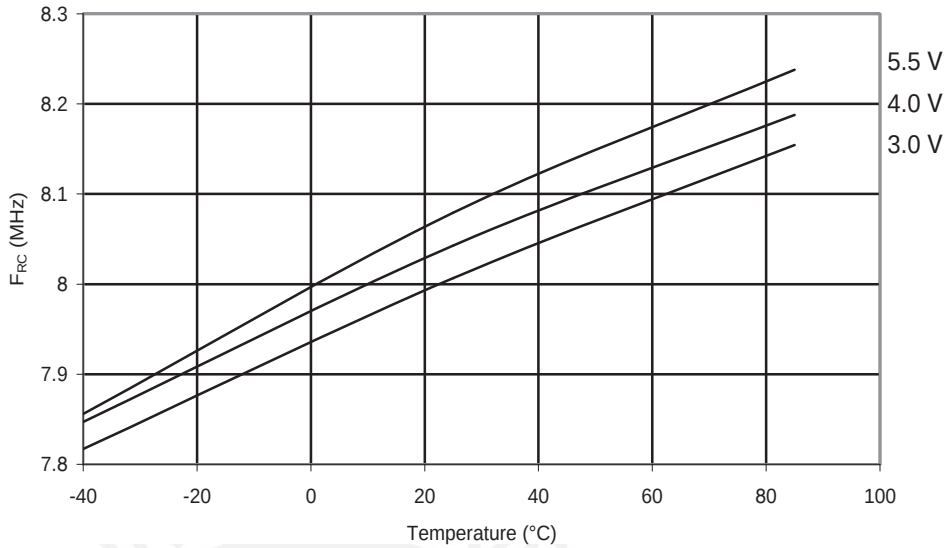
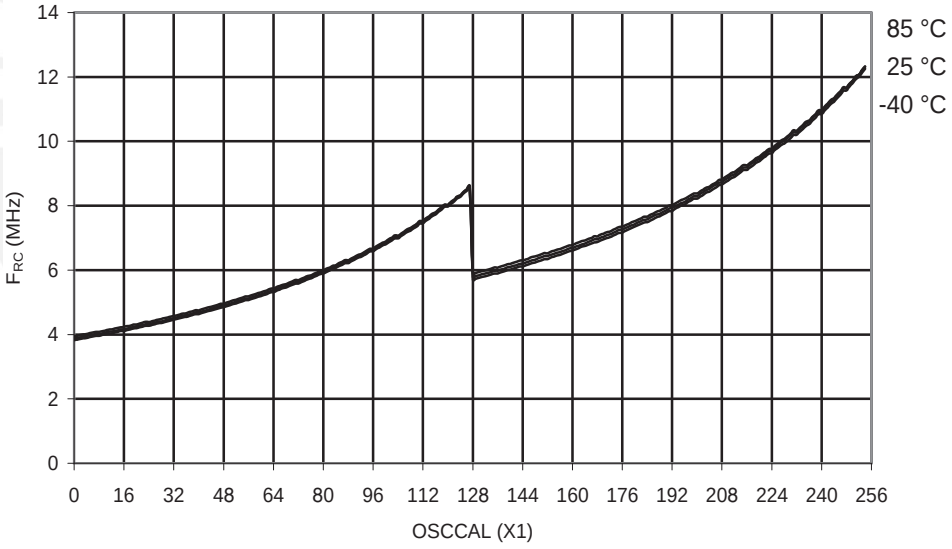


Figure 29-84. ATmega88PA: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.2.12 Current Consumption of Peripheral Units

Figure 29-85. ATmega88PA: ADC Current vs. V_{CC} ($AREF = AV_{CC}$)

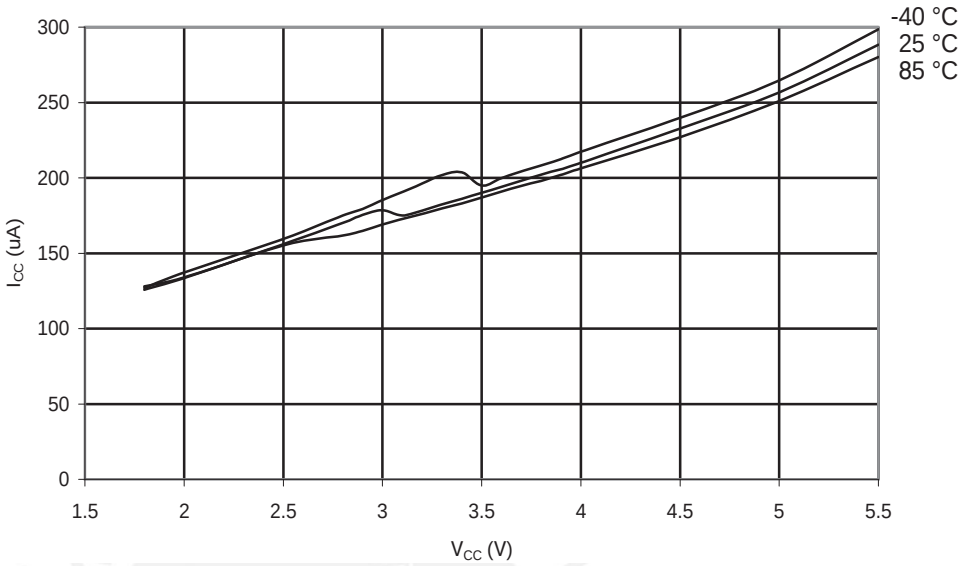


Figure 29-86. ATmega88PA: Analog Comparator Current vs. V_{CC}

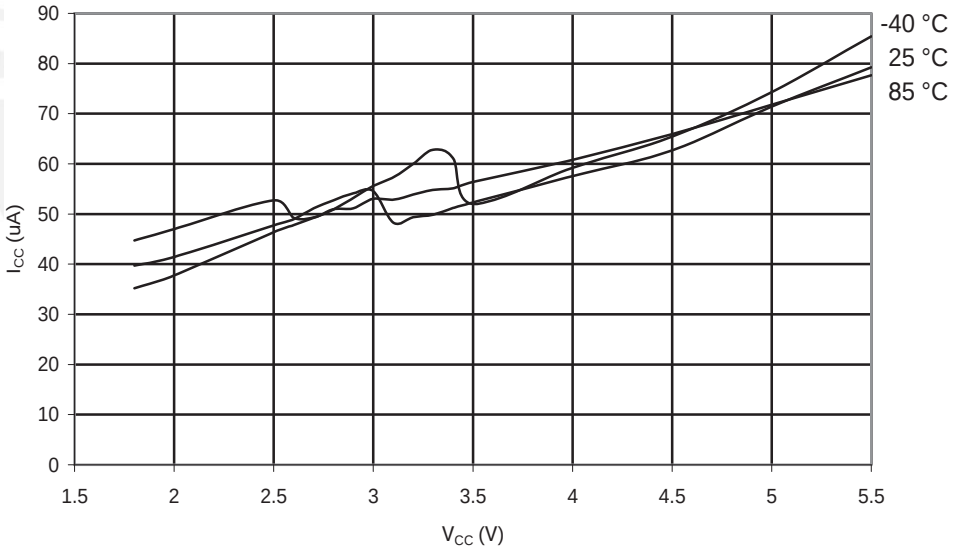


Figure 29-87. ATmega88PA: AREF External Reference Current vs. V_{CC}

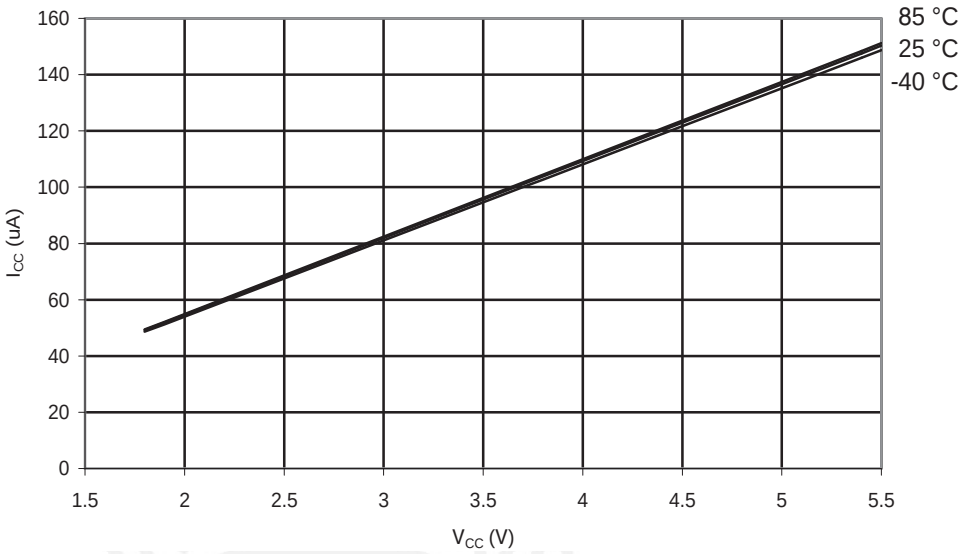


Figure 29-88. ATmega88PA: Brownout Detector Current vs. V_{CC}

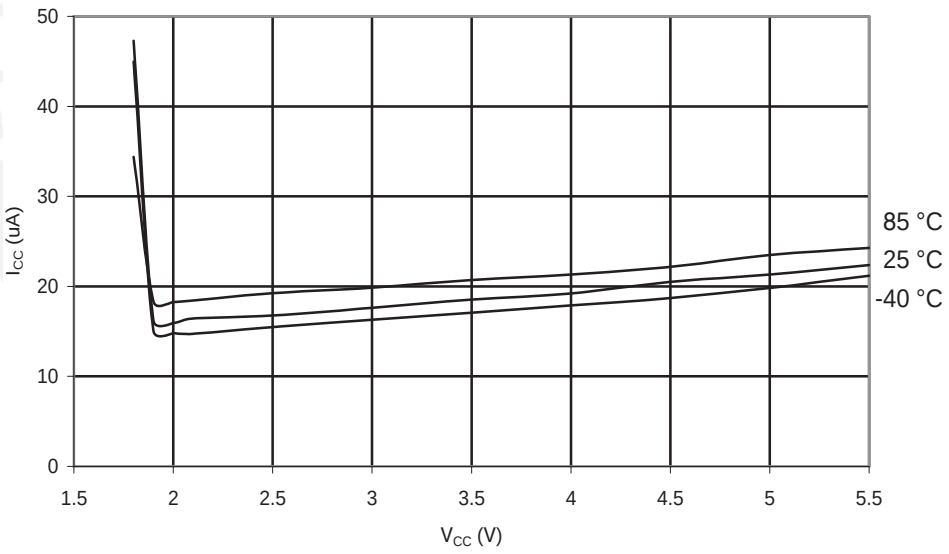
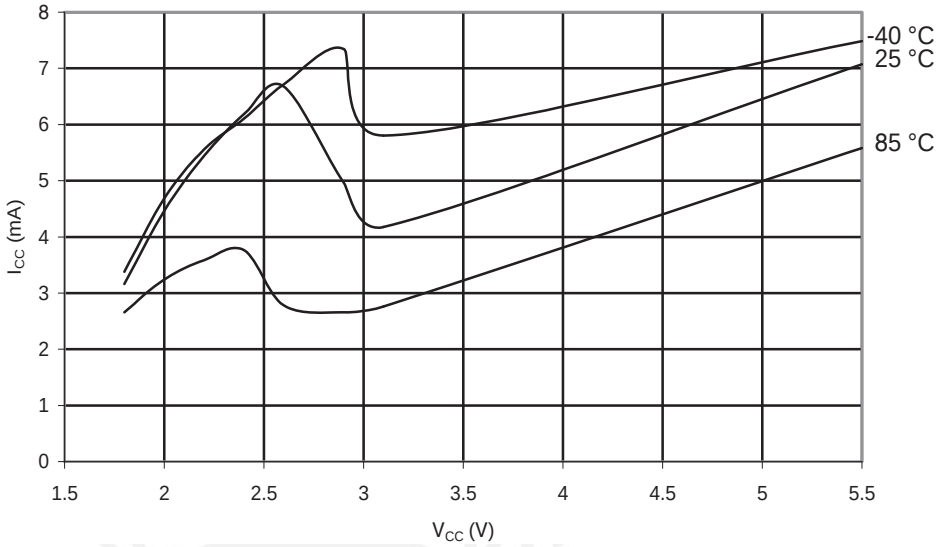


Figure 29-89. ATmega88PA: Programming Current vs. V_{CC}



29.2.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-90. ATmega88PA: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

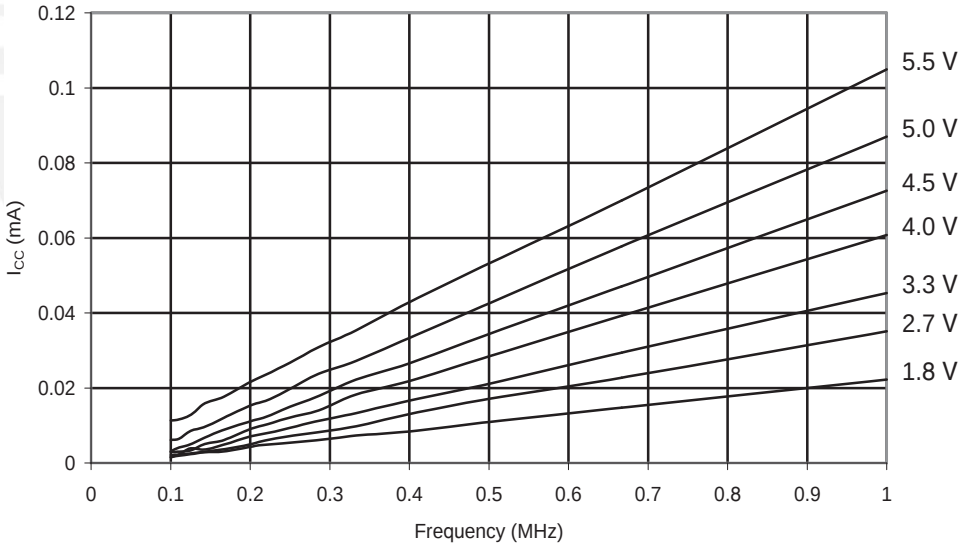


Figure 29-91. ATmega88PA: Reset Supply Current vs. Frequency (1 - 20 MHz)

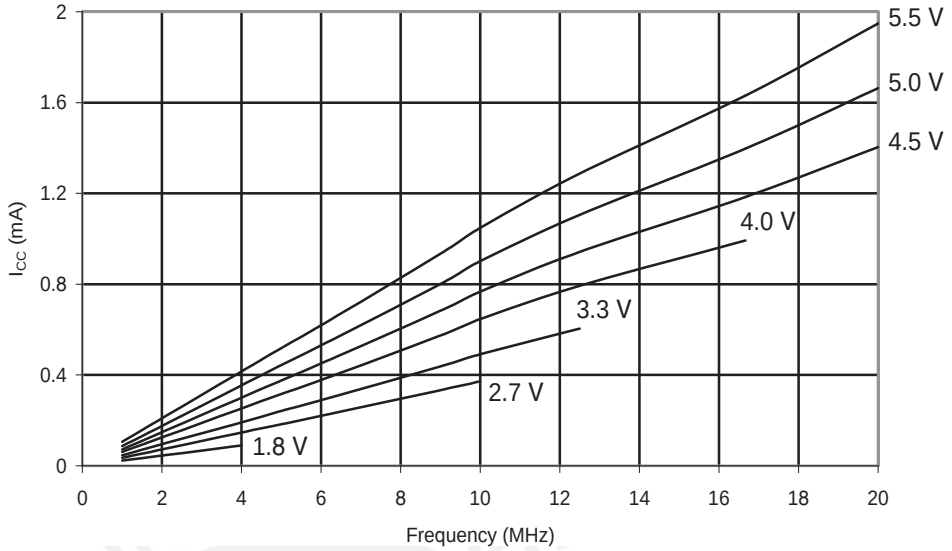
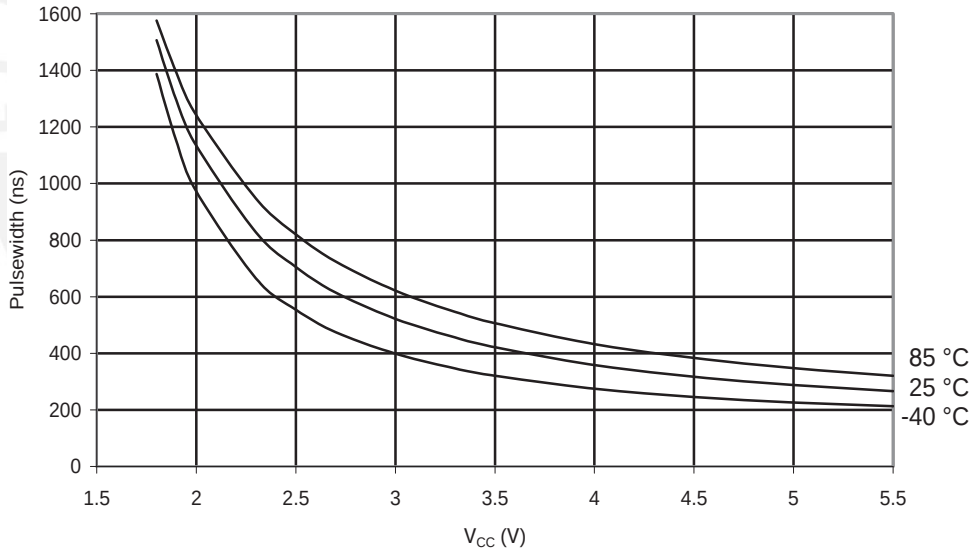


Figure 29-92. ATmega88PA: Minimum Reset Pulse width vs. V_{CC}



29.3 ATmega168PA Typical Characteristics

29.3.1 Active Supply Current

Figure 29-93. ATmega168PA: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

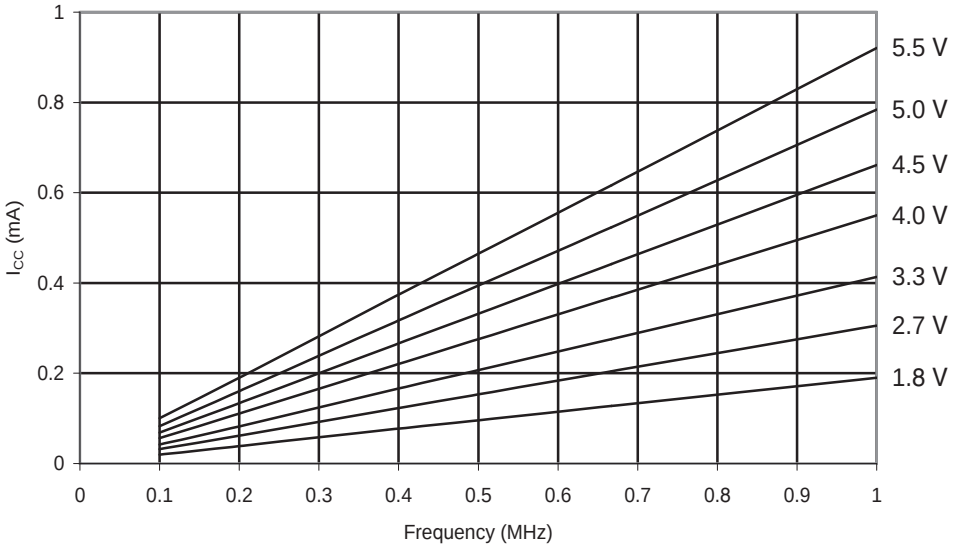


Figure 29-94. ATmega168PA: Active Supply Current vs. Frequency (1-20 MHz)

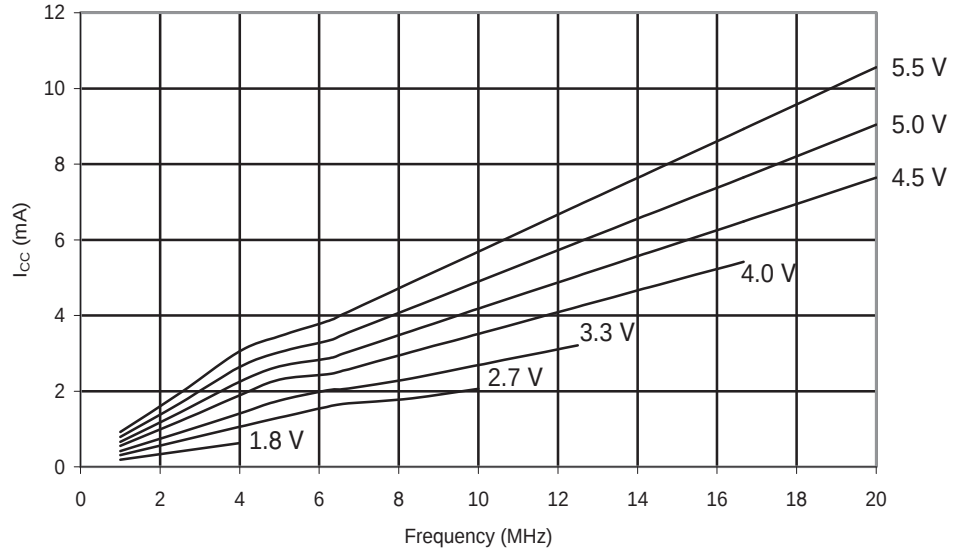


Figure 29-95. ATmega168PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

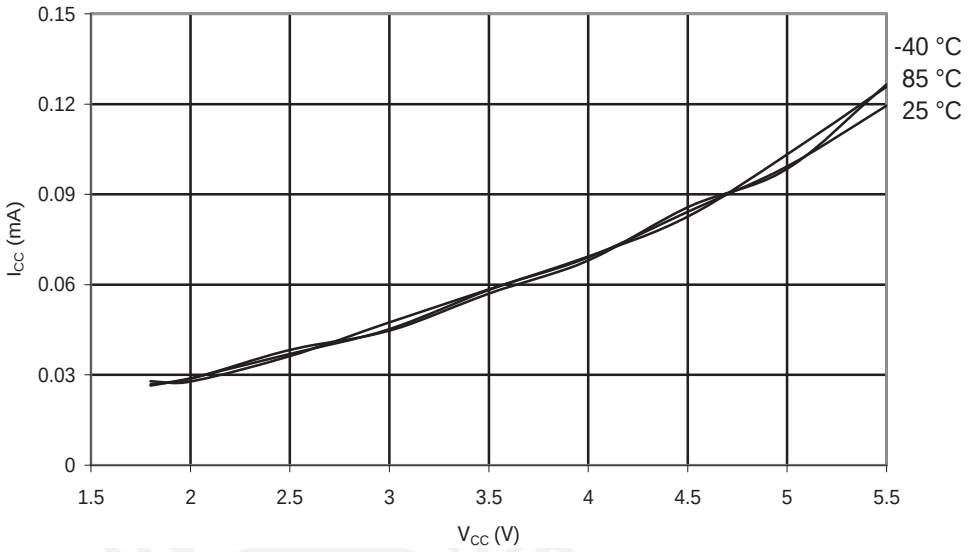


Figure 29-96. ATmega168PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

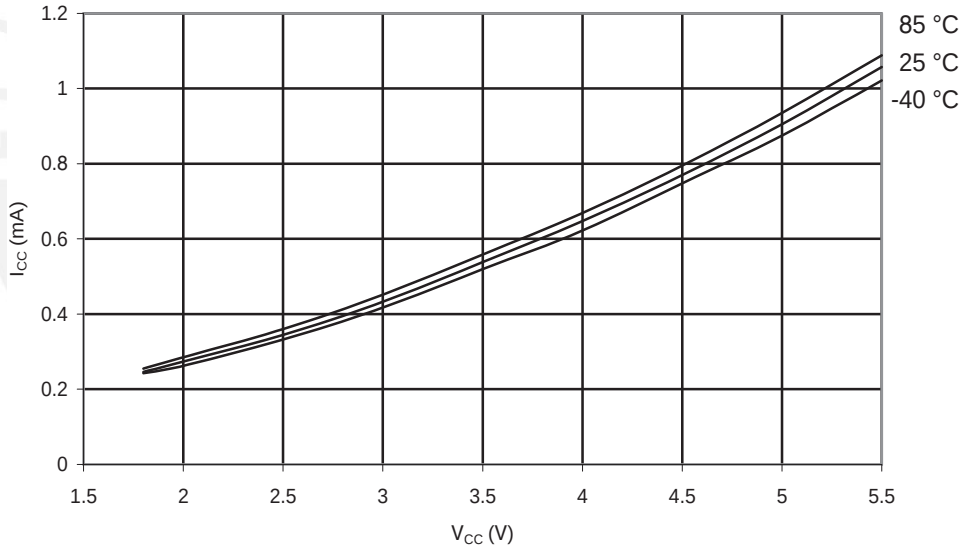
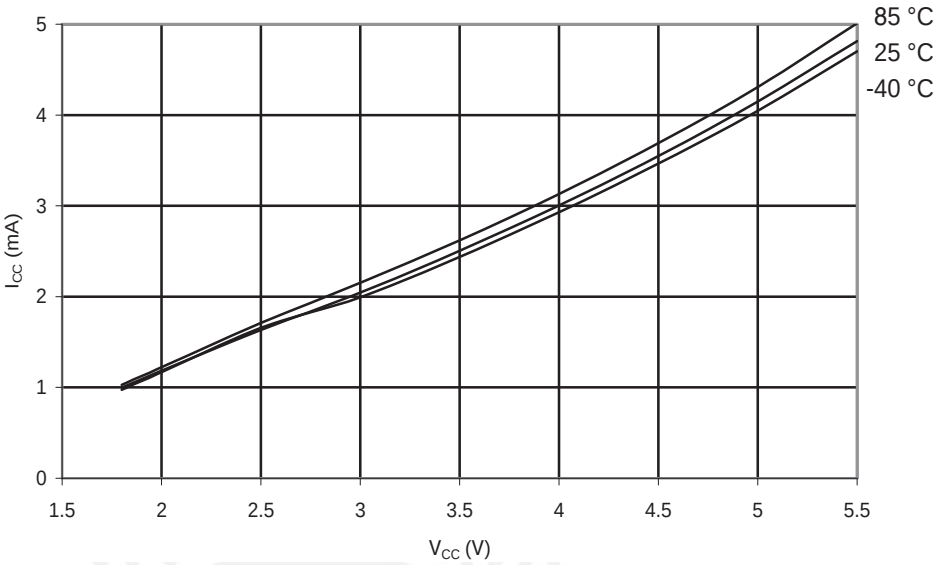


Figure 29-97. ATmega168PA: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.3.2 Idle Supply Current

Figure 29-98. ATmega168PA: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

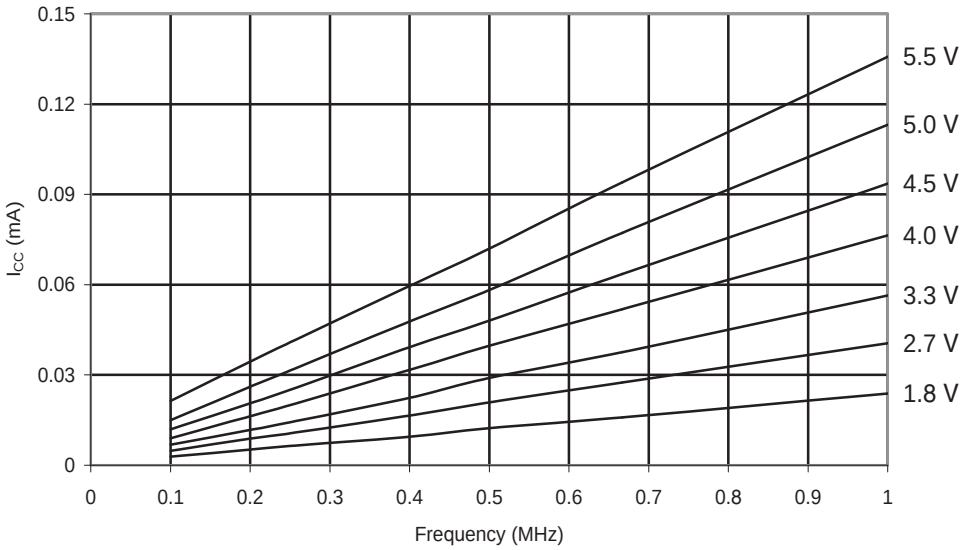


Figure 29-99. ATmega168PA: Idle Supply Current vs. Frequency (1-20 MHz)

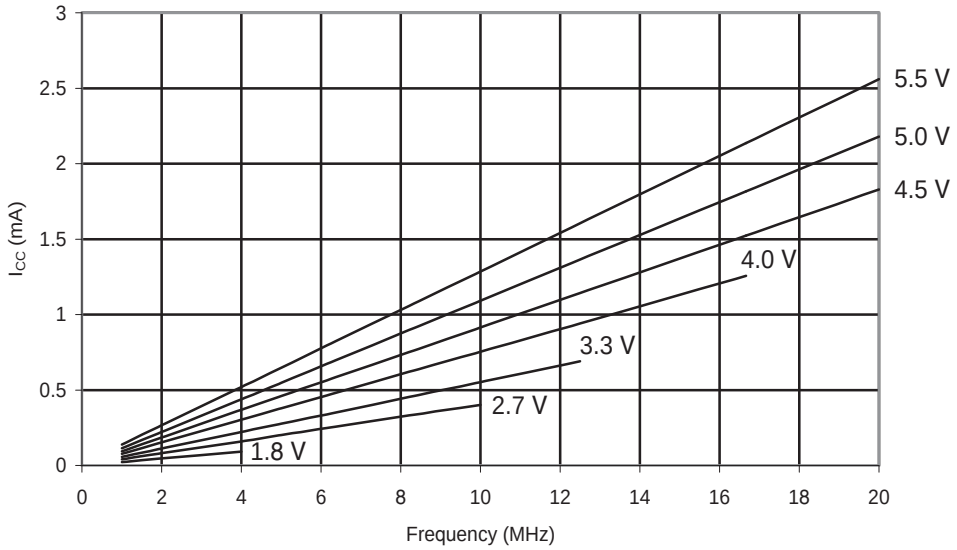


Figure 29-100.IATmega168PA: dle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

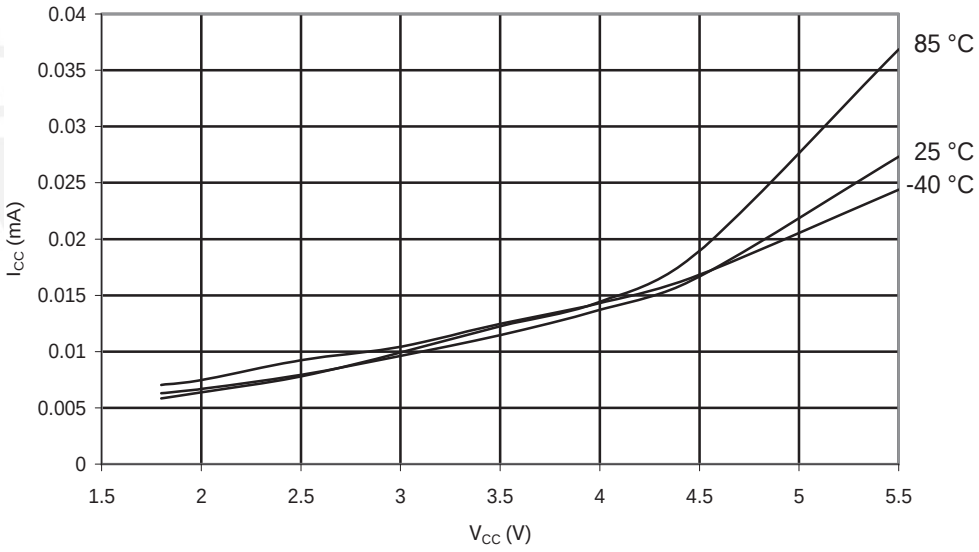


Figure 29-101. ATmega168PA: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

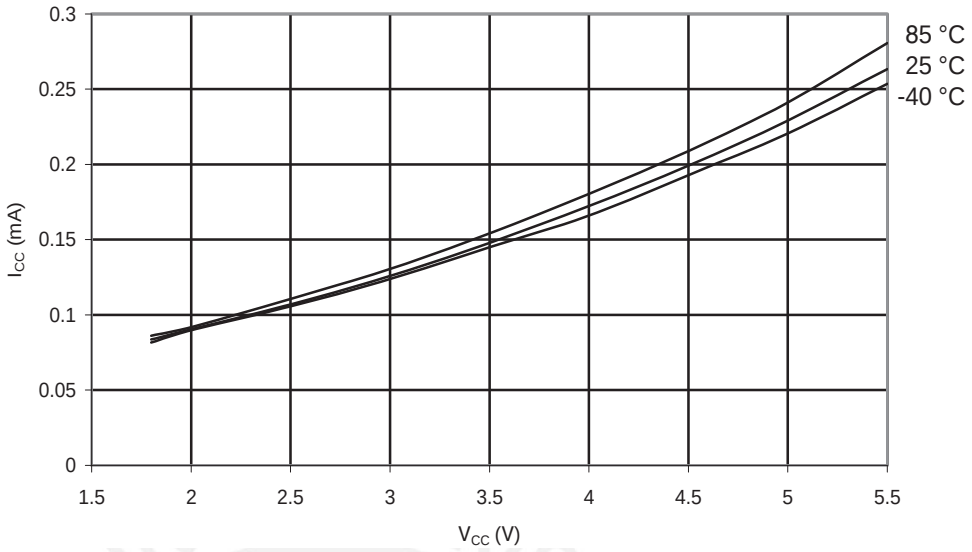
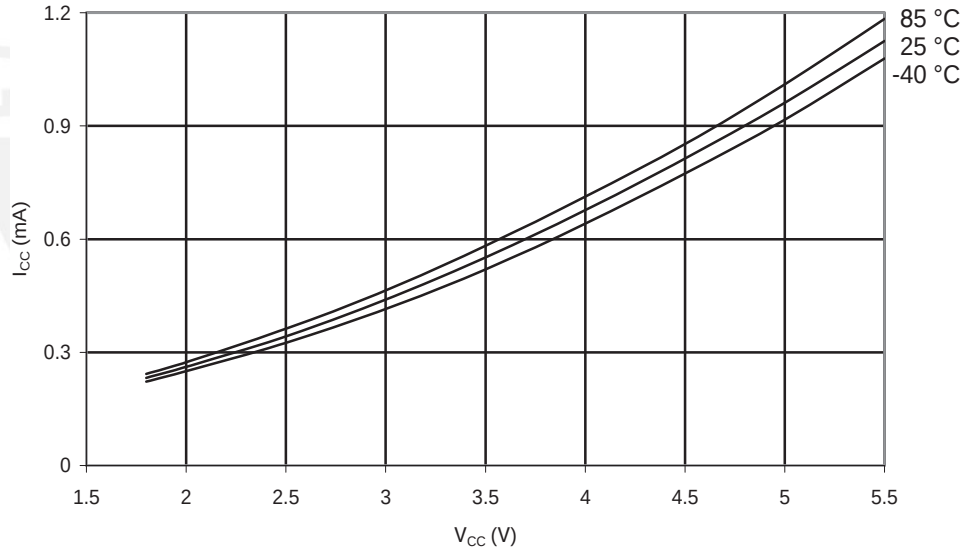


Figure 29-102. ATmega168PA: Idle Supply Current vs. Vcc (Internal RC Oscillator, 8 MHz)



29.3.3 ATmega168PA Supply Current of IO Modules

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See ”Power Reduction Register” on page 42 for details.

Table 29-5. ATmega168PA: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V_{CC} = 2V, F = 1 MHz	V_{CC} = 3V, F = 4 MHz	V_{CC} = 5V, F = 8 MHz
PRUSART0	2.86 uA	20.3 uA	52.2 uA
PRTWI	6.00 uA	44.1uA	122.0 uA
PRTIM2	4.97 uA	33.2 uA	79.8 uA
PRTIM1	3.50 uA	23.0 uA	55.3 uA
PRTIM0	1.43 uA	9.2 uA	21.4 uA
PRSPI	5.01 uA	38.6 uA	111.4 uA
PRADC	6.34 uA	45.7 uA	123.6 uA

Table 29-6. ATmega168PA: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-93 on page 375 and Figure 29-94 on page 375)	Additional Current consumption compared to Idle with external clock (see Figure 29-98 on page 377 and Figure 29-99 on page 378)
PRUSART0	1.5%	8.9%
PRTWI	3.2%	19.5%
PRTIM2	2.4%	14.8%
PRTIM1	1.7%	10.3%
PRTIM0	0.7%	4.1%
PRSPI	2.9%	17.1%
PRADC	3.4%	20.3%

It is possible to calculate the typical current consumption based on the numbers from Table 29-6 on page 380 for other V_{CC} and frequency settings than listed in Table 29-5 on page 380.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-6 on page 380, third column, we see that we need to add 10.3% for the TIMER1, 20.3% for the ADC, and 17.1% for the SPI module. Reading from Figure 29-98 on page 377, we find that the idle current consumption is ~0.027 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CCtotal} \approx 0.027\ mA \cdot (1 + 0.103 + 0.203 + 0.171) \approx 0.040\ mA$$

29.3.4 Power-down Supply Current

Figure 29-103. ATmega168PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

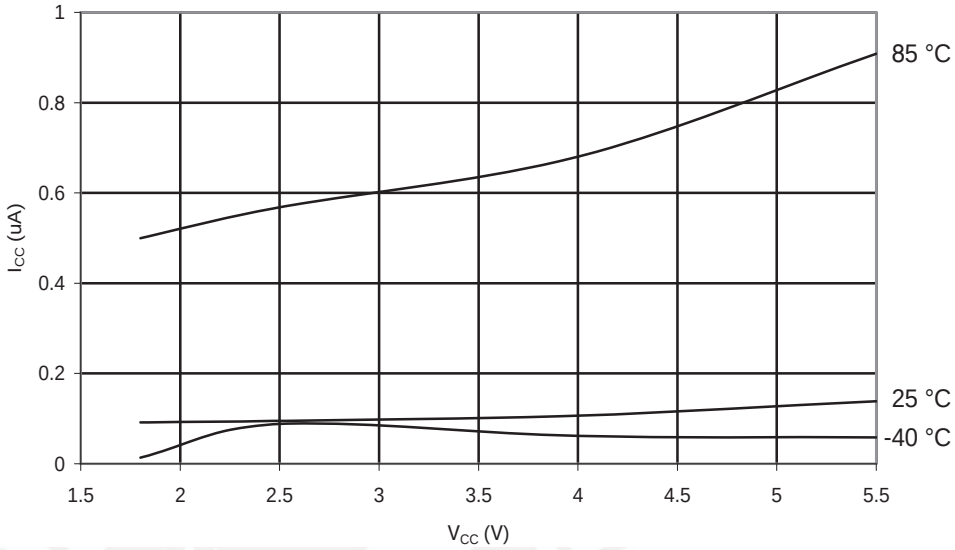
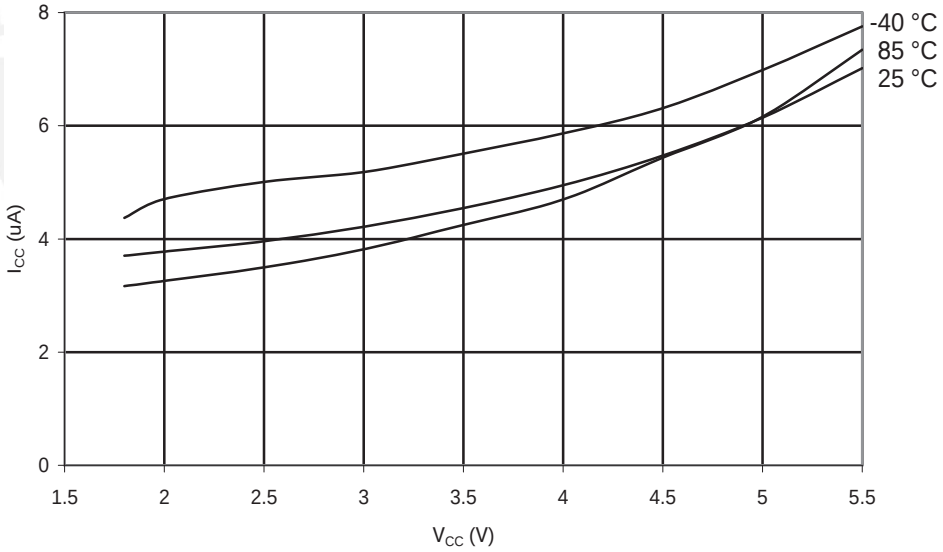
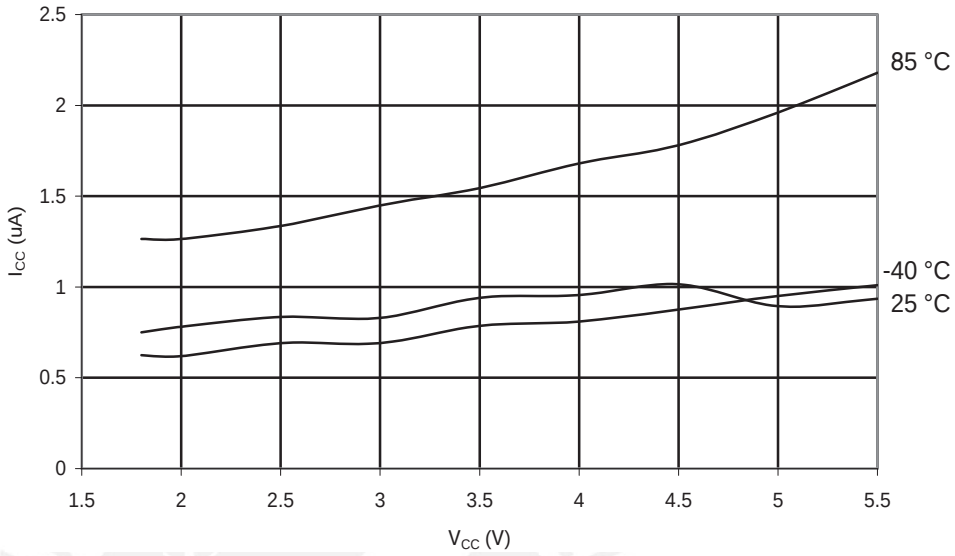


Figure 29-104. ATmega168PA: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



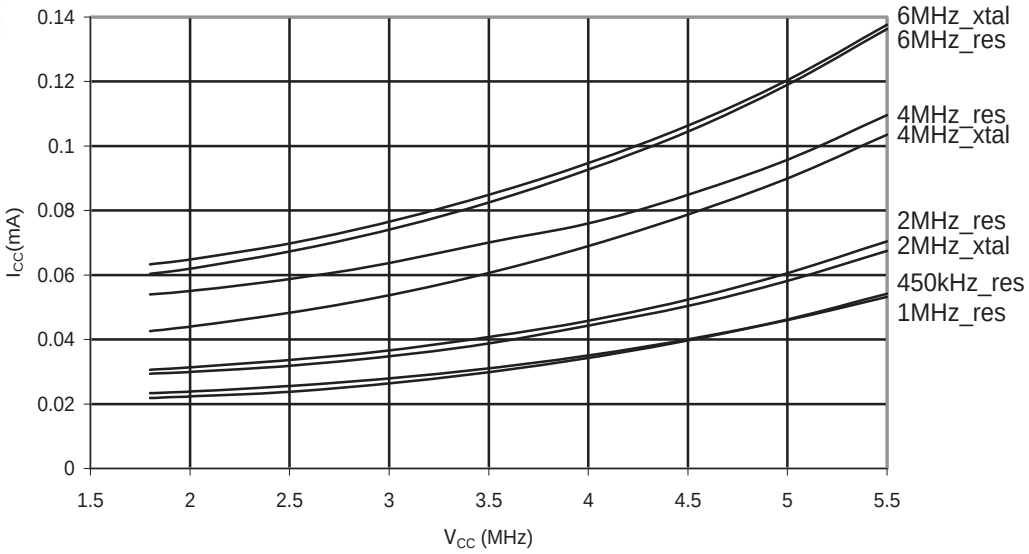
29.3.5 Power-save Supply Current

Figure 29-105. ATmega168PA: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.3.6 Standby Supply Current

Figure 29-106. ATmega168PA: Standby Supply Current vs. Vcc (Watchdog Timer Disabled)



29.3.7 Pin Pull-Up

Figure 29-107. ATmega168PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 1.8 V)

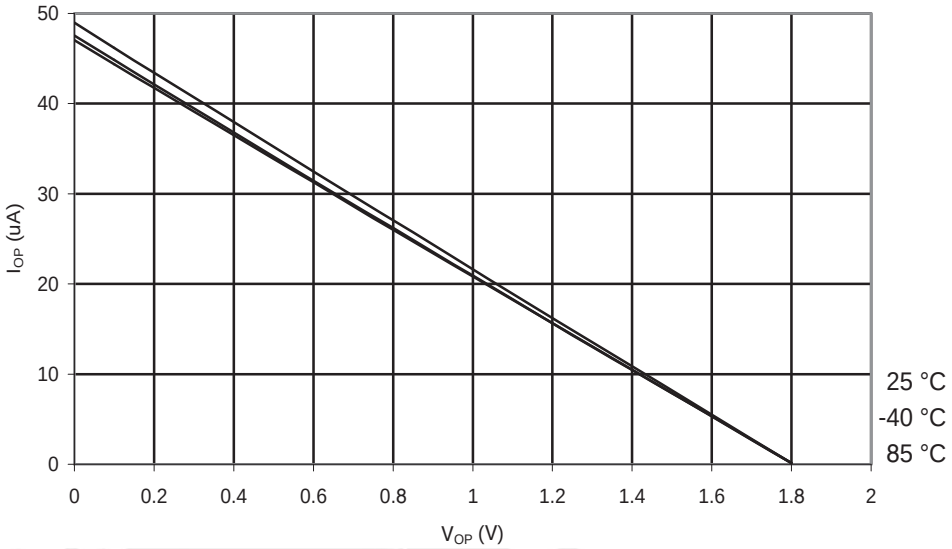


Figure 29-108. ATmega168PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 2.7 V)

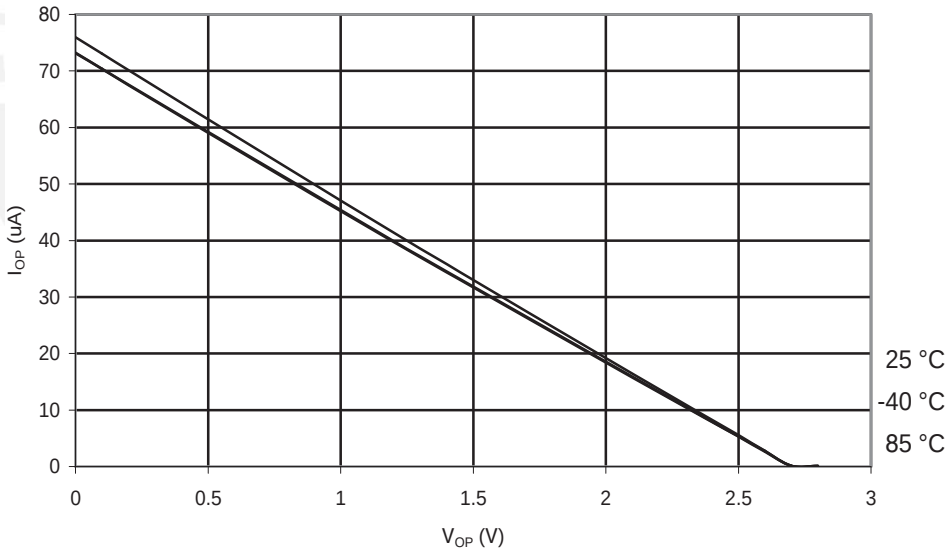


Figure 29-109. ATmega168PA: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 5 V)

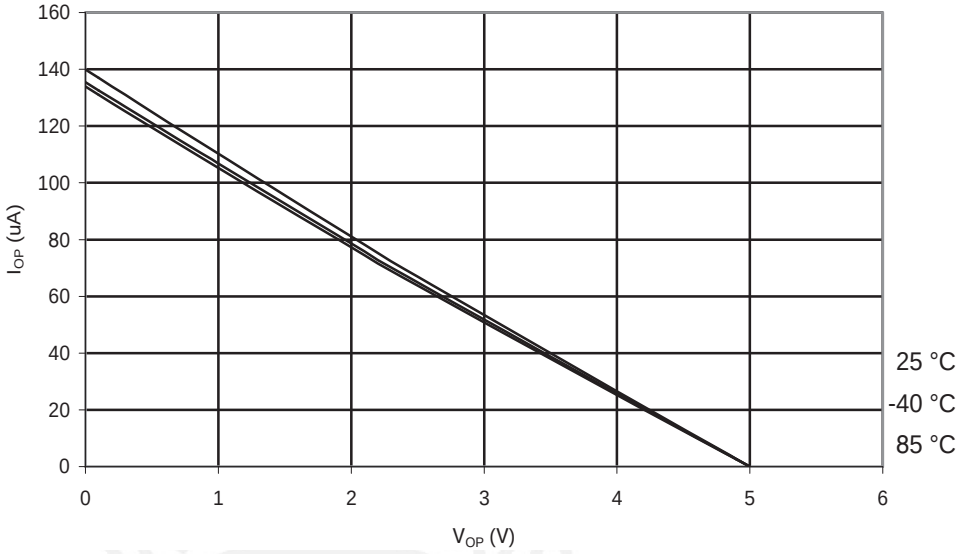


Figure 29-110. ATmega168PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 1.8 V)

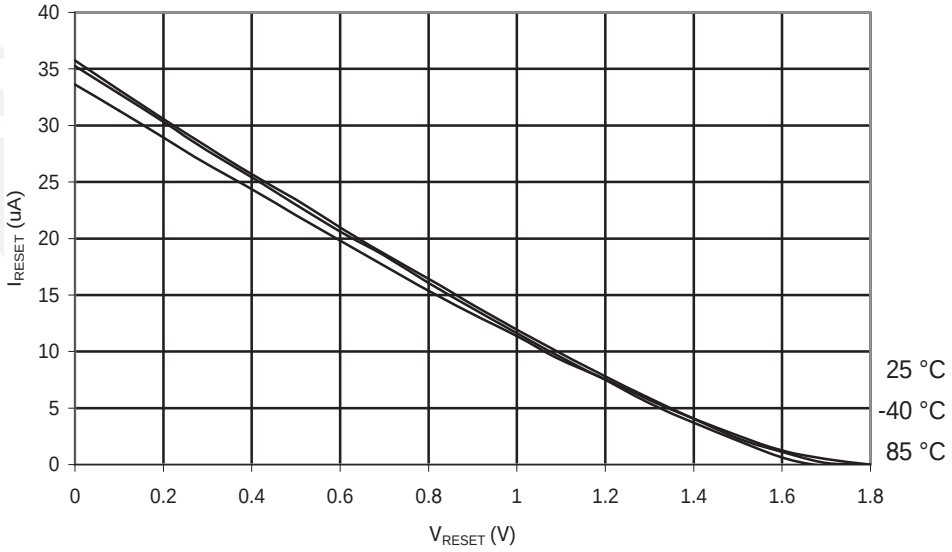


Figure 29-111. ATmega168PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 2.7 V)

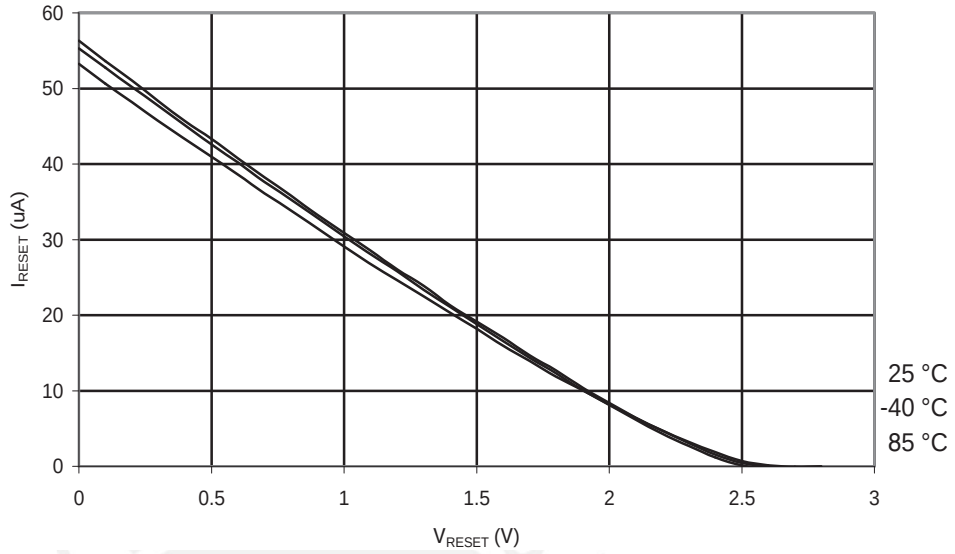
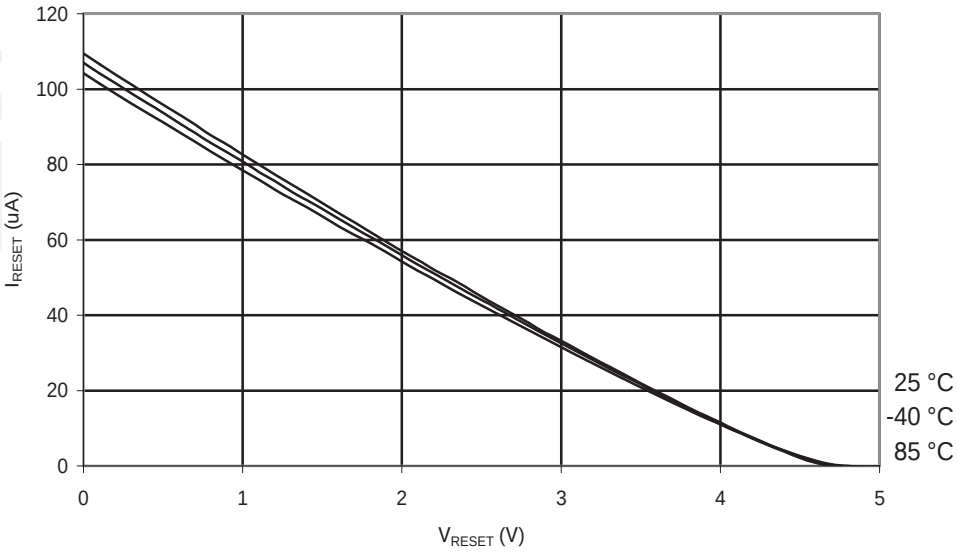


Figure 29-112. ATmega168PA: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 5V)



29.3.8 Pin Driver Strength

Figure 29-113. ATmega168PA: I/O Pin Output Voltage vs. Sink Current(V_{CC} = 3 V)

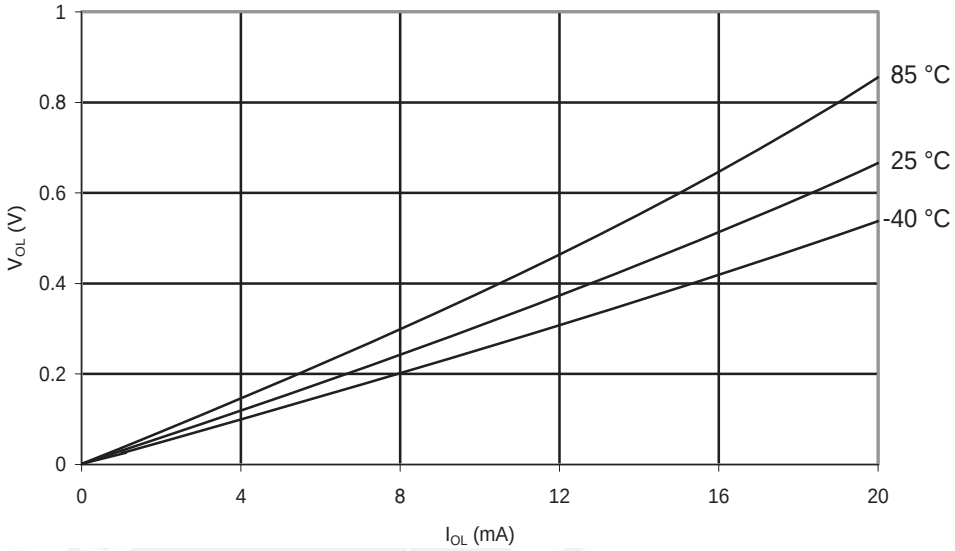


Figure 29-114. ATmega168PA: I/O Pin Output Voltage vs. Sink Current(V_{CC} = 5 V)

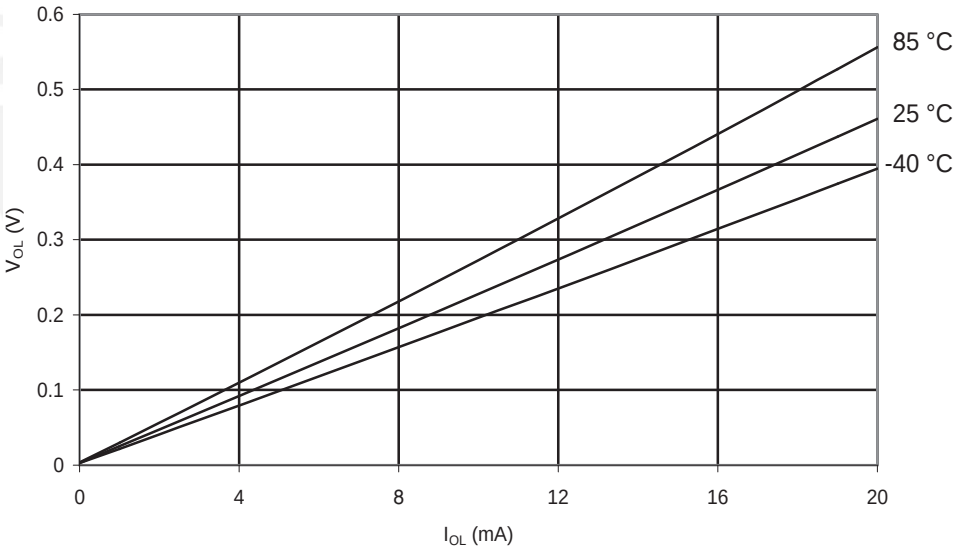


Figure 29-115. ATmega168PA: I/O Pin Output Voltage vs. Source Current(Vcc = 3 V)

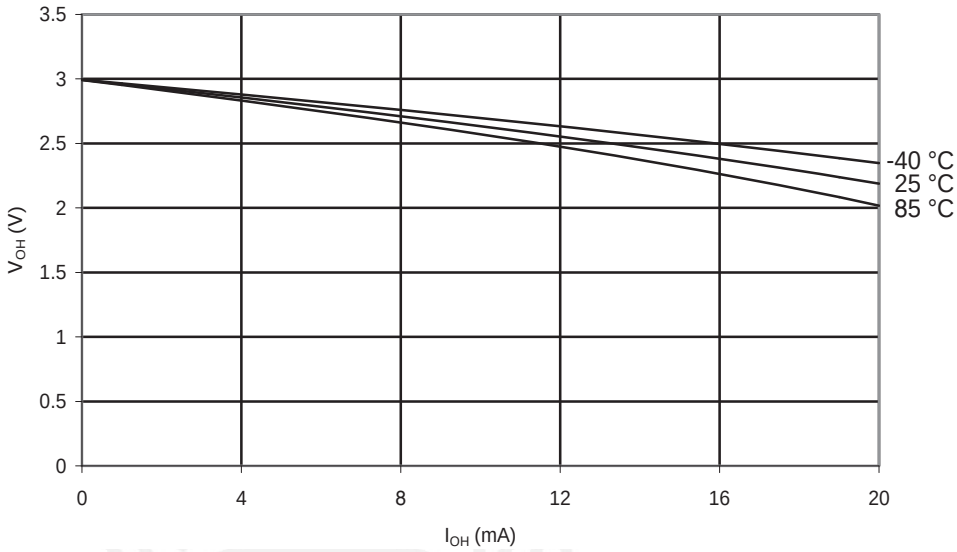
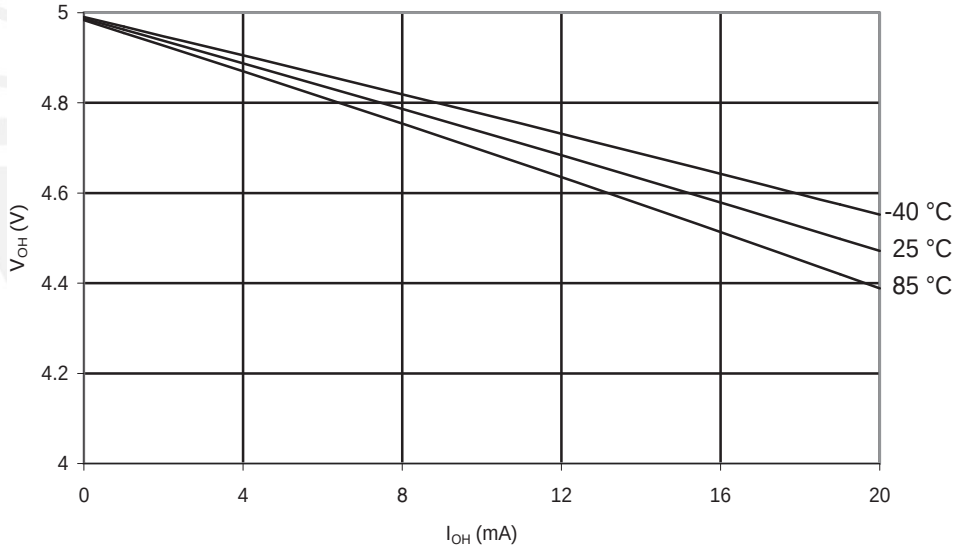


Figure 29-116. ATmega168PA: I/O Pin Output Voltage vs. Source Current(V_{CC} = 5 V)



29.3.9 Pin Threshold and Hysteresis

Figure 29-117. ATmega168PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

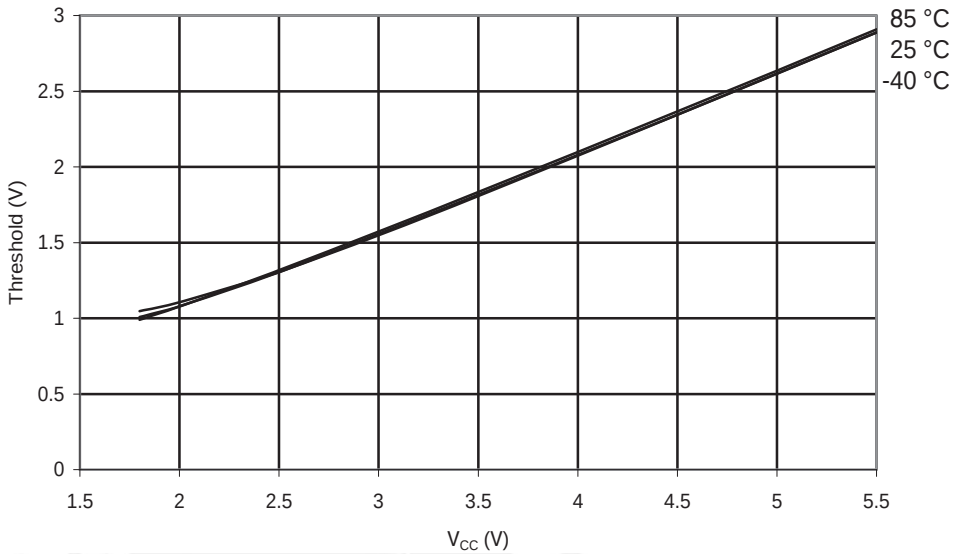


Figure 29-118. ATmega168PA: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

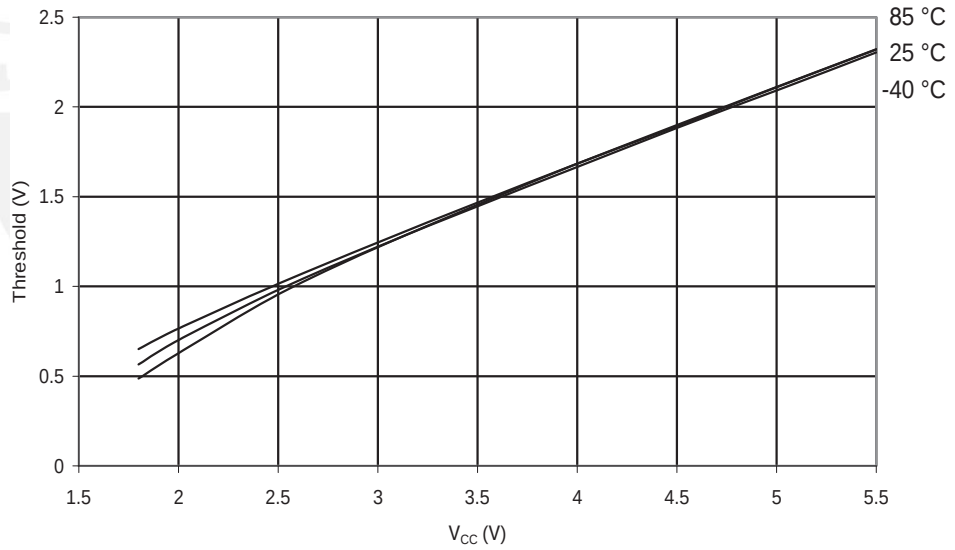


Figure 29-119. ATmega168PA: I/O Pin Input Hysteresis vs. V_{CC}

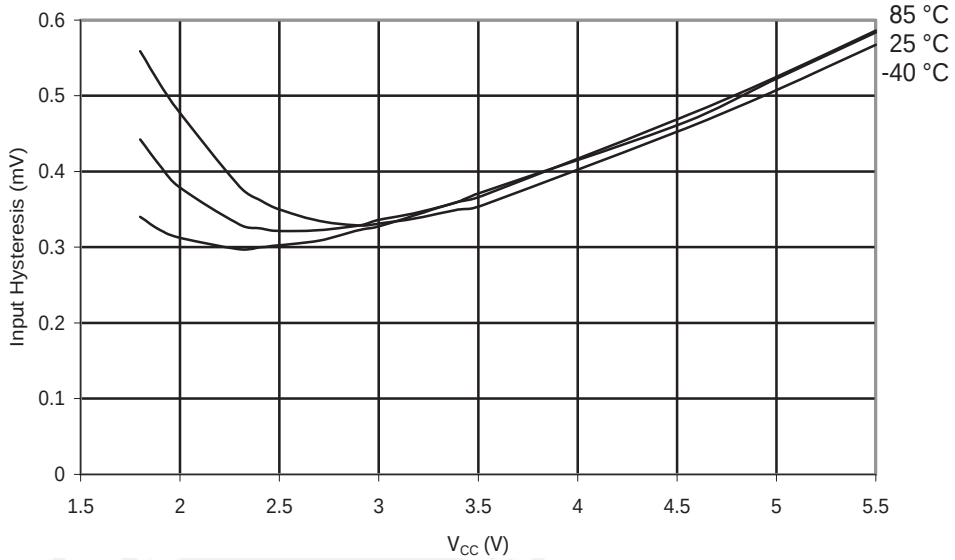


Figure 29-120. ATmega168PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

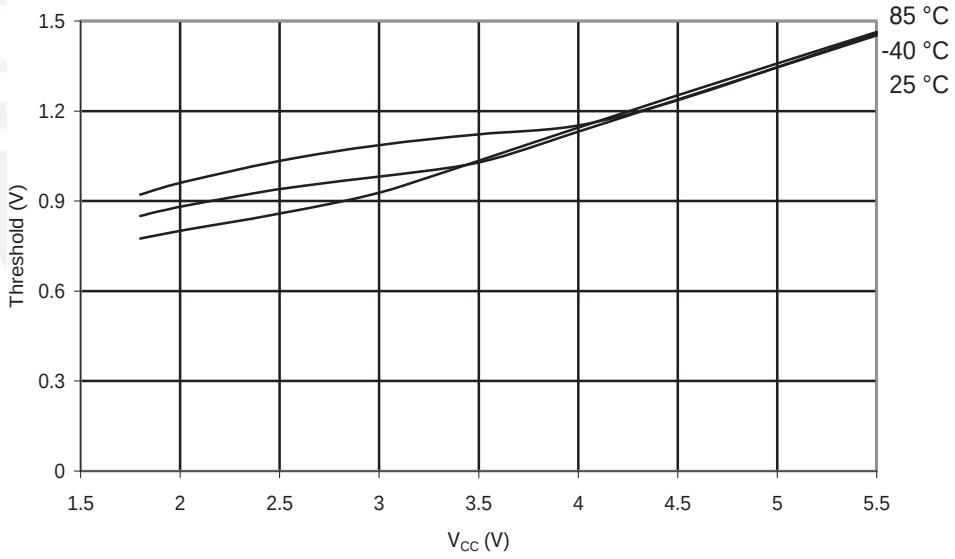


Figure 29-121. ATmega168PA: Reset Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

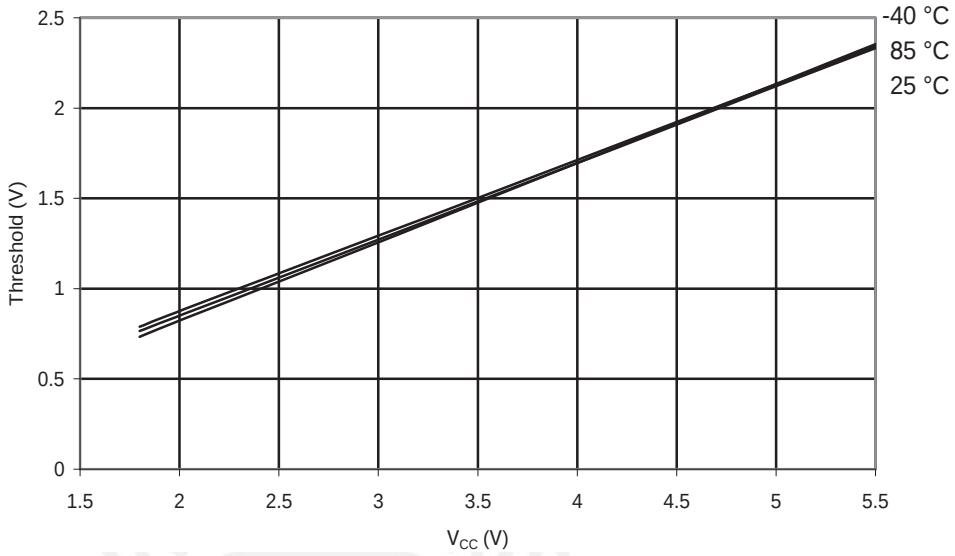
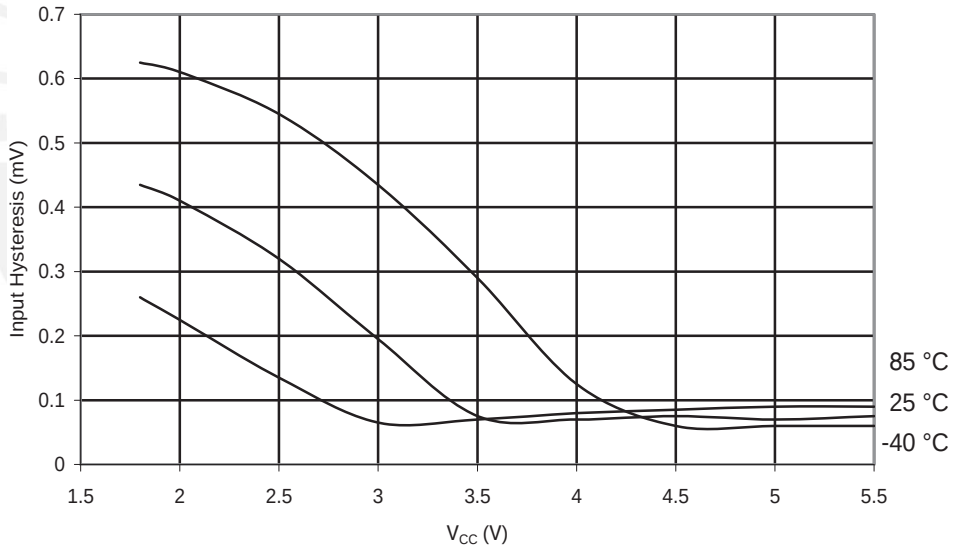


Figure 29-122. ATmega168PA: Reset Pin Input Hysteresis vs. V_{CC}



29.3.10 BOD Threshold

Figure 29-123. ATmega168PA: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

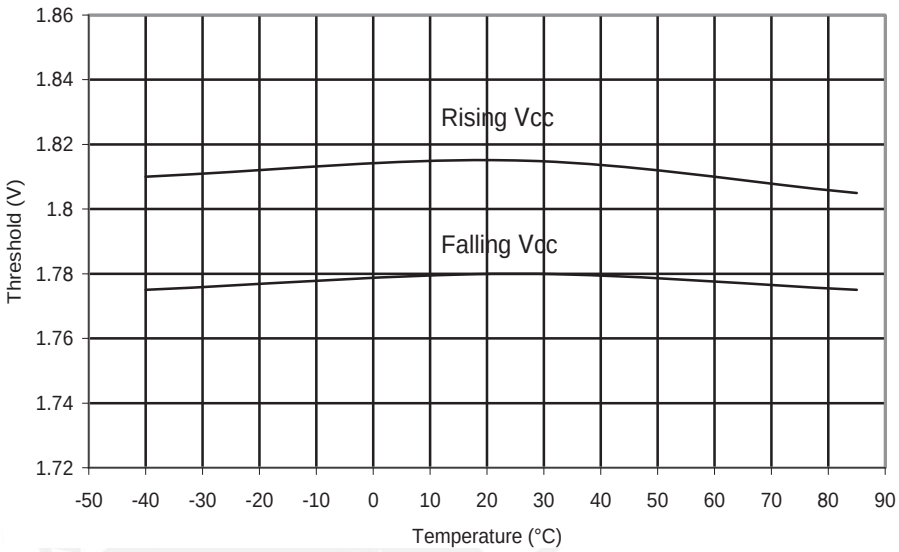


Figure 29-124. ATmega168PA: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

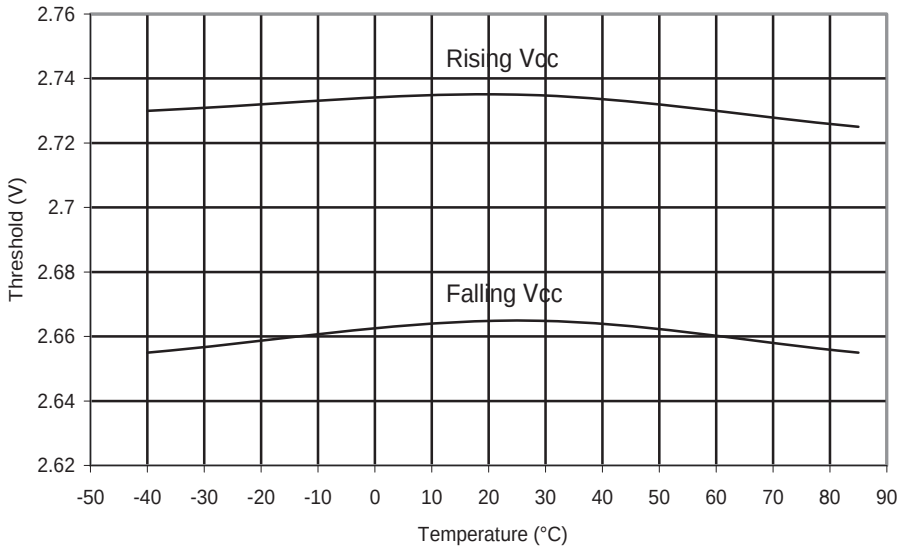
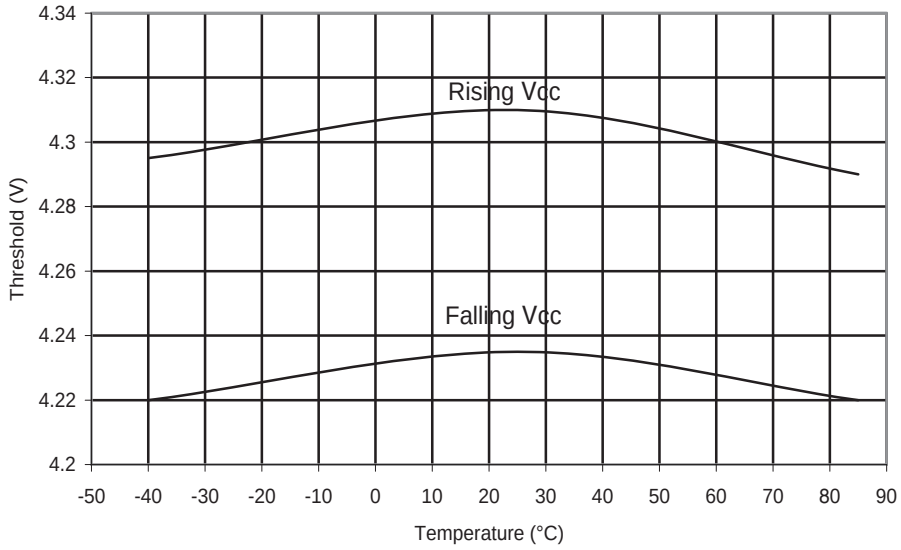


Figure 29-125.ATmega168PA: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.3.11 Internal Oscilllator Speed

Figure 29-126.ATmega168PA: Watchdog Oscillator Frequency vs. Temperature

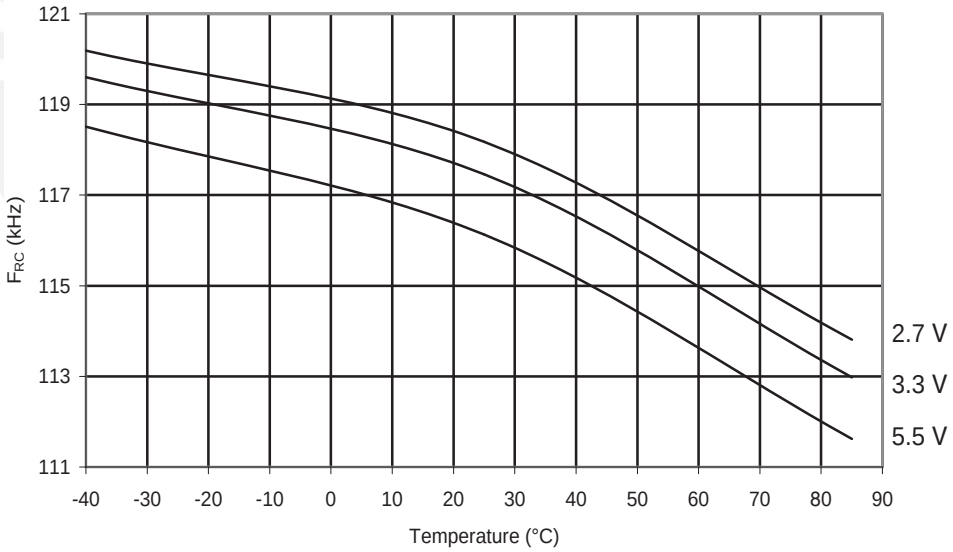


Figure 29-127. ATmega168PA: Watchdog Oscillator Frequency vs. V_{CC}

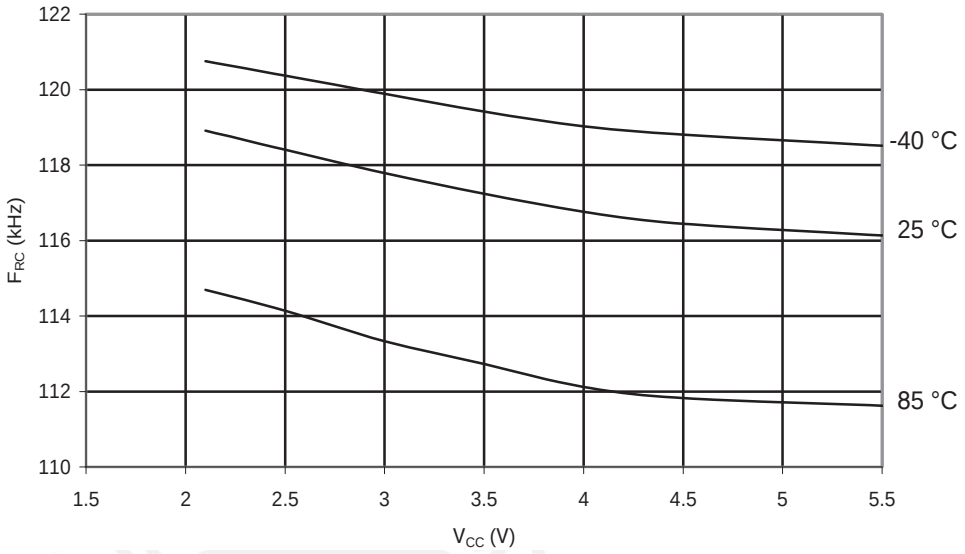


Figure 29-128. ATmega168PA: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

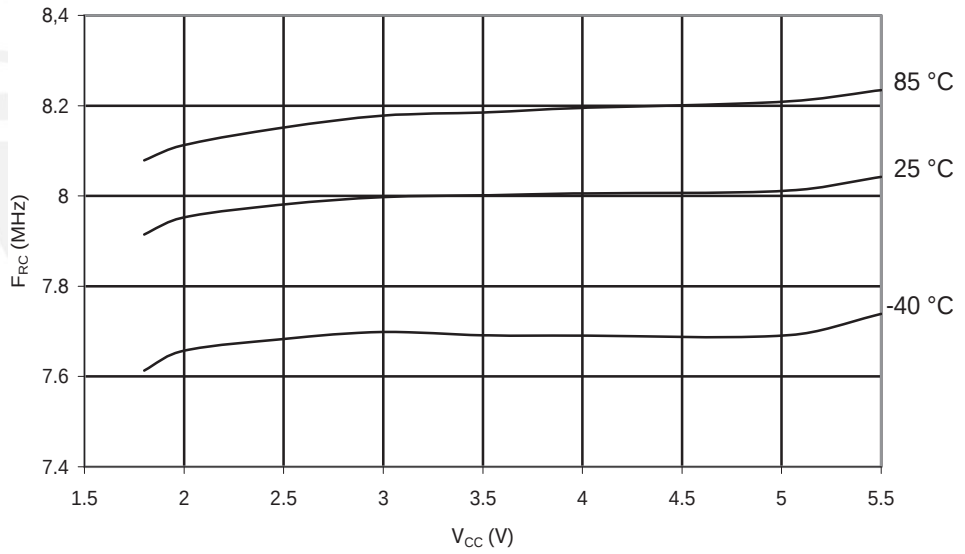


Figure 29-129.ATmega168PA: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

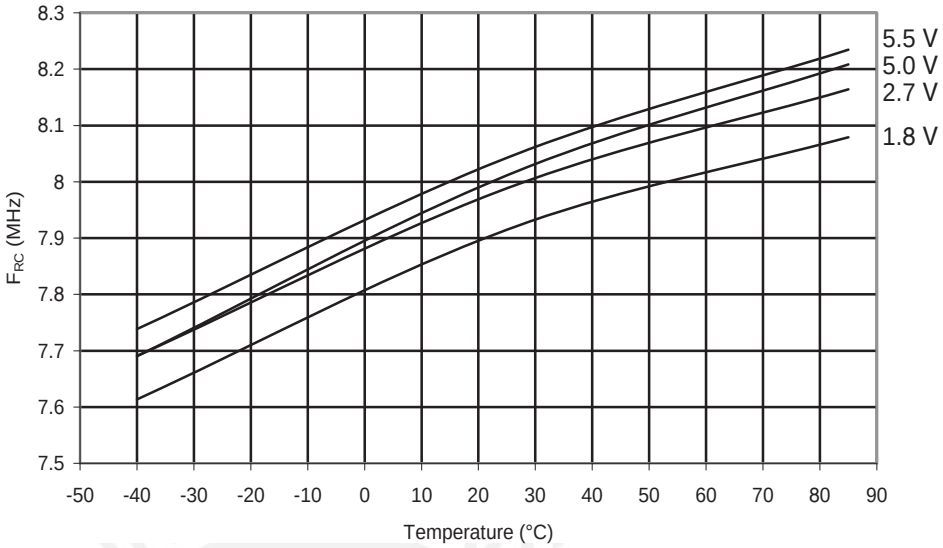
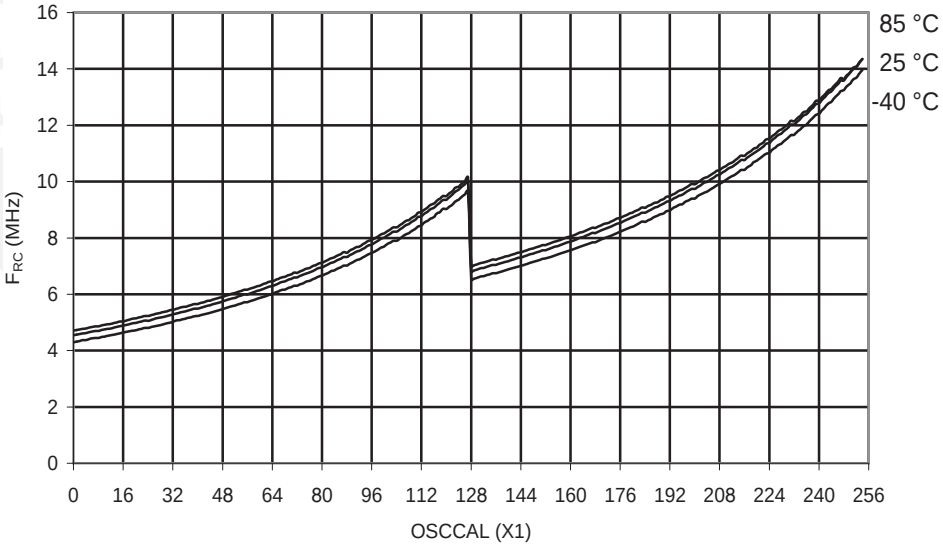


Figure 29-130.ATmega168PA: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.3.12 Current Consumption of Peripheral Units

Figure 29-131. ATmega168PA: ADC Current vs. V_{CC} ($AREF = AV_{CC}$)

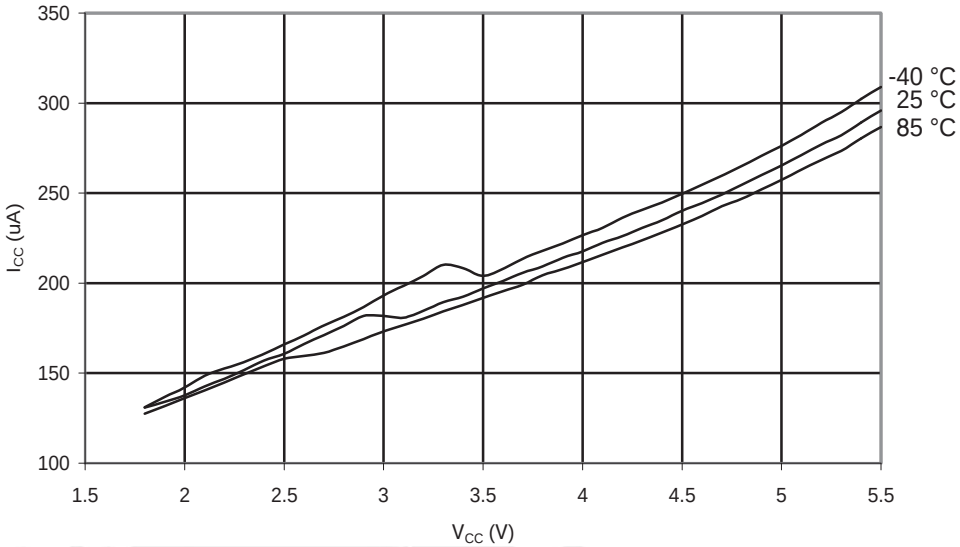


Figure 29-132. ATmega168PA: Analog Comparator Current vs. V_{CC}

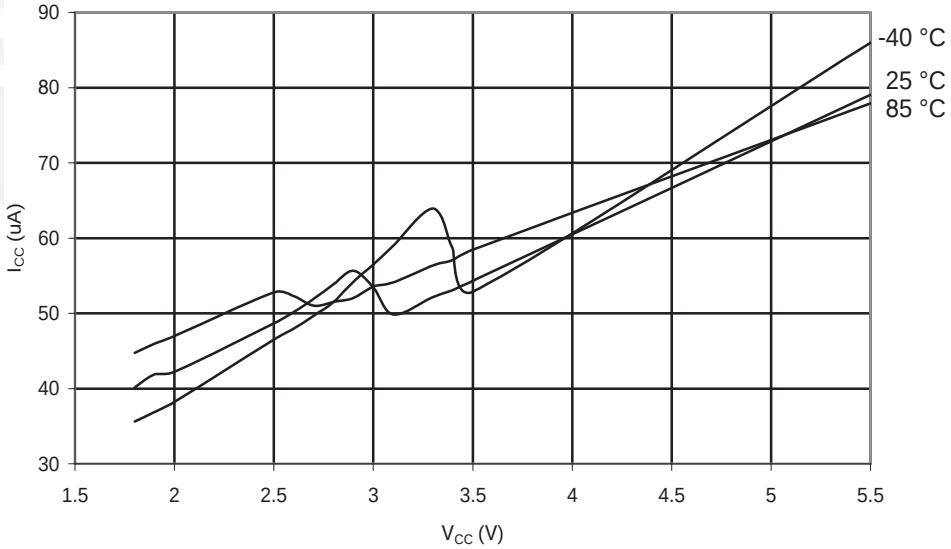


Figure 29-133. ATmega168PA: AREF External Reference Current vs. V_{CC}

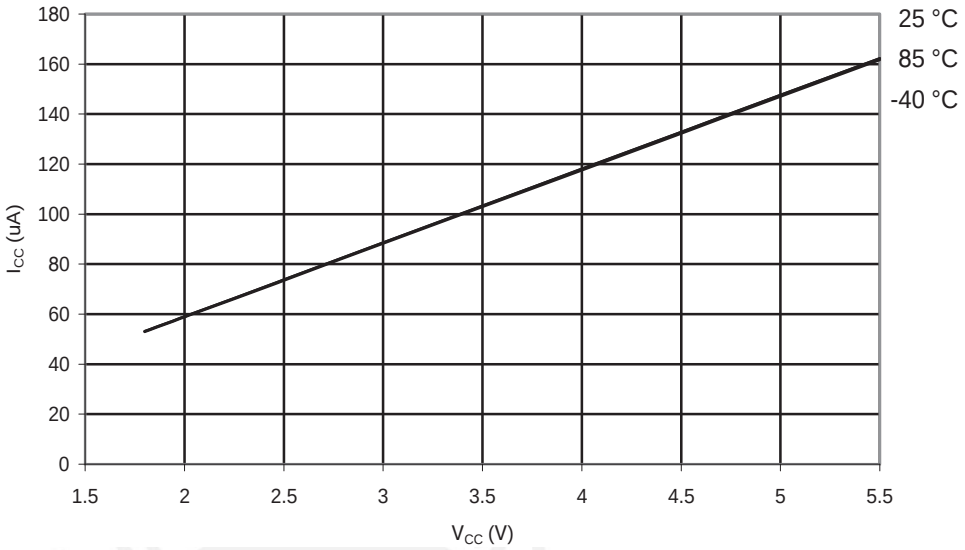


Figure 29-134. ATmega168PA: Brownout Detector Current vs. V_{CC}

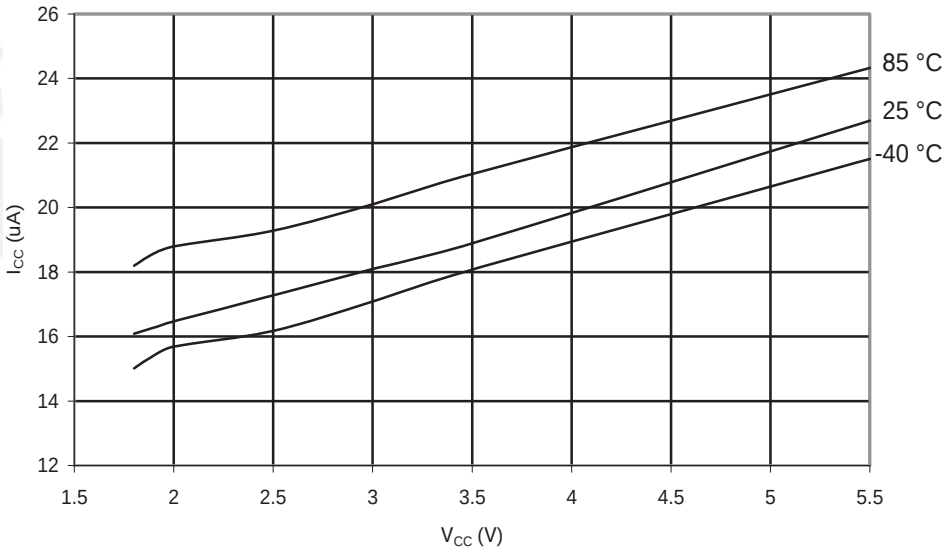
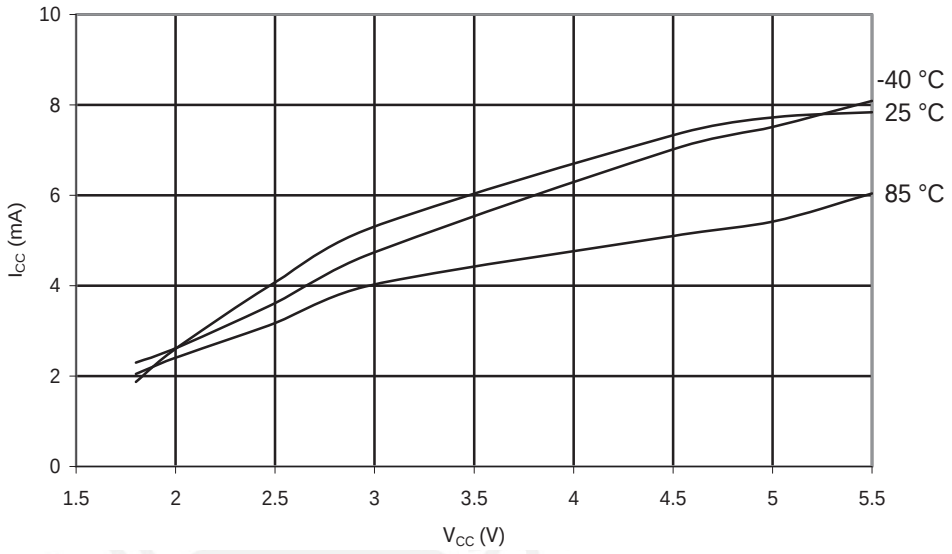


Figure 29-135.ATmega168PA: Programming Current vs. V_{CC}



29.3.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-136.ATmega168PA: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

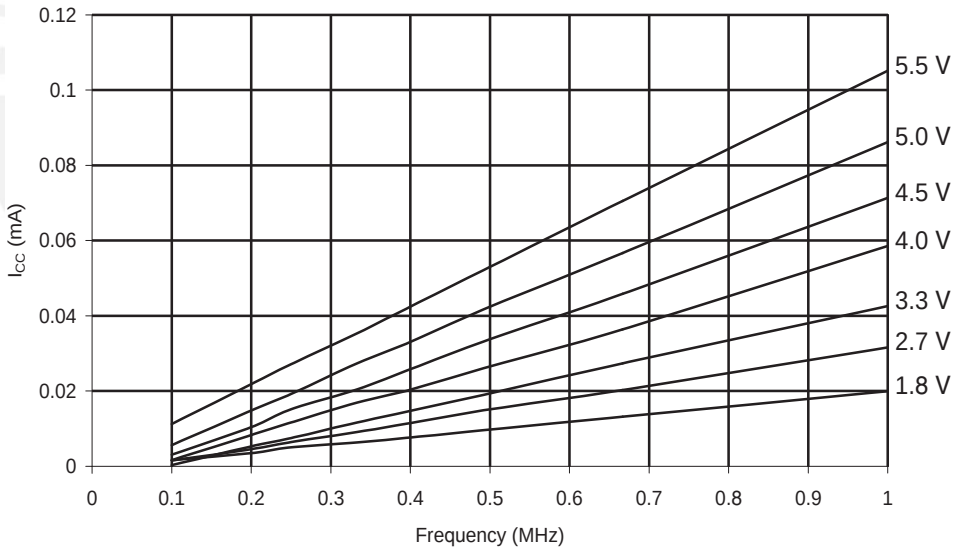


Figure 29-137. ATmega168PA: Reset Supply Current vs. Frequency (1 - 20 MHz)

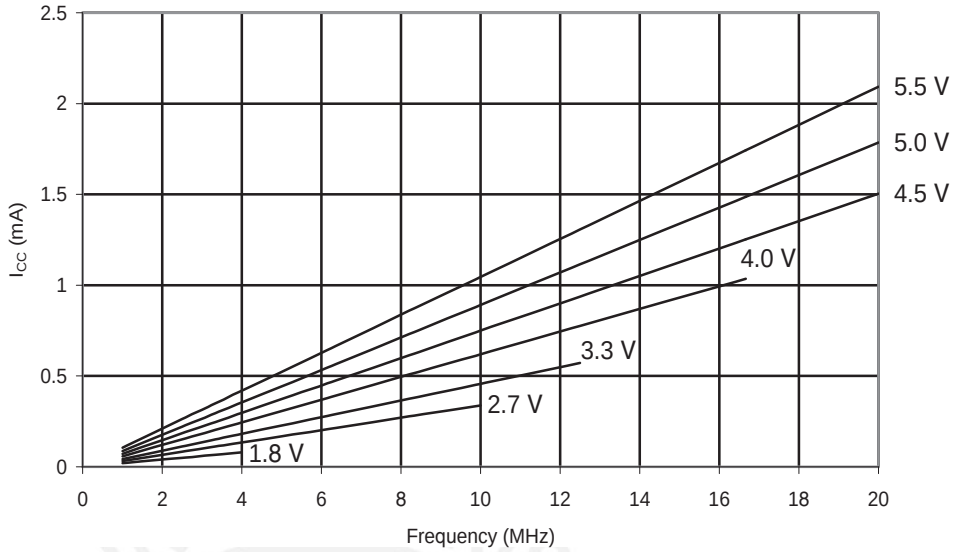
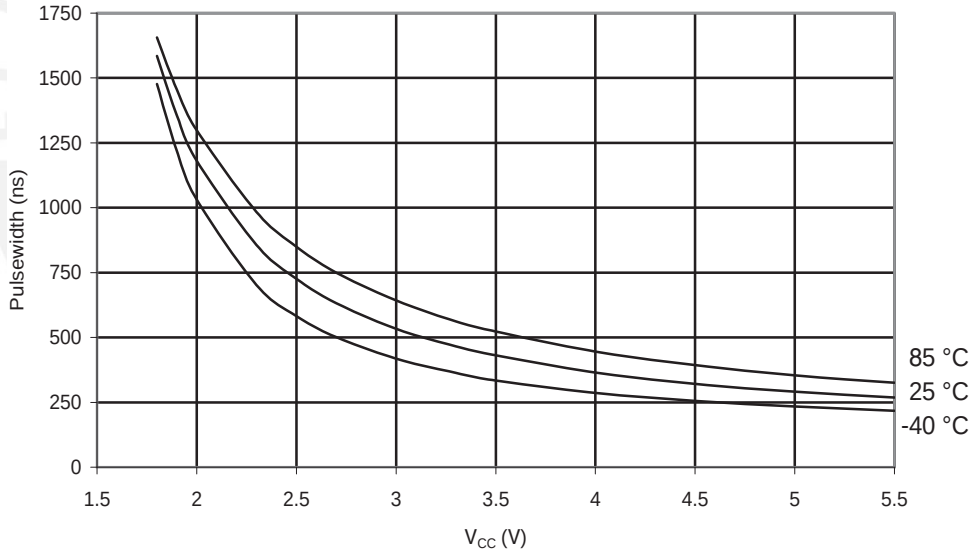


Figure 29-138. ATmega168PA: Minimum Reset Pulse width vs. V_{CC}



29.4 ATmega328P Typical Characteristics

29.4.1 Active Supply Current

Figure 29-139. ATmega328P: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

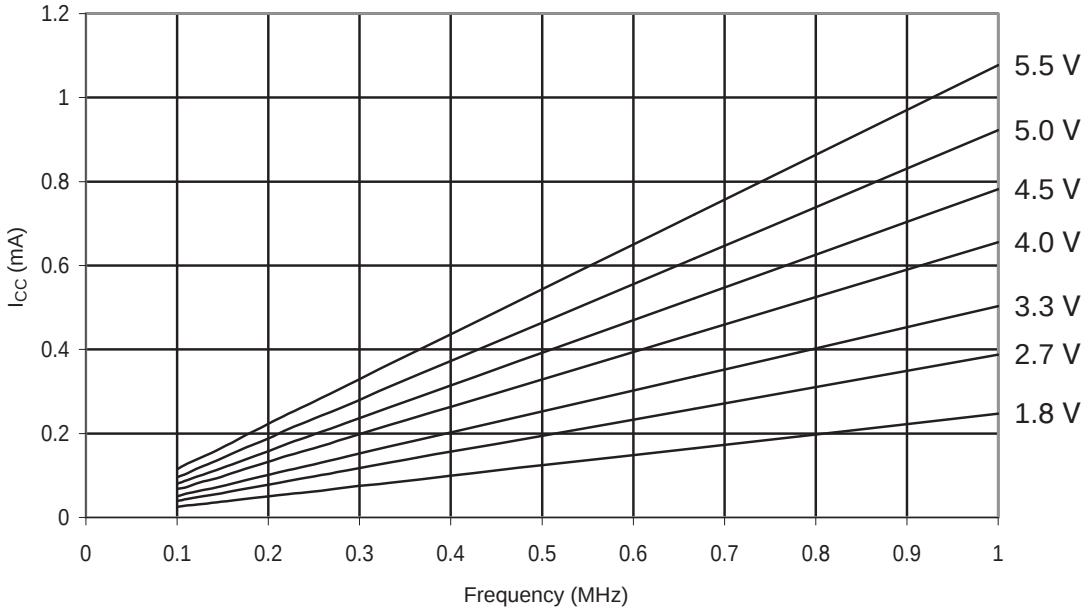


Figure 29-140. ATmega328P: Active Supply Current vs. Frequency (1-20 MHz)

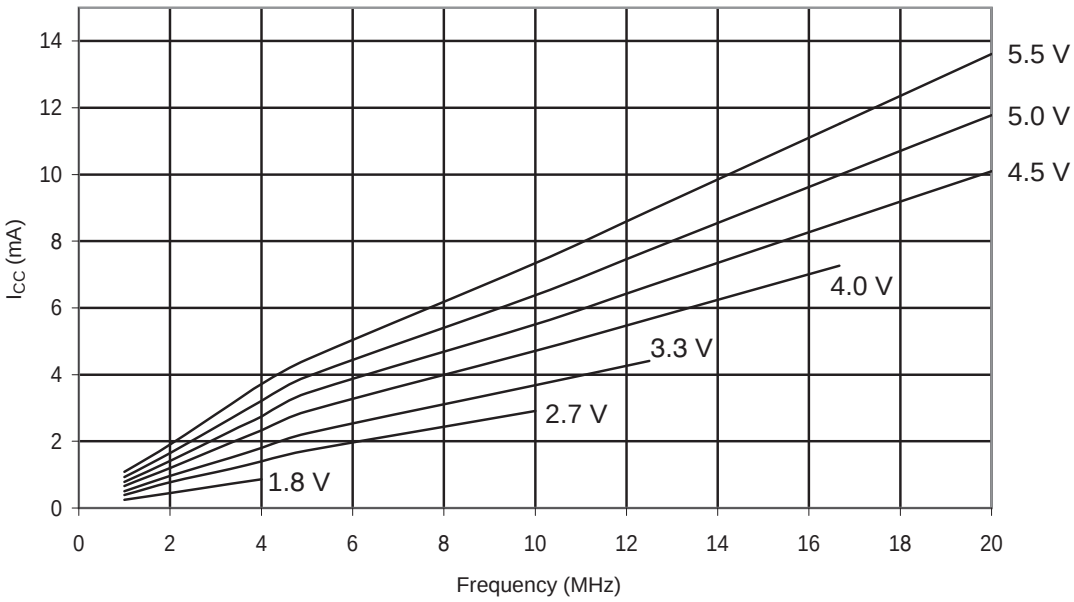


Figure 29-141. ATmega328P: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

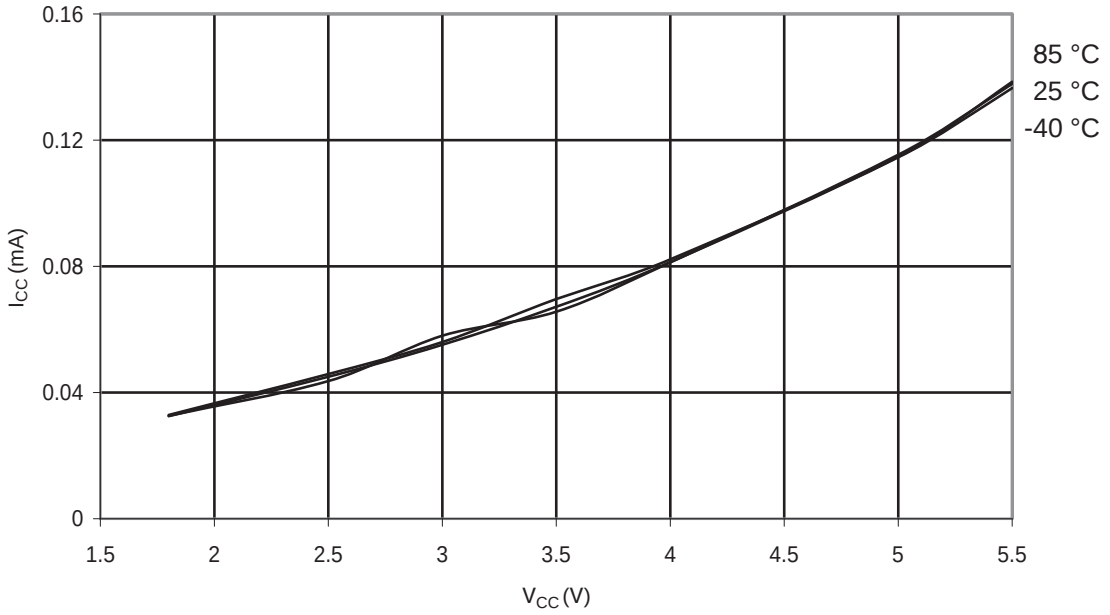


Figure 29-142. ATmega328P: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

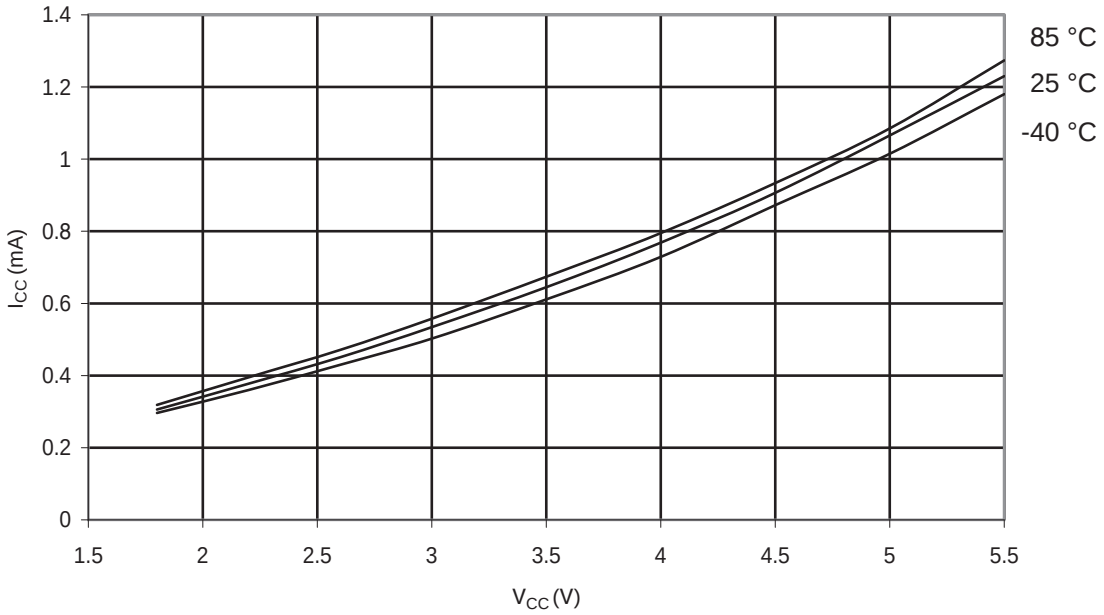
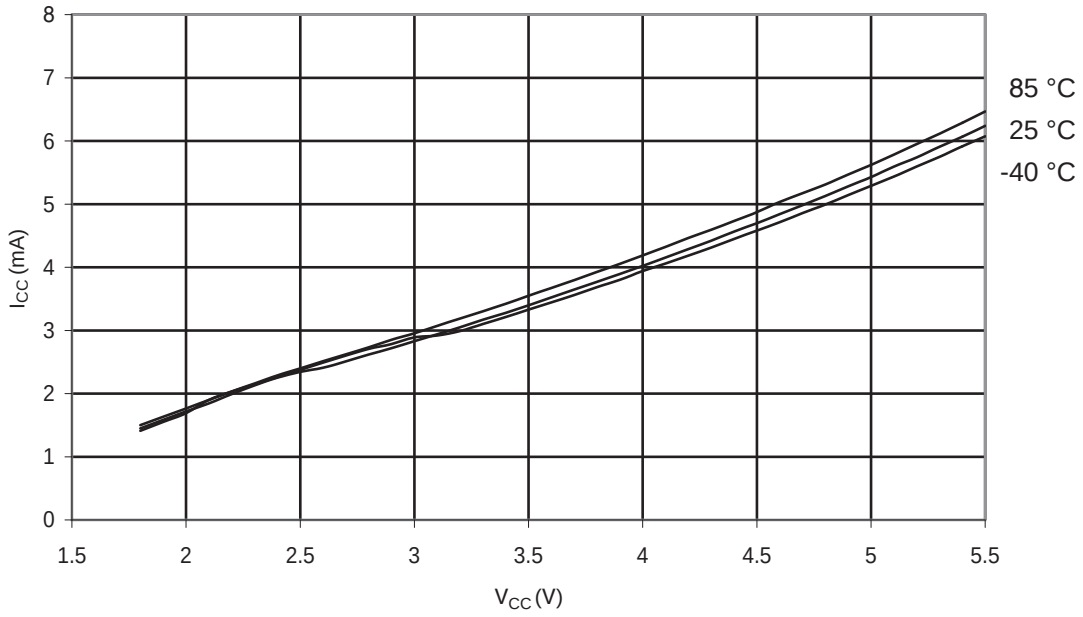


Figure 29-143. ATmega328P: Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)



29.4.2 Idle Supply Current

Figure 29-144. ATmega328P: Idle Supply Current vs. Low Frequency (0.1-1.0 MHz)

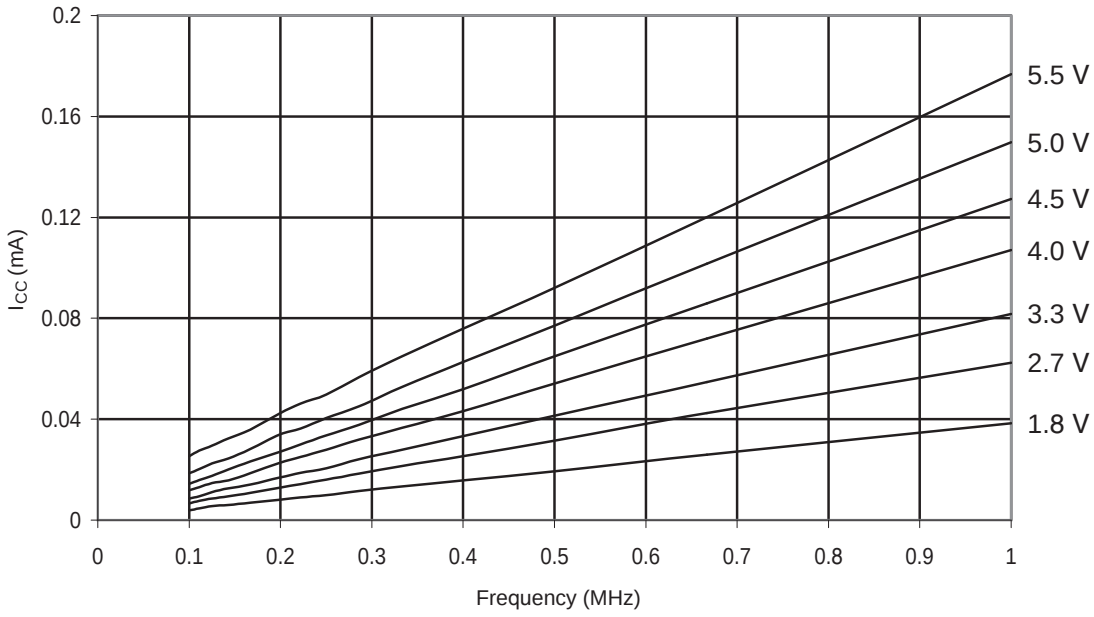


Figure 29-145. ATmega328P: Idle Supply Current vs. Frequency (1-20 MHz)

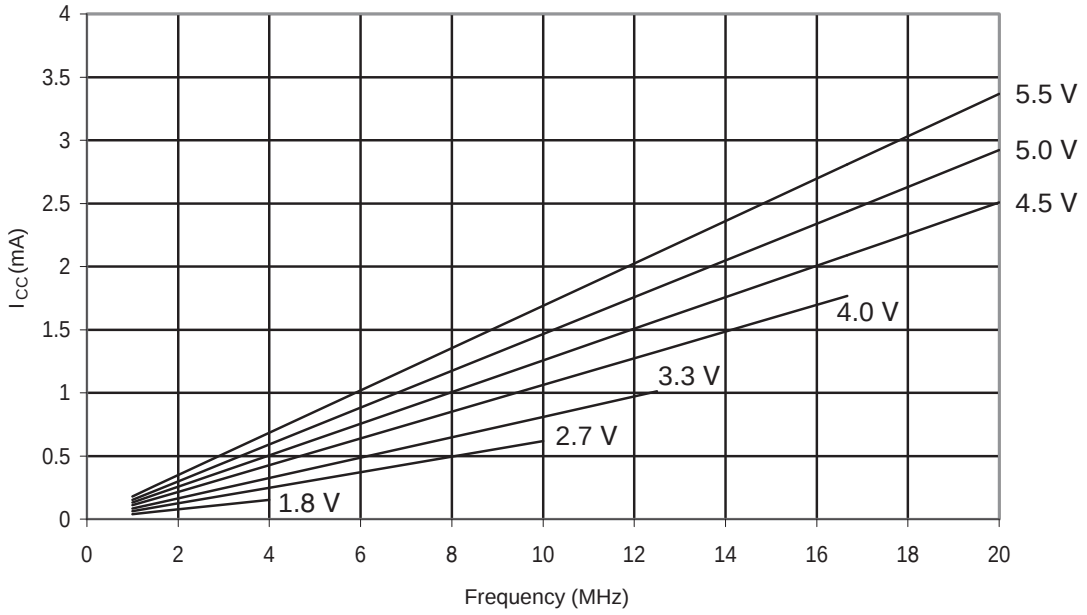


Figure 29-146. ATmega328P: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128 kHz)

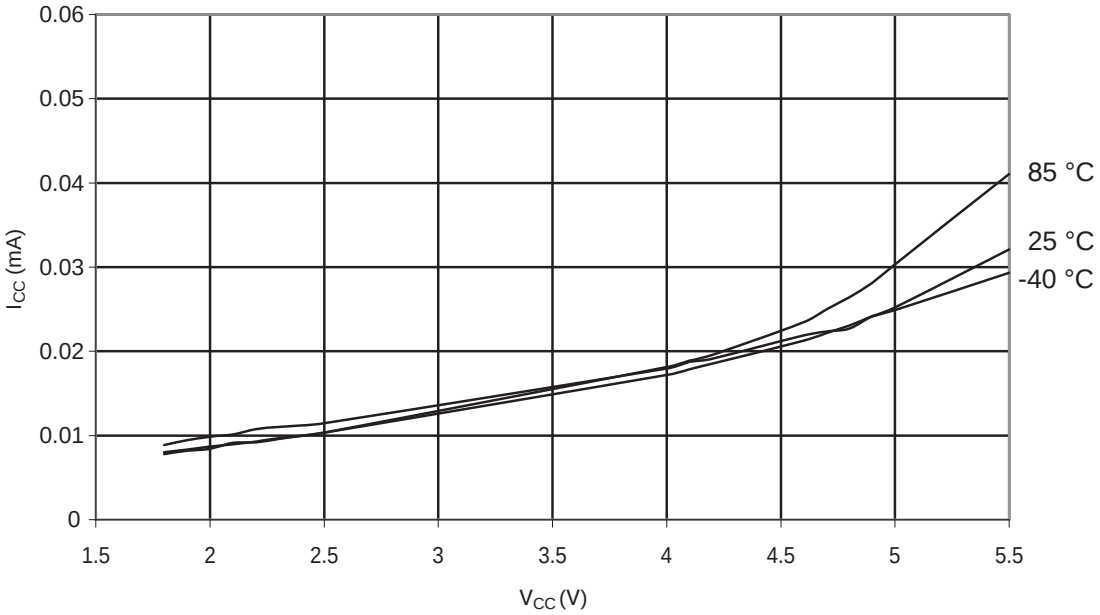


Figure 29-147. ATmega328P: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

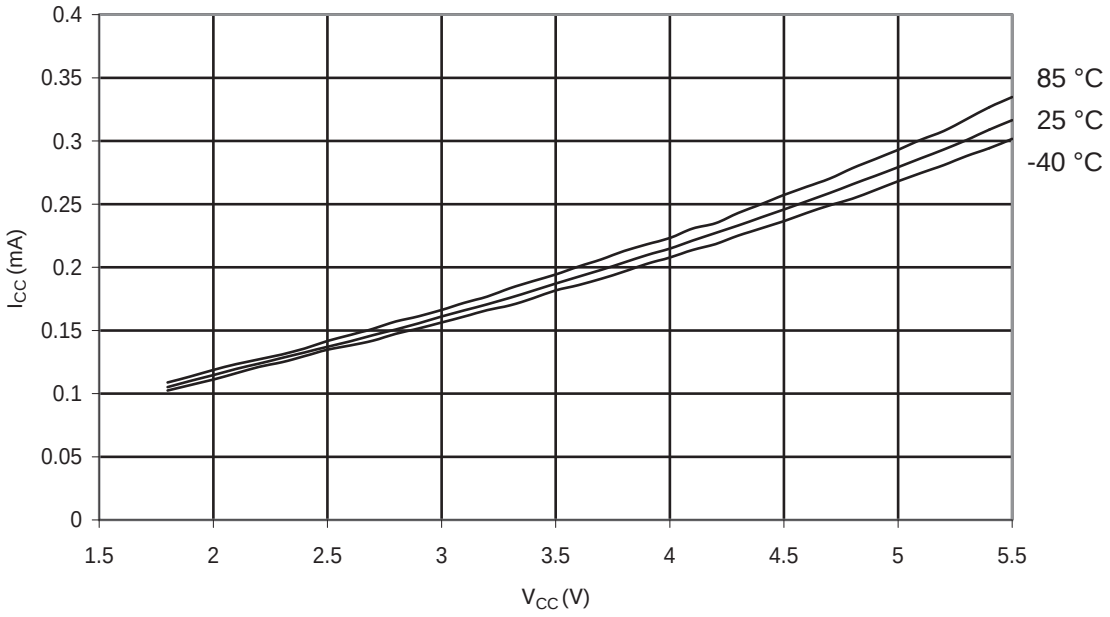
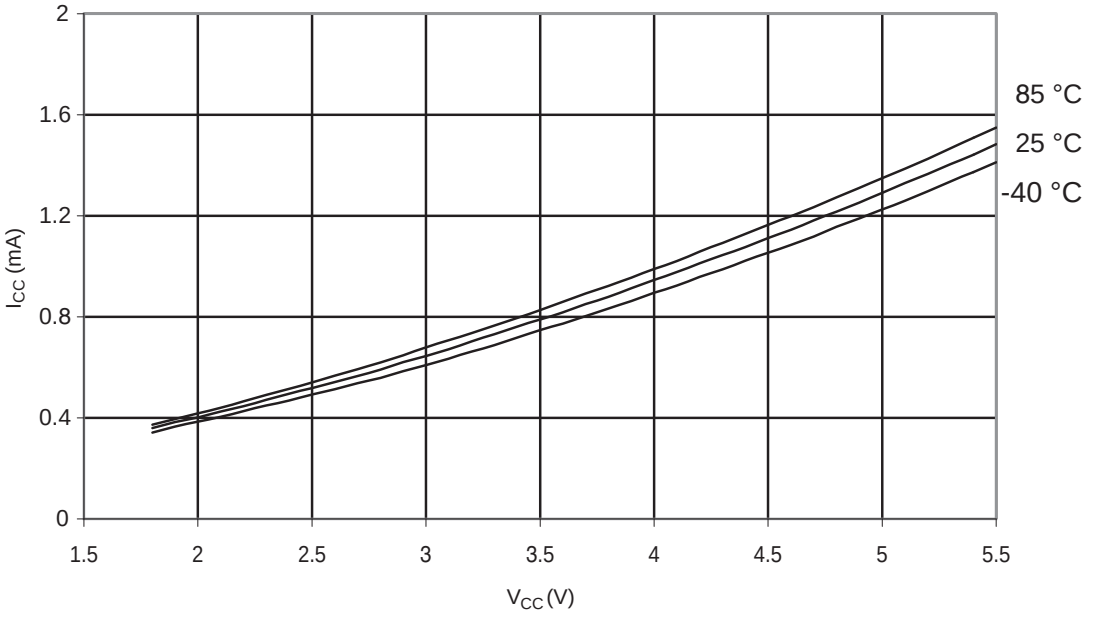


Figure 29-148. ATmega328P: Idle Supply Current vs. Vcc (Internal RC Oscillator, 8 MHz)



29.4.3 ATmega328P Supply Current of IO Modules

The tables and formulas below can be used to calculate the additional current consumption for the different I/O modules in Active and Idle mode. The enabling or disabling of the I/O modules are controlled by the Power Reduction Register. See ”Power Reduction Register” on page 42 for details.

Table 29-7. ATmega328P: Additional Current Consumption for the different I/O modules (absolute values)

PRR bit	Typical numbers		
	V_{CC} = 2V, F = 1MHz	V_{CC} = 3V, F = 4MHz	V_{CC} = 5V, F = 8MHz
PRUSART0	3.20 µA	22.17 µA	100.25 µA
PRTWI	7.34 µA	46.55 µA	199.25 µA
PRTIM2	7.34 µA	50.79 µA	224.25 µA
PRTIM1	6.19 µA	41.25 µA	176.25 µA
PRTIM0	1.89 µA	14.28 µA	61.13 µA
PRSPI	6.94 µA	43.84 µA	186.50 µA
PRADC	8.66 µA	61.80 µA	295.38 µA

Table 29-8. ATmega328P: Additional Current Consumption (percentage) in Active and Idle mode

PRR bit	Additional Current consumption compared to Active with external clock (see Figure 29-139 on page 399 and Figure 29-140 on page 399)	Additional Current consumption compared to Idle with external clock (see Figure 29-144 on page 401 and Figure 29-145 on page 402)
PRUSART0	1.4 %	7.8%
PRTWI	3.0 %	16.6 %
PRTIM2	3.3 %	17.8 %
PRTIM1	2.7 %	14.5 %
PRTIM0	0.9 %	4.8 %
PRSPI	2.9 %	15.7 %
PRADC	4.1 %	22.1 %

It is possible to calculate the typical current consumption based on the numbers from Table 29-8 on page 404 for other V_{CC} and frequency settings than listed in Table 29-7 on page 404.

Example

Calculate the expected current consumption in idle mode with TIMER1, ADC, and SPI enabled at V_{CC} = 2.0V and F = 1MHz. From Table 29-8 on page 404, third column, we see that we need to add 14.5% for the TIMER1, 22.1% for the ADC, and 15.7% for the SPI module. Reading from Figure 29-145 on page 402, we find that the idle current consumption is ~0.055 mA at V_{CC} = 2.0V and F = 1MHz. The total current consumption in idle mode with TIMER1, ADC, and SPI enabled, gives:

$$I_{CCtotal} \approx 0.045\ mA \cdot (1 + 0.145 + 0.221 + 0.157) \approx 0.069\ mA$$

29.4.4 Power-down Supply Current

Figure 29-149. ATmega328P: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Disabled)

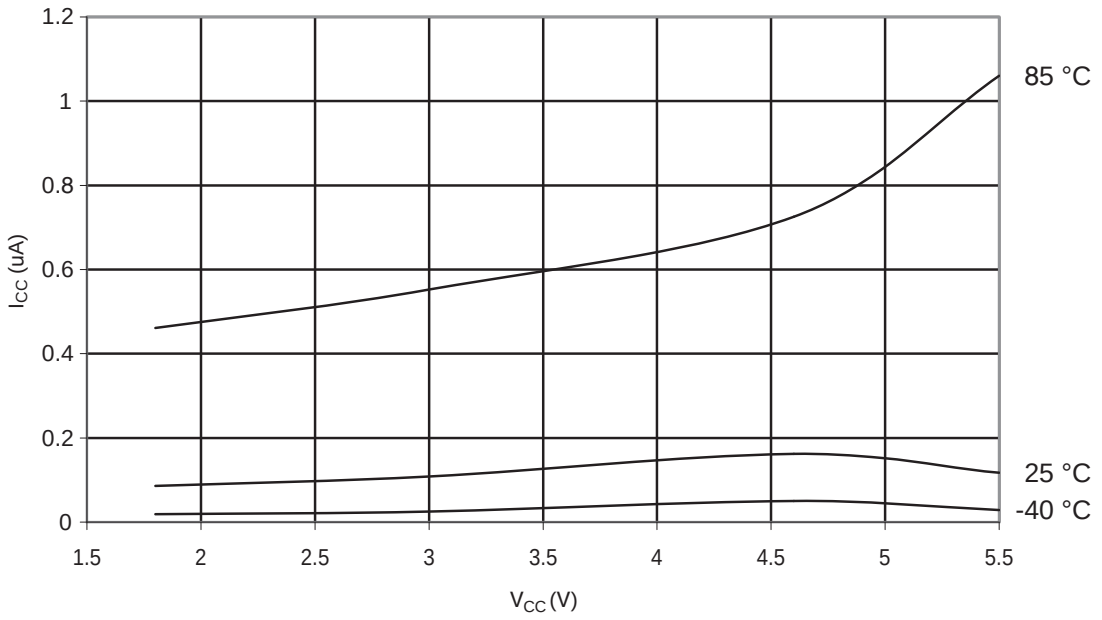
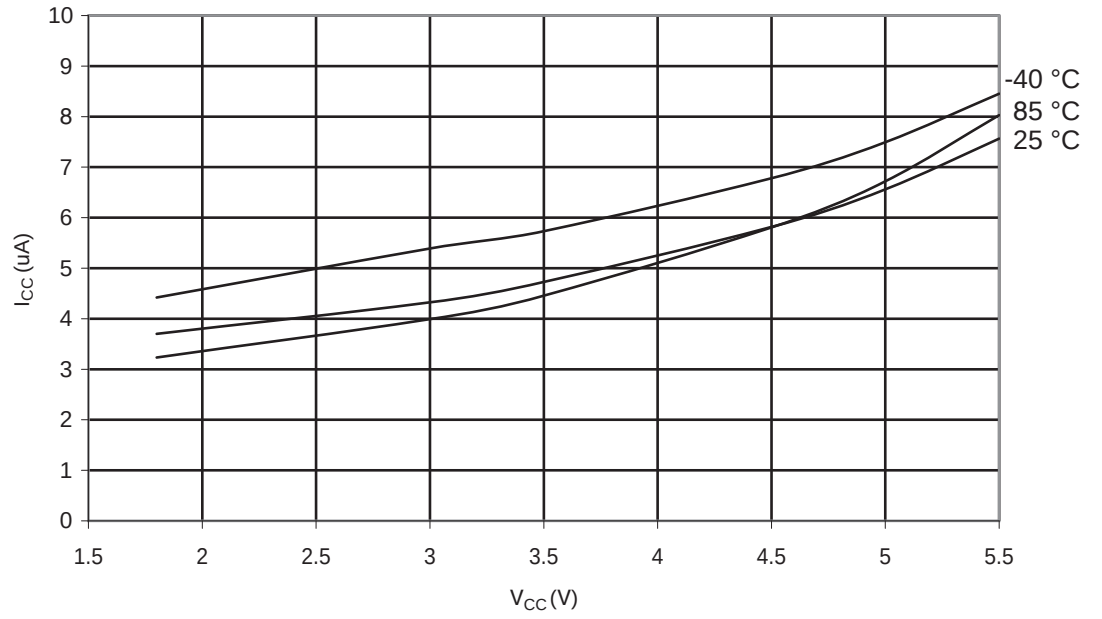
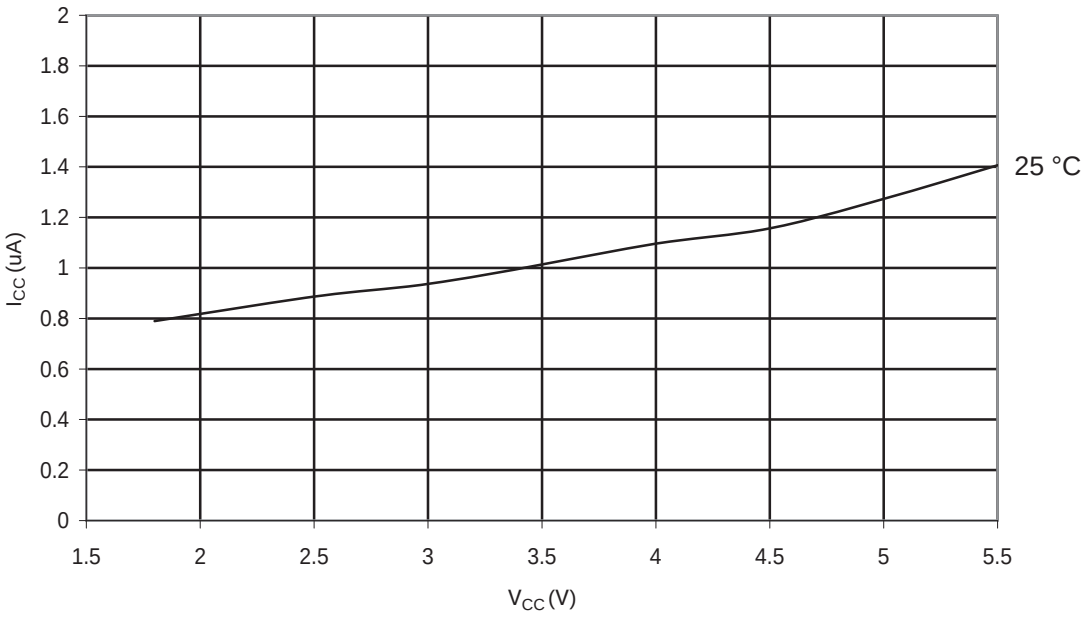


Figure 29-150. ATmega328P: Power-Down Supply Current vs. V_{CC} (Watchdog Timer Enabled)



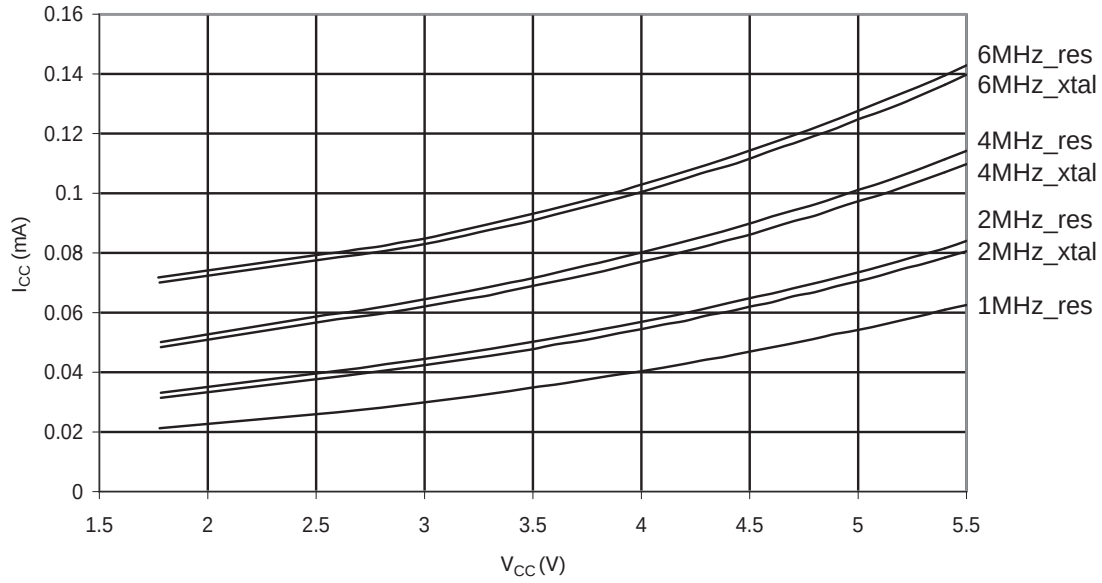
29.4.5 Power-save Supply Current

Figure 29-151. ATmega328P: Power-Save Supply Current vs. V_{CC} (Watchdog Timer Disabled and 32 kHz Crystal Oscillator Running)



29.4.6 Standby Supply Current

Figure 29-152. ATmega328P: Standby Supply Current vs. Vcc (Watchdog Timer Disabled)



29.4.7 Pin Pull-Up

Figure 29-153. ATmega328P: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 1.8 V)

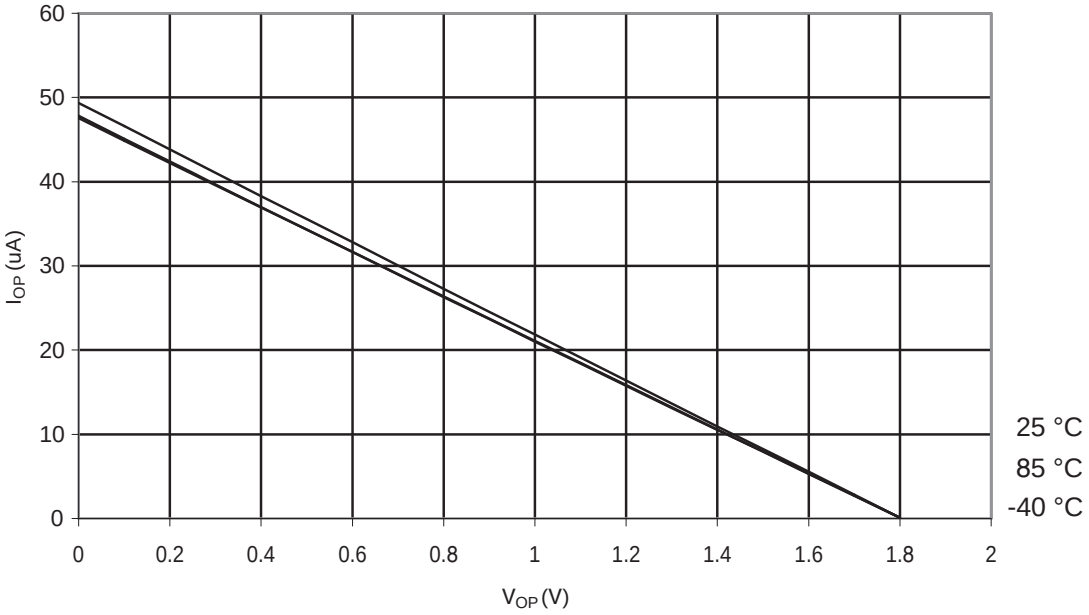


Figure 29-154. ATmega328P: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 2.7 V)

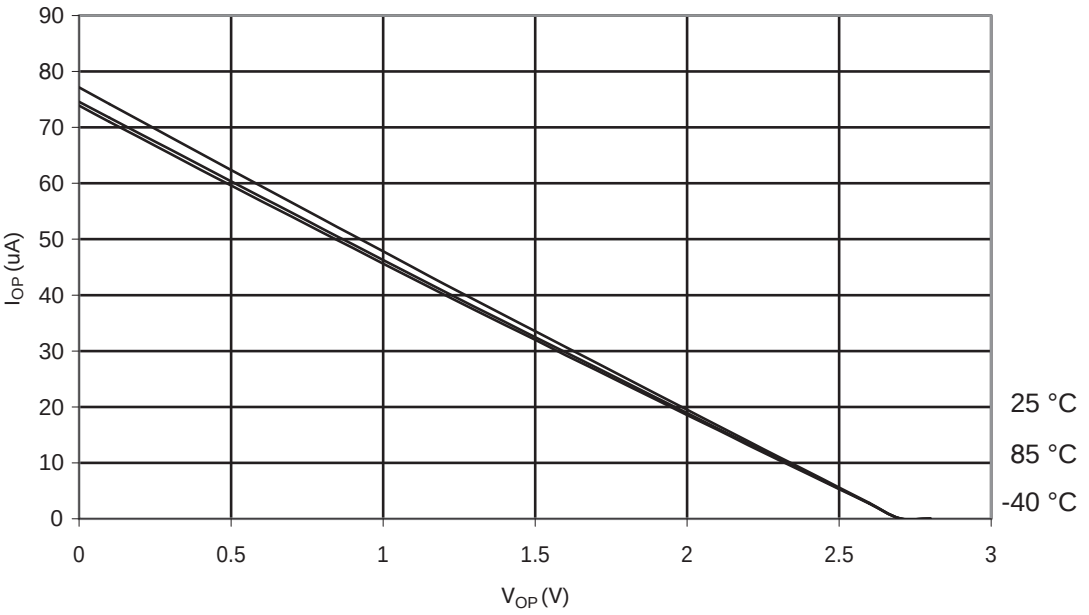


Figure 29-155. ATmega328P: I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 5 V)

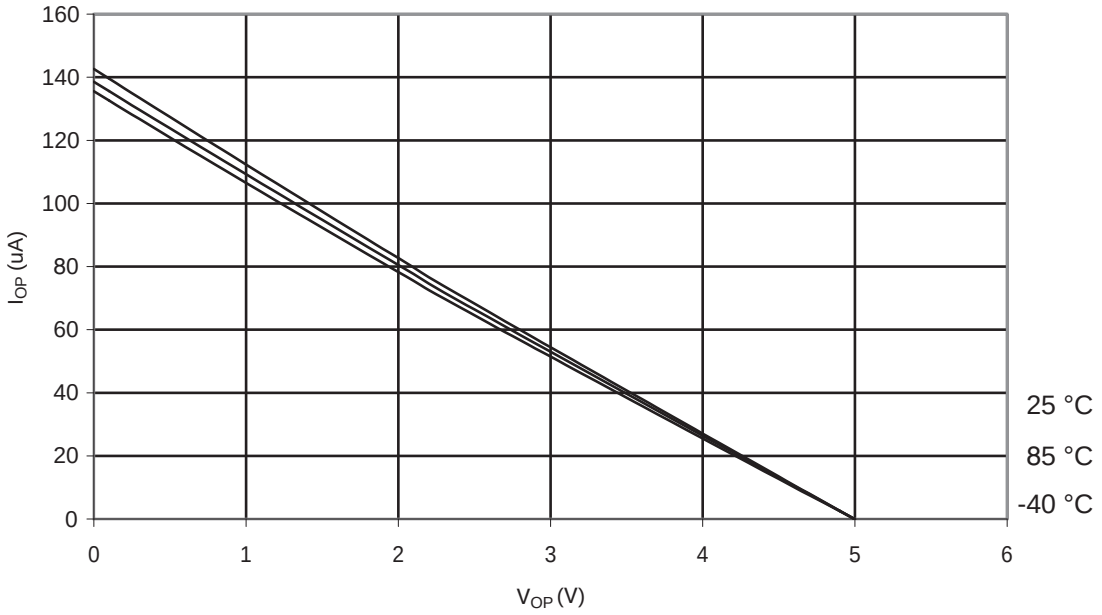


Figure 29-156. ATmega328P: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 1.8 V)

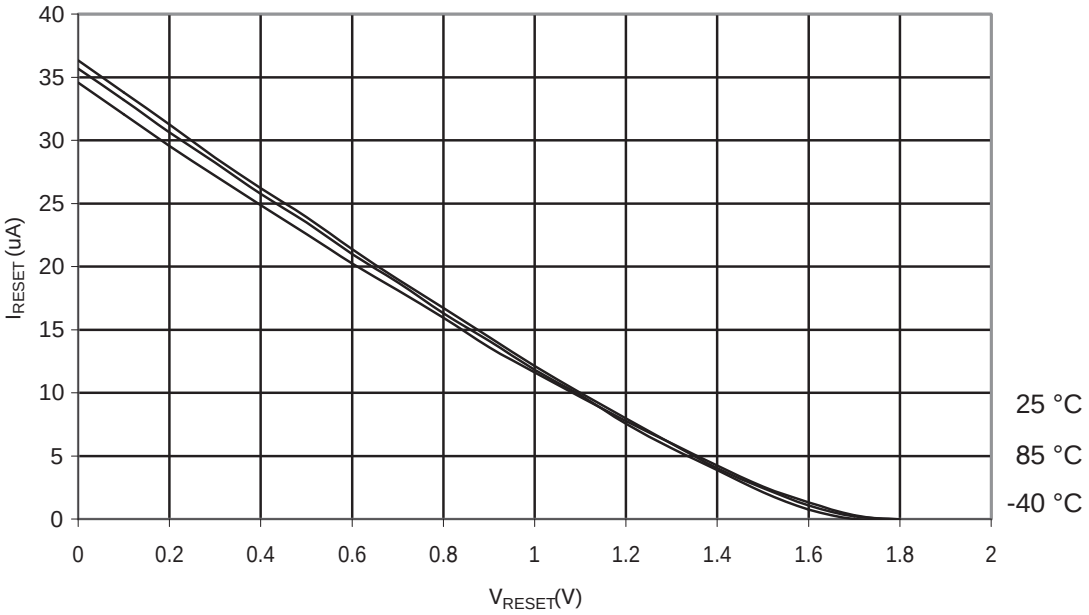


Figure 29-157. ATmega328P: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 2.7 V)

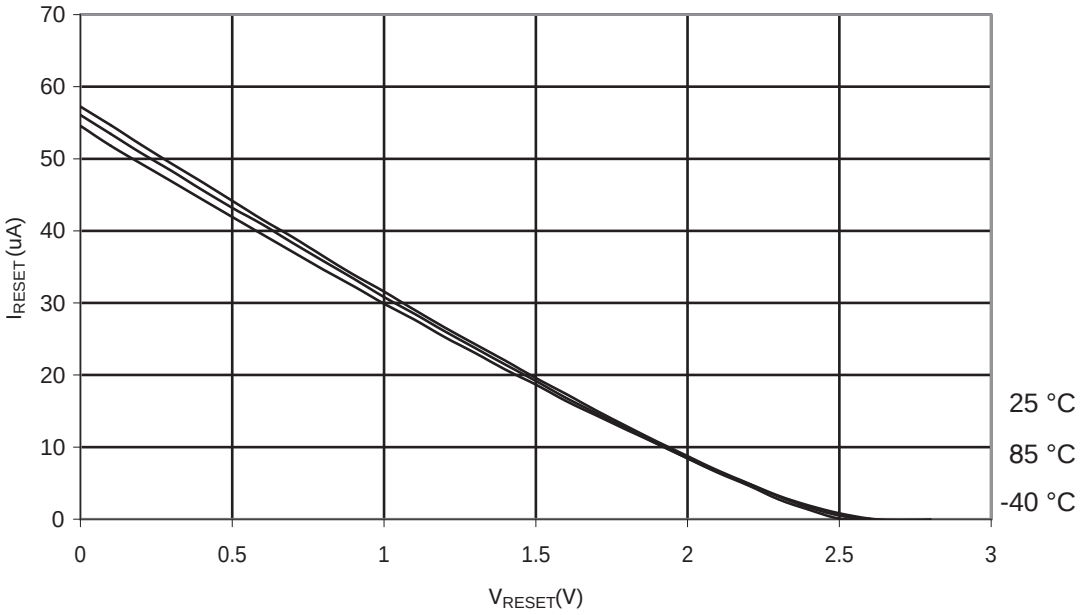
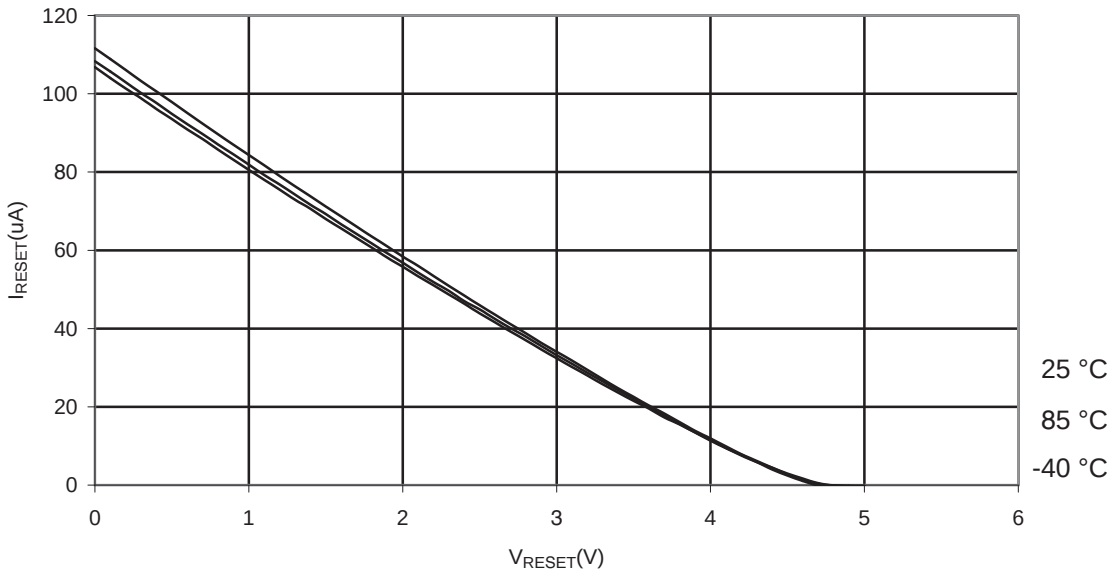


Figure 29-158. ATmega328P: Reset Pull-up Resistor Current vs. Reset Pin Voltage (V_{CC} = 5 V)



29.4.8 Pin Driver Strength

Figure 29-159. ATmega328P: I/O Pin Output Voltage vs. Sink Current (V_{CC} = 3 V)

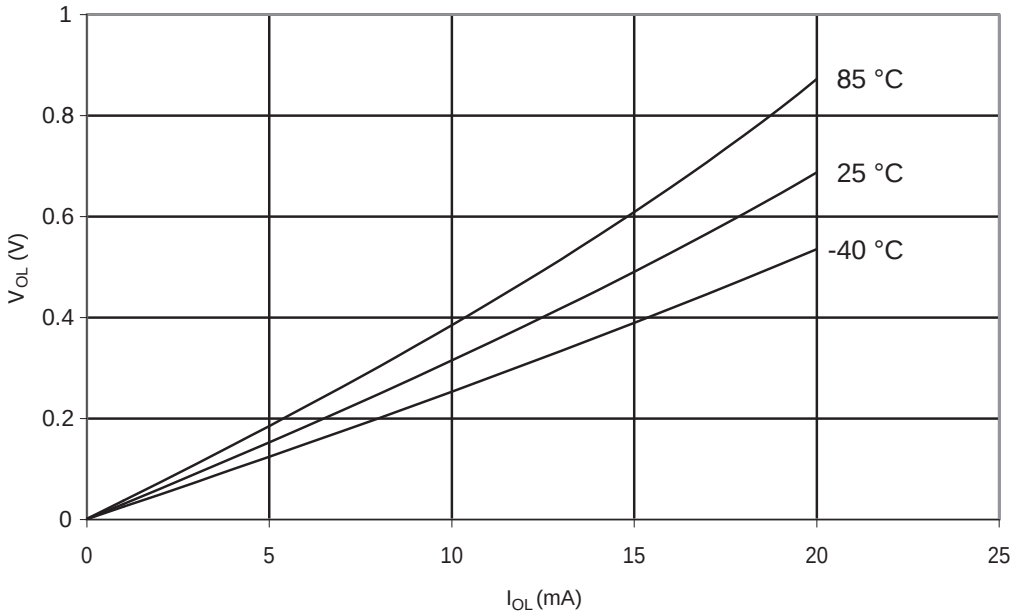


Figure 29-160. ATmega328P: I/O Pin Output Voltage vs. Sink Current (V_{CC} = 5 V)

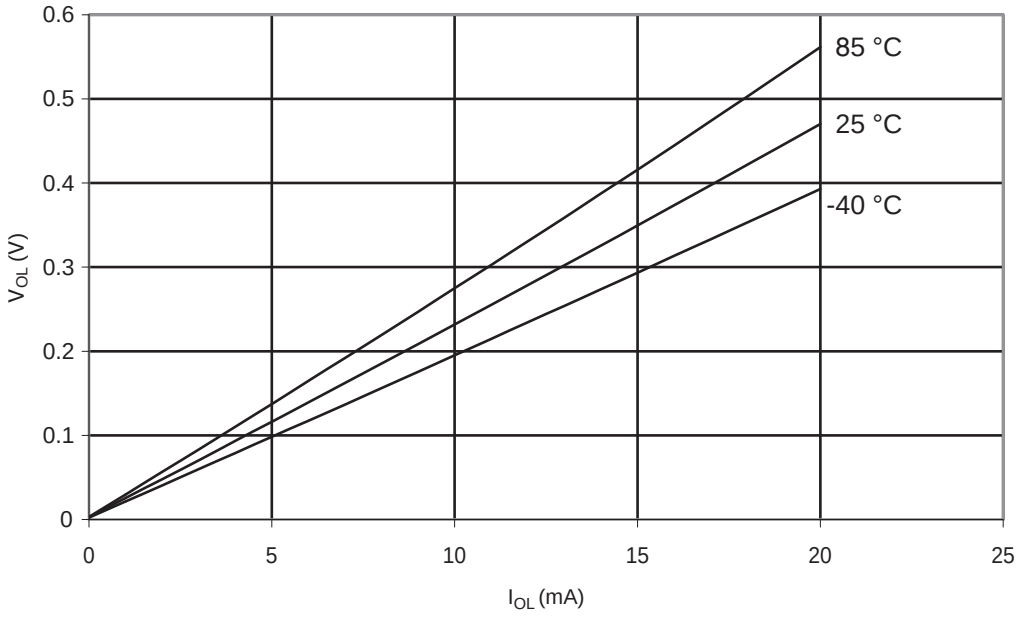


Figure 29-161. ATmega328P: I/O Pin Output Voltage vs. Source Current (Vcc = 3 V)

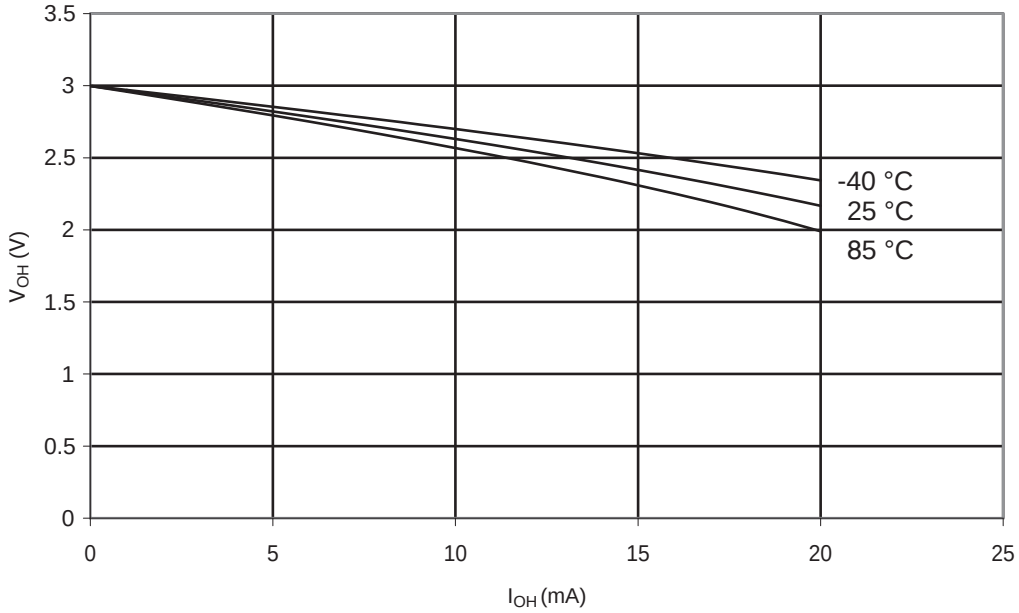
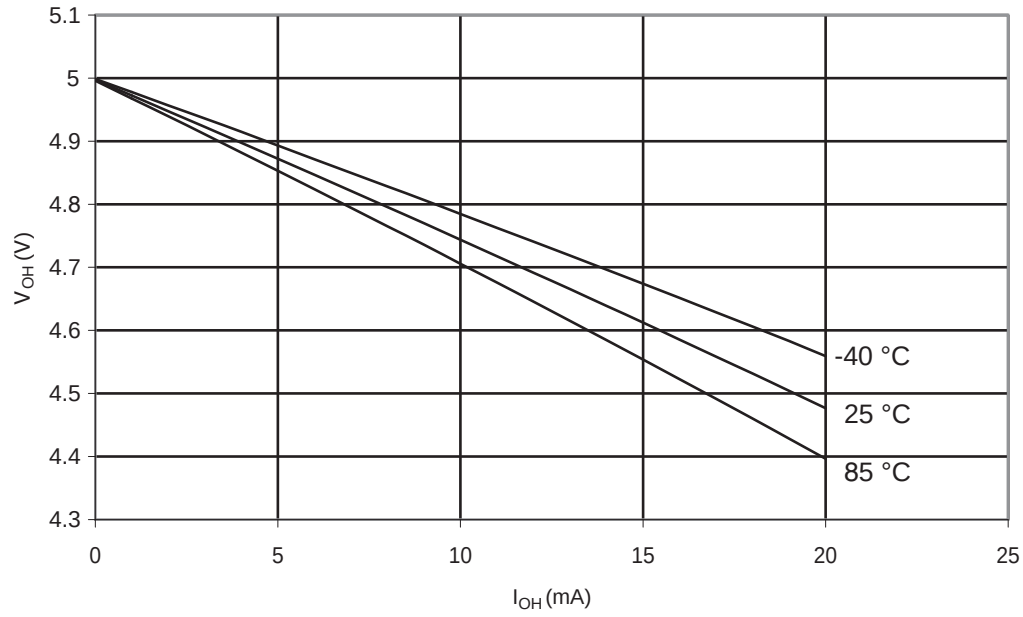


Figure 29-162. ATmega328P: I/O Pin Output Voltage vs. Source Current(V_{CC} = 5 V)



29.4.9 Pin Threshold and Hysteresis

Figure 29-163. ATmega328P: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

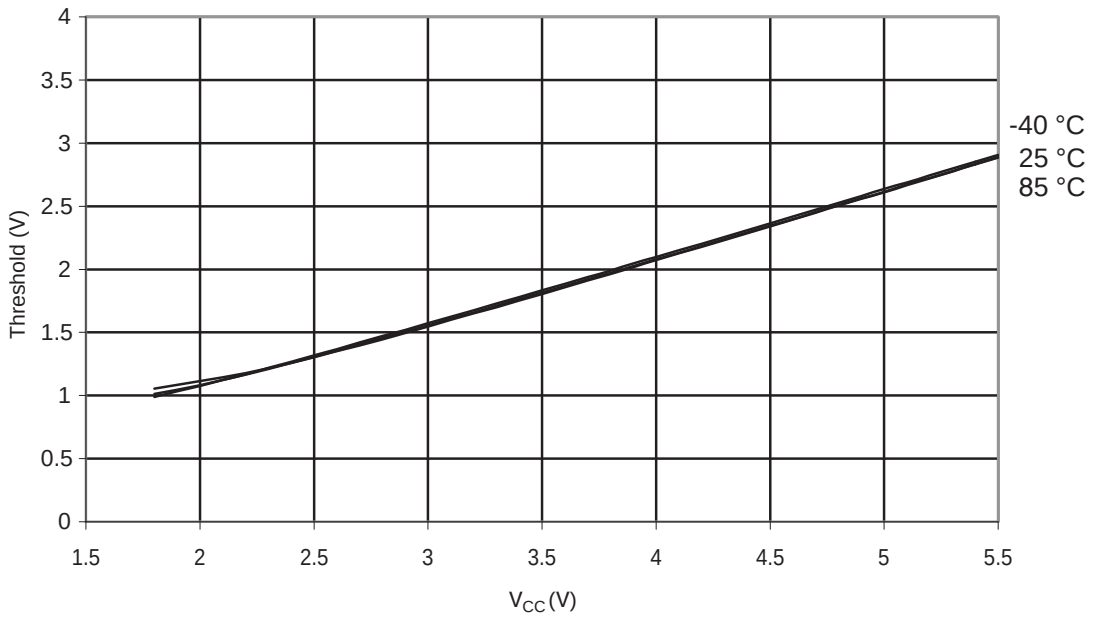


Figure 29-164. ATmega328P: I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

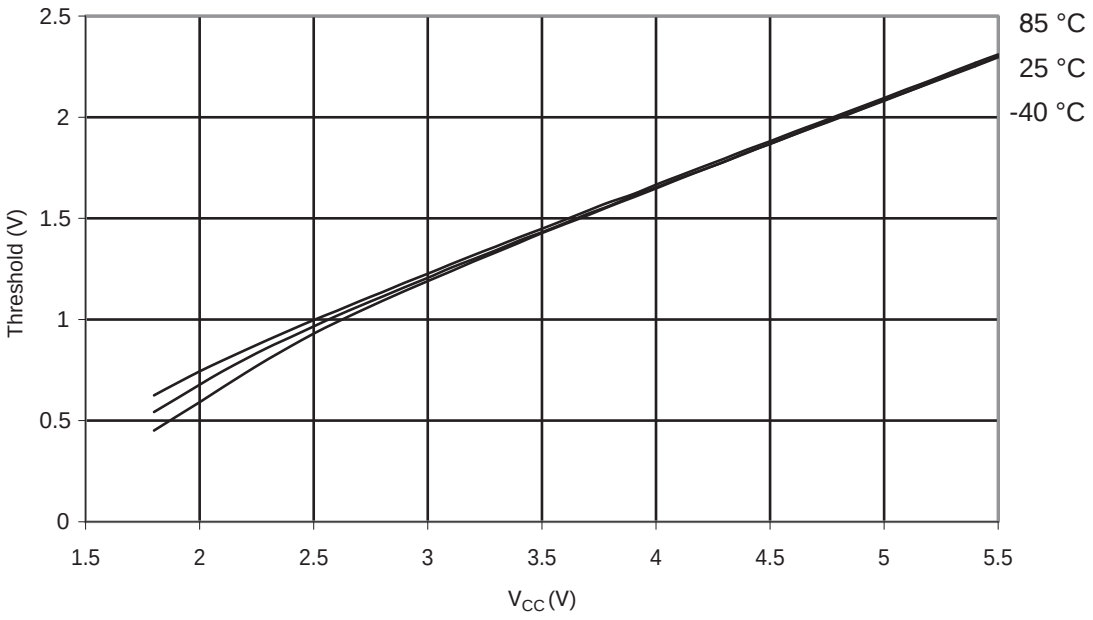


Figure 29-165. ATmega328P: I/O Pin Input Hysteresis vs. V_{CC}

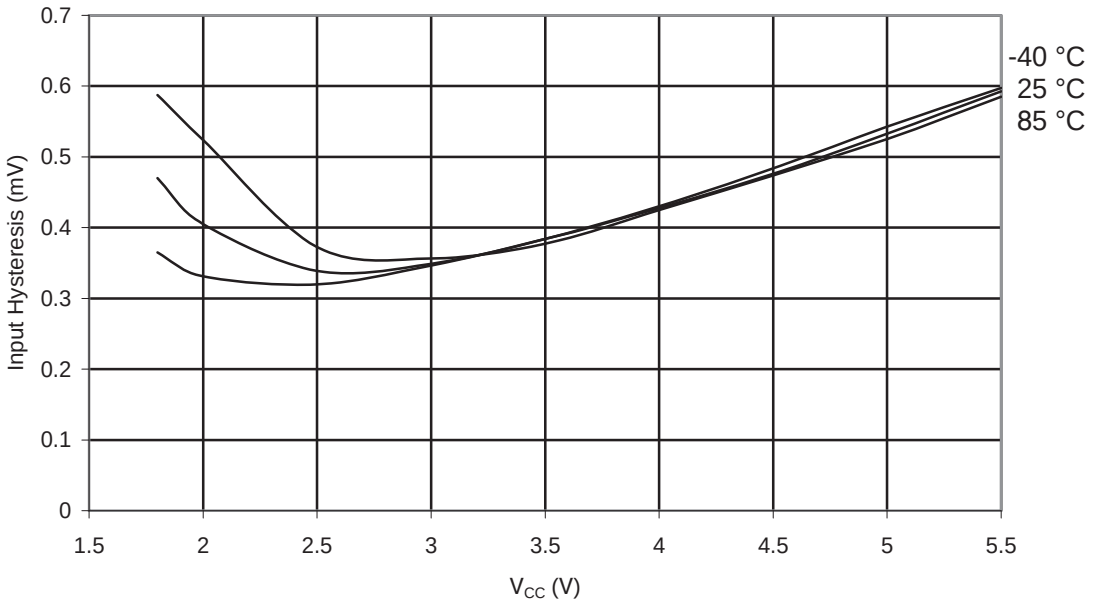


Figure 29-166. ATmega328P: Reset Input Threshold Voltage vs. V_{CC} (V_{IH}, I/O Pin read as '1')

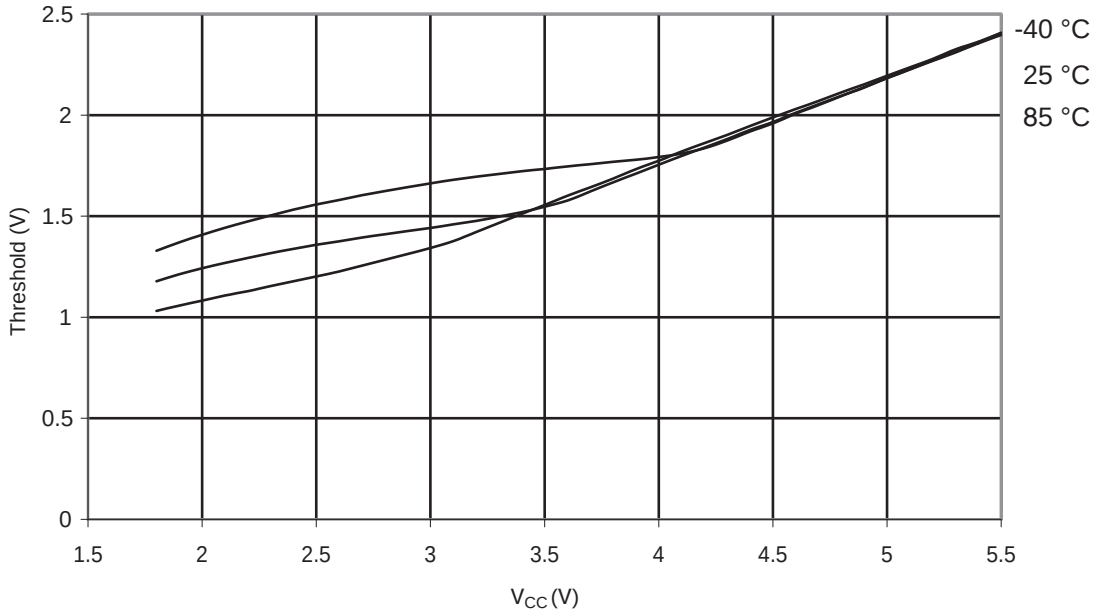


Figure 29-167. ATmega328P: Reset Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin read as '0')

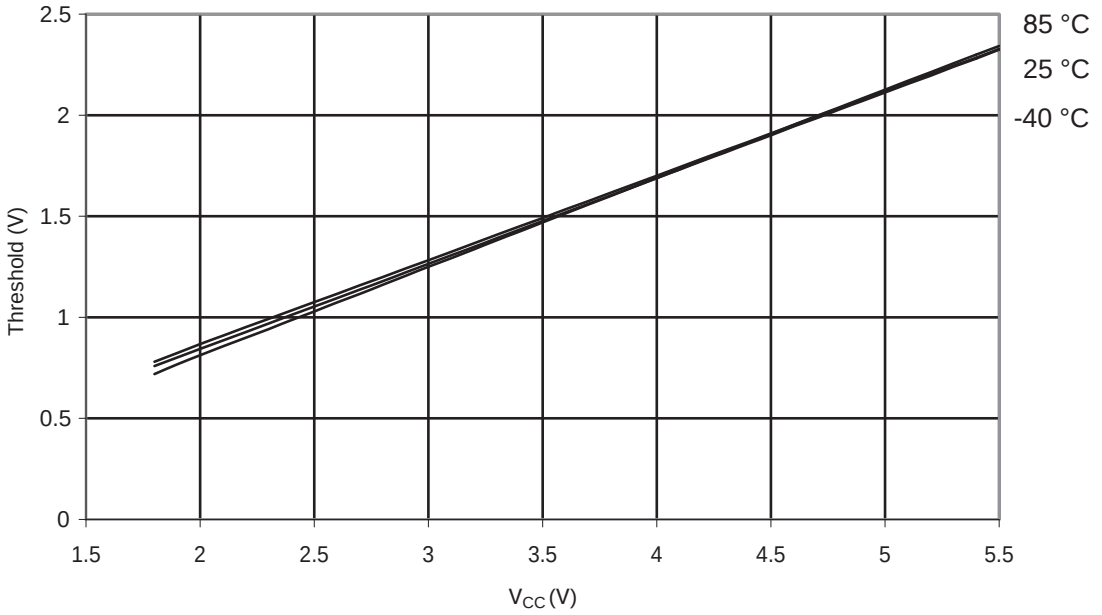
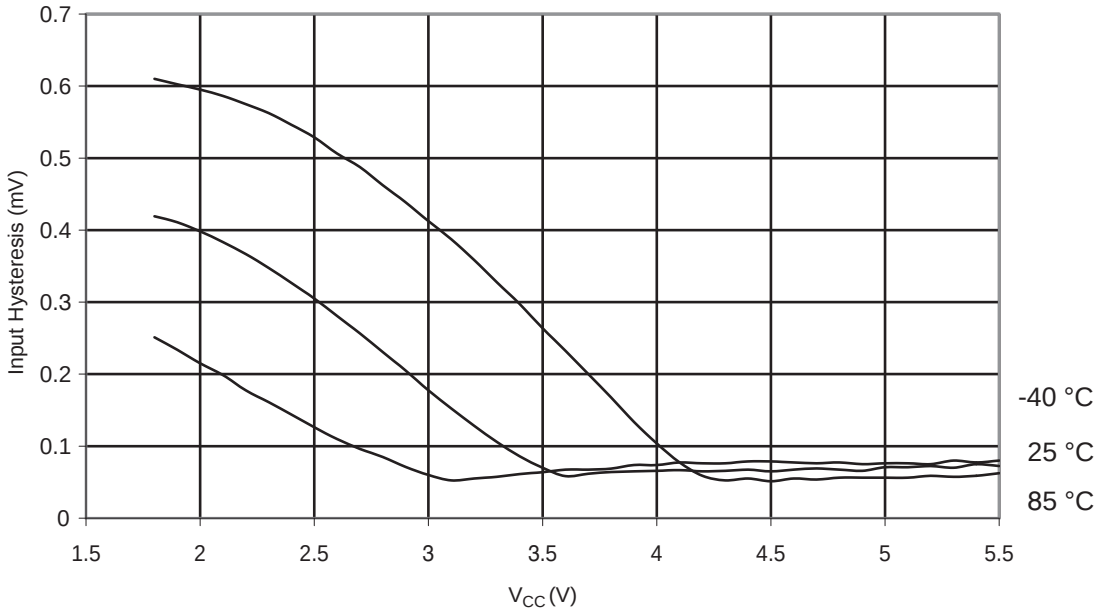


Figure 29-168. ATmega328P: Reset Pin Input Hysteresis vs. V_{CC}



29.4.10 BOD Threshold

Figure 29-169. ATmega328P: BOD Thresholds vs. Temperature (BODLEVEL is 1.8 V)

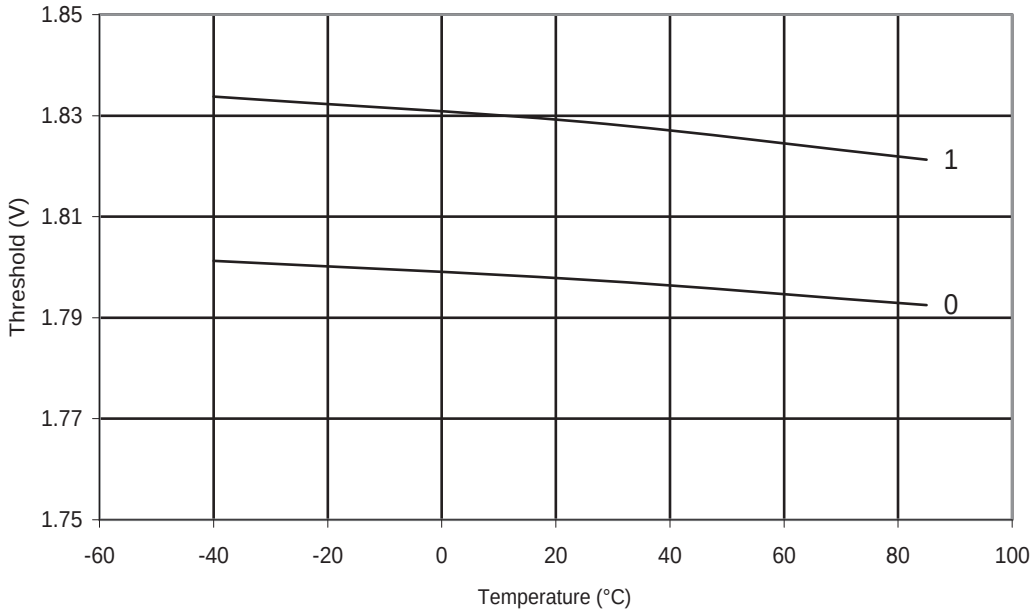


Figure 29-170. ATmega328P: BOD Thresholds vs. Temperature (BODLEVEL is 2.7 V)

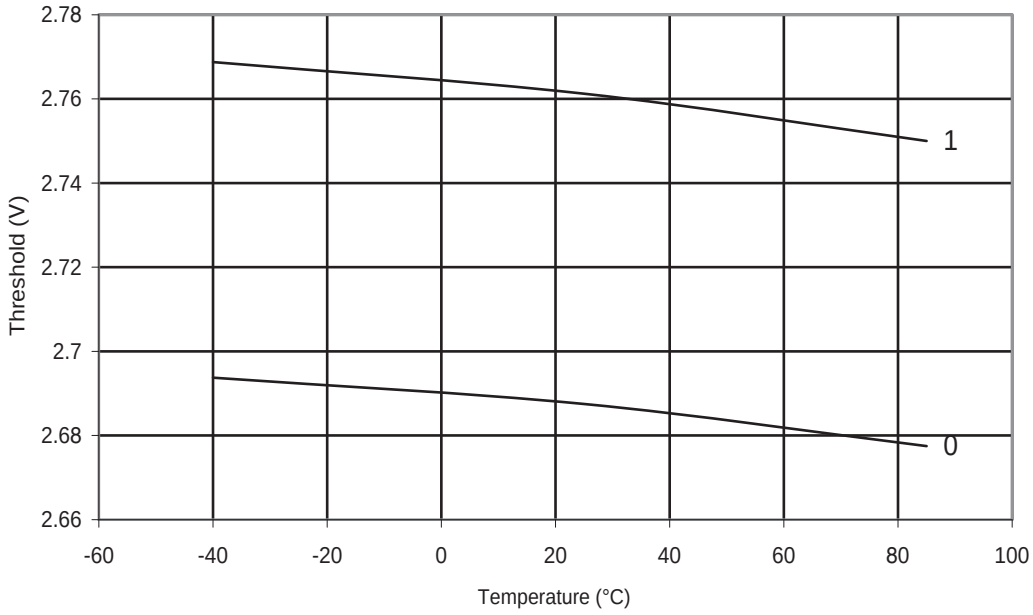
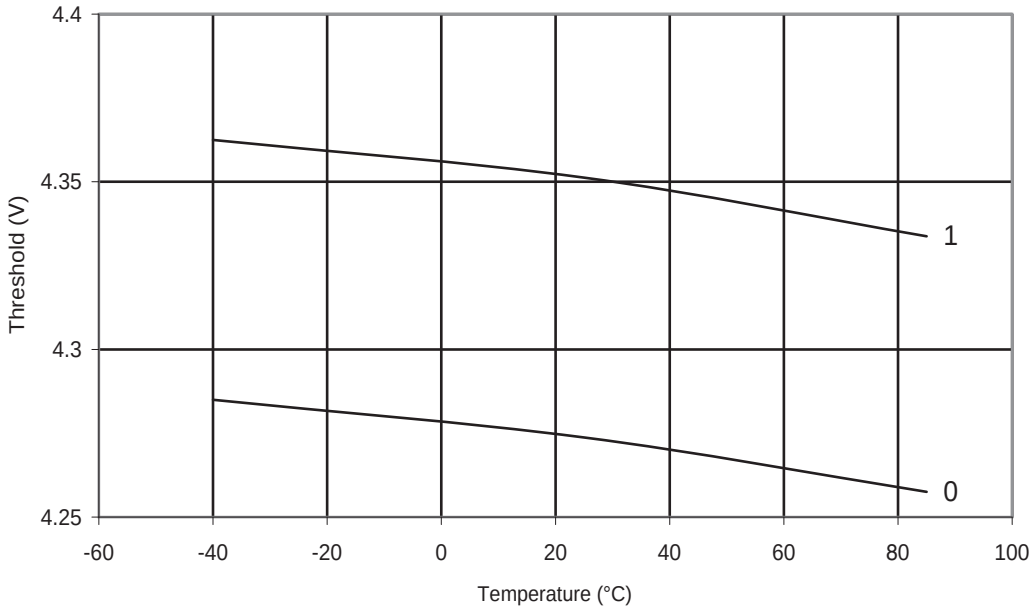


Figure 29-171.ATmega328P: BOD Thresholds vs. Temperature (BODLEVEL is 4.3 V)



29.4.11 Internal Oscilllator Speed

Figure 29-172.ATmega328P: Watchdog Oscillator Frequency vs. Temperature

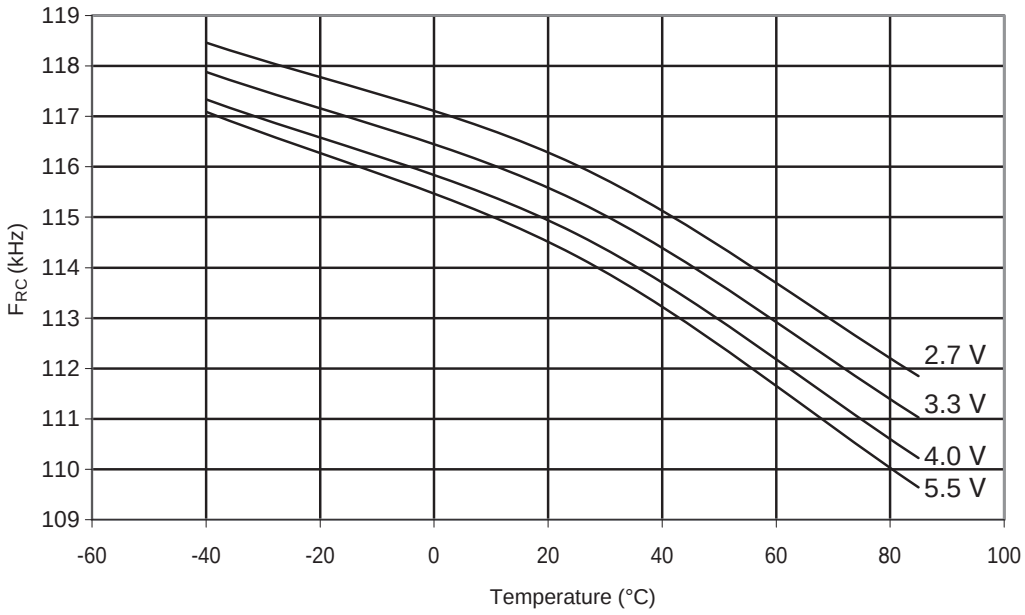


Figure 29-173. ATmega328P: Watchdog Oscillator Frequency vs. V_{CC}

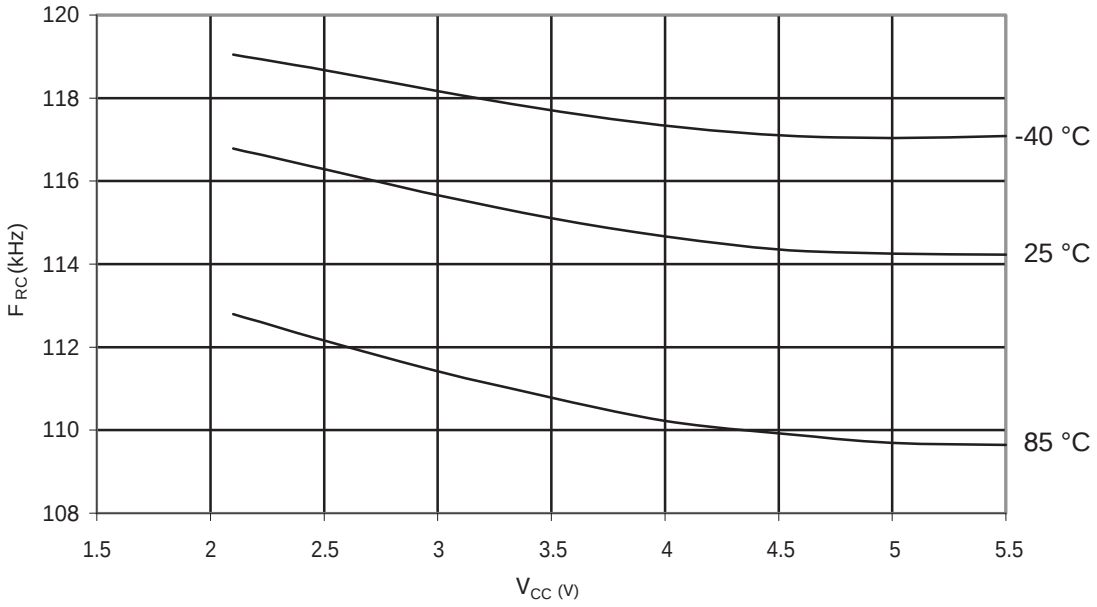


Figure 29-174. ATmega328P: Calibrated 8 MHz RC Oscillator Frequency vs. V_{CC}

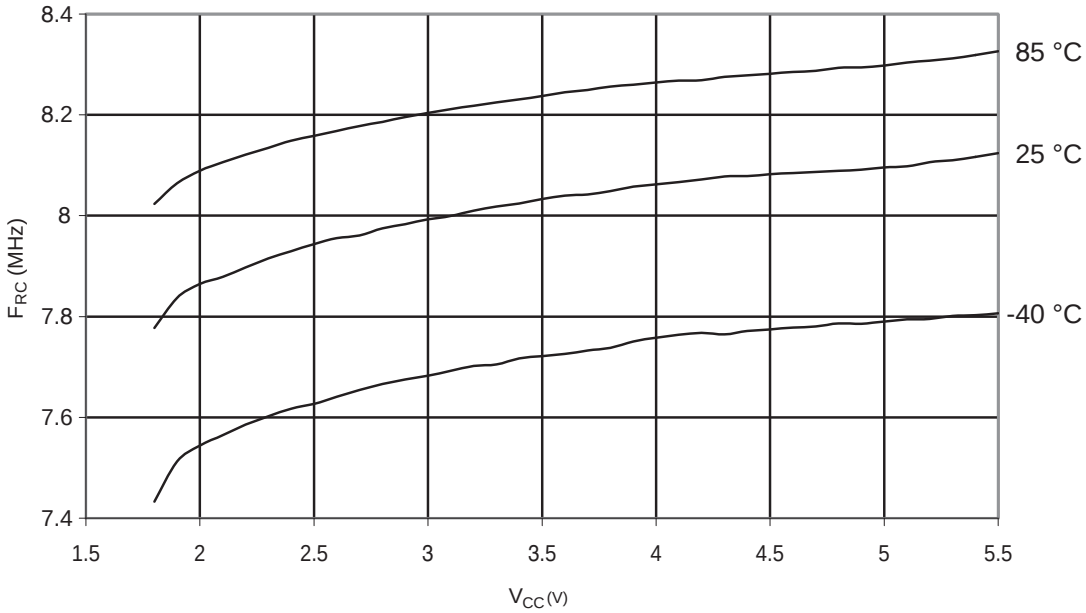


Figure 29-175. ATmega328P: Calibrated 8 MHz RC Oscillator Frequency vs. Temperature

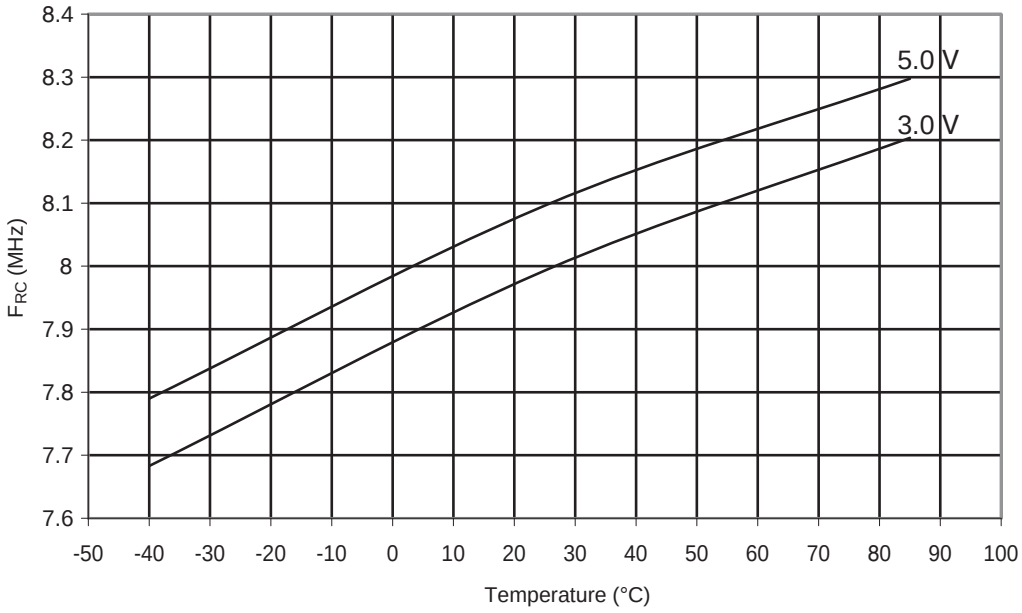
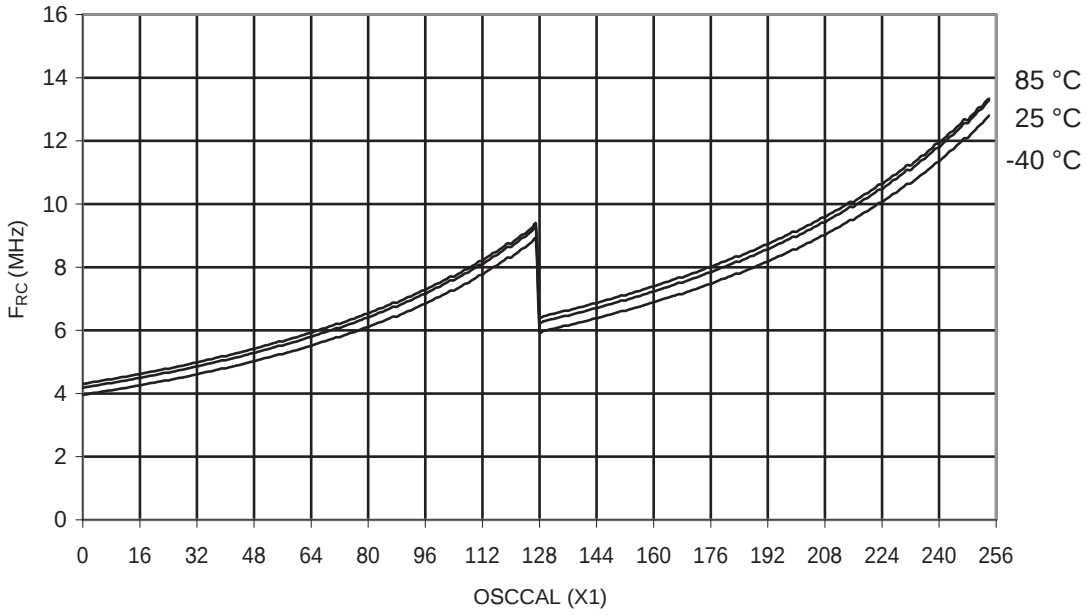


Figure 29-176. ATmega328P: Calibrated 8 MHz RC Oscillator Frequency vs. OSCCAL Value



29.4.12 Current Consumption of Peripheral Units

Figure 29-177. ATmega328P: ADC Current vs. V_{CC} ($AREF = AV_{CC}$)

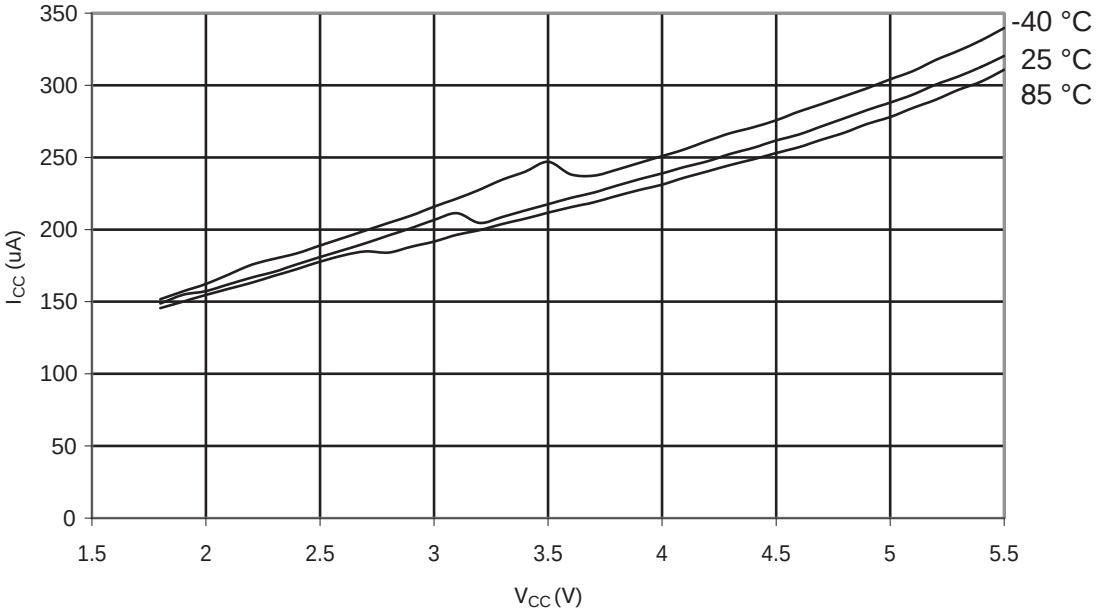


Figure 29-178. ATmega328P: Analog Comparator Current vs. V_{CC}

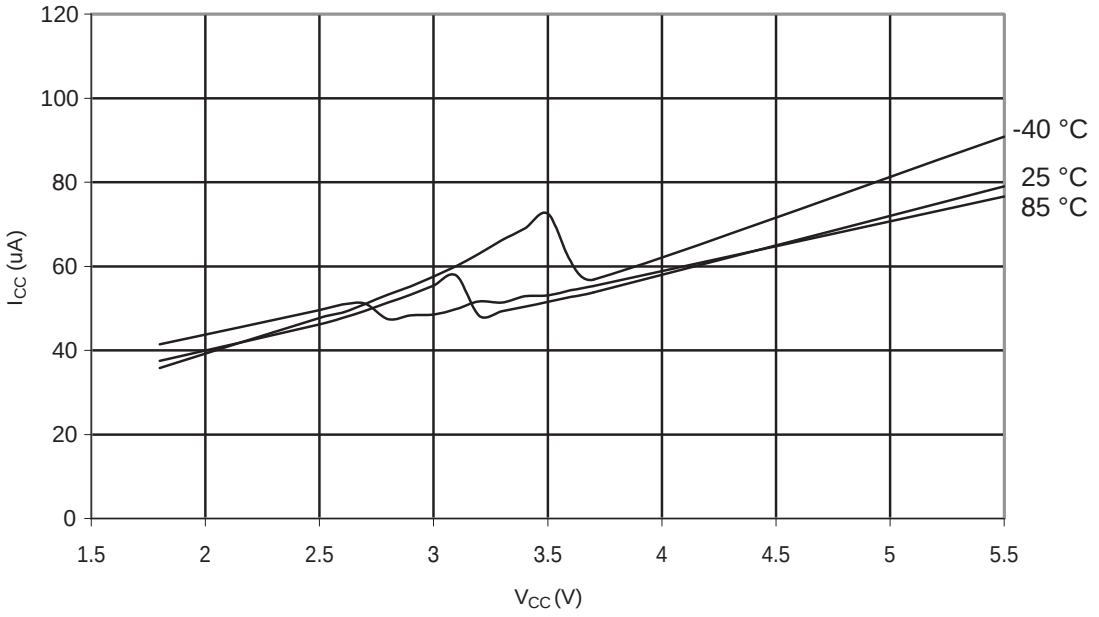


Figure 29-179. ATmega328P: AREF External Reference Current vs. V_{CC}

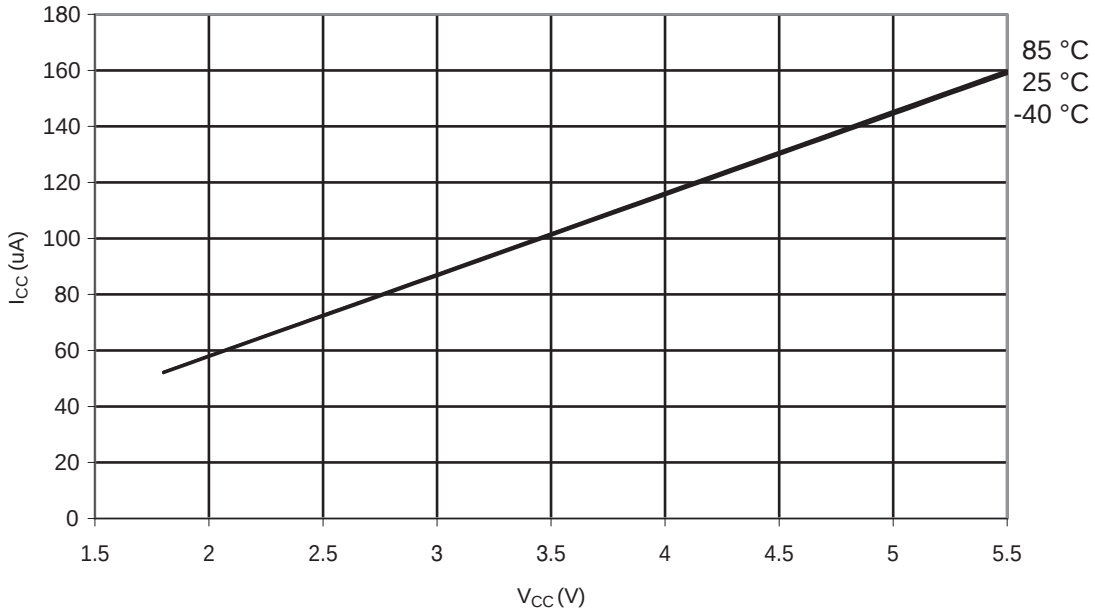


Figure 29-180. ATmega328P: Brownout Detector Current vs. V_{CC}

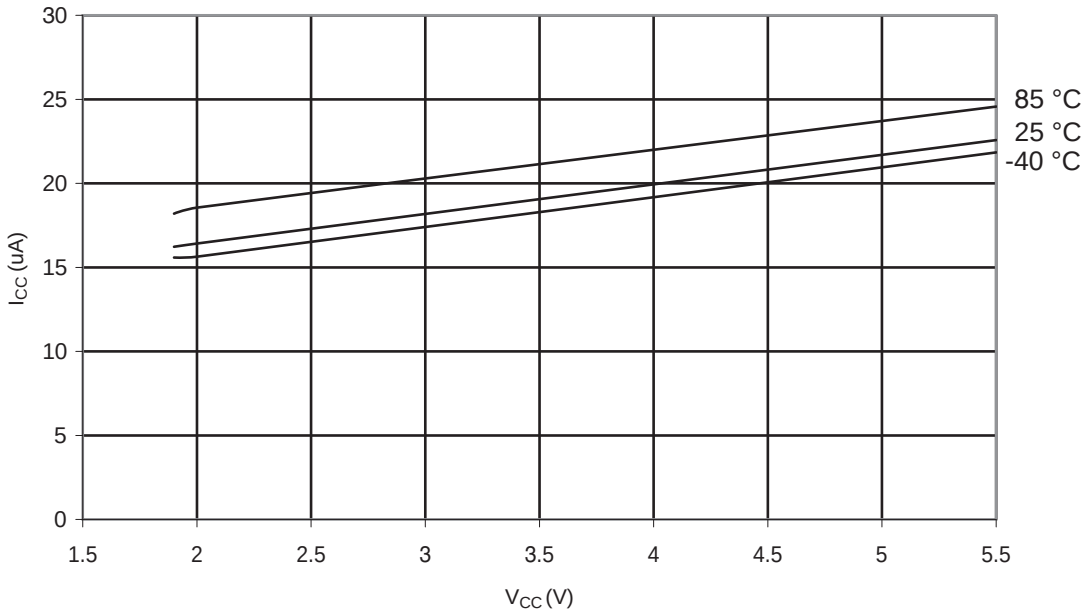
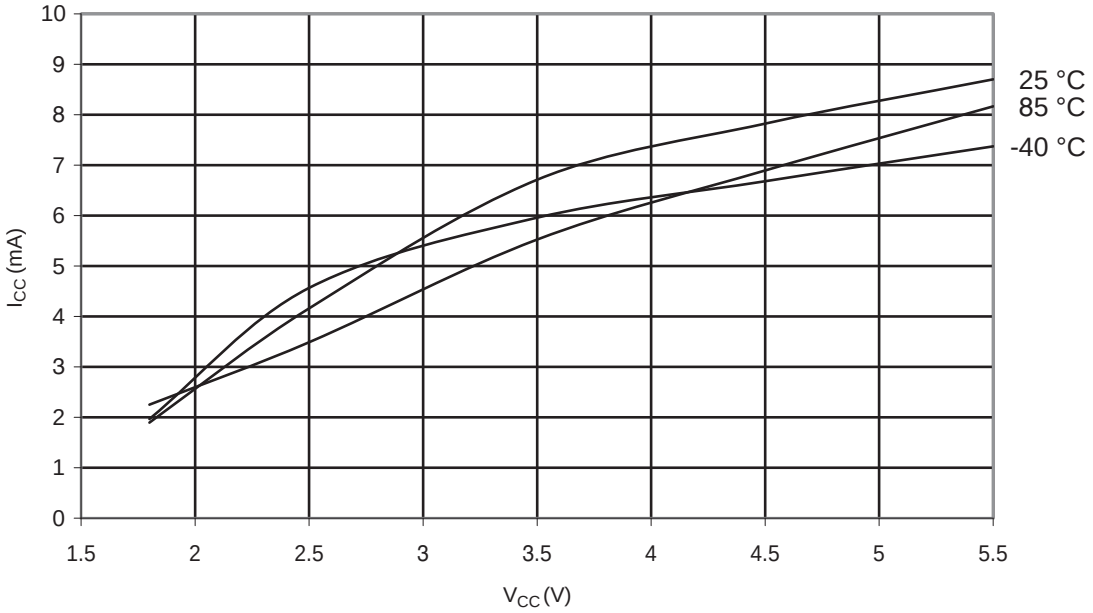


Figure 29-181.ATmega328P: Programming Current vs. V_{CC}



29.4.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-182.ATmega328P: Reset Supply Current vs. Low Frequency (0.1 - 1.0 MHz)

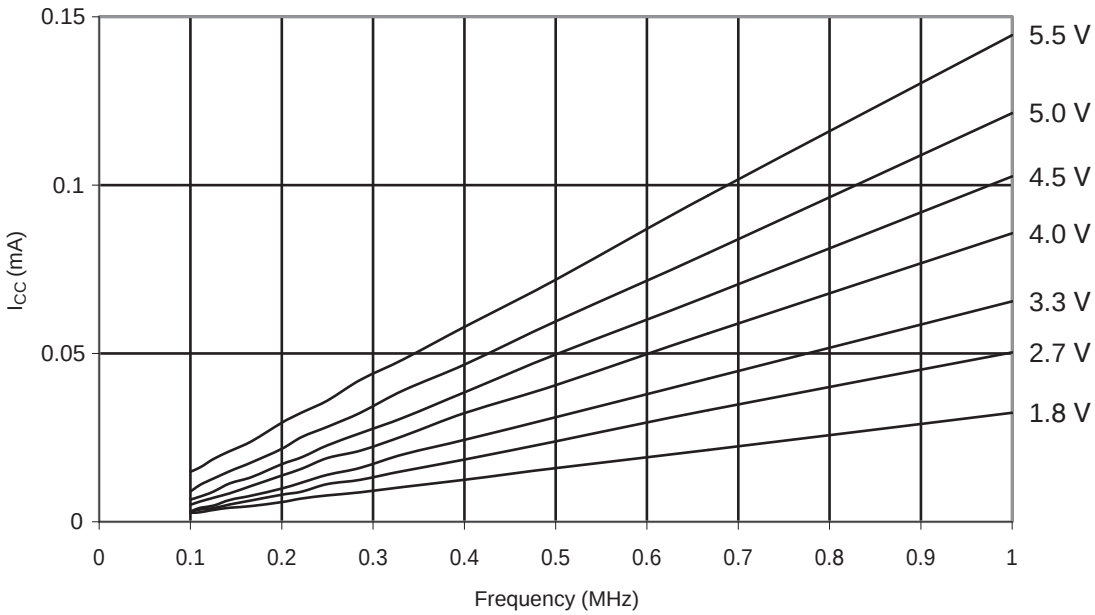


Figure 29-183. ATmega328P: Reset Supply Current vs. Frequency (1 - 20 MHz)

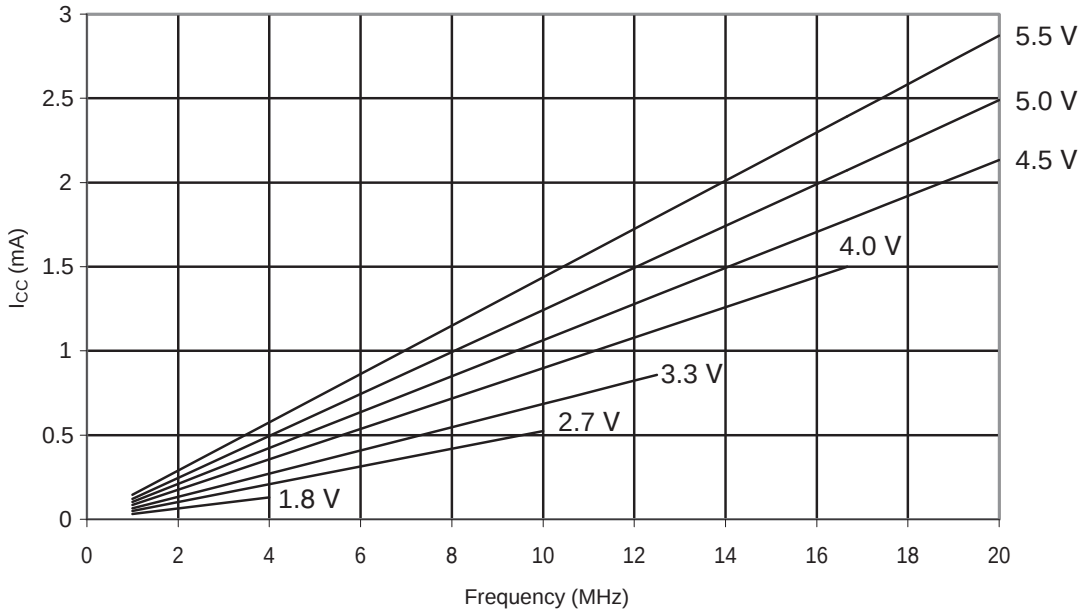
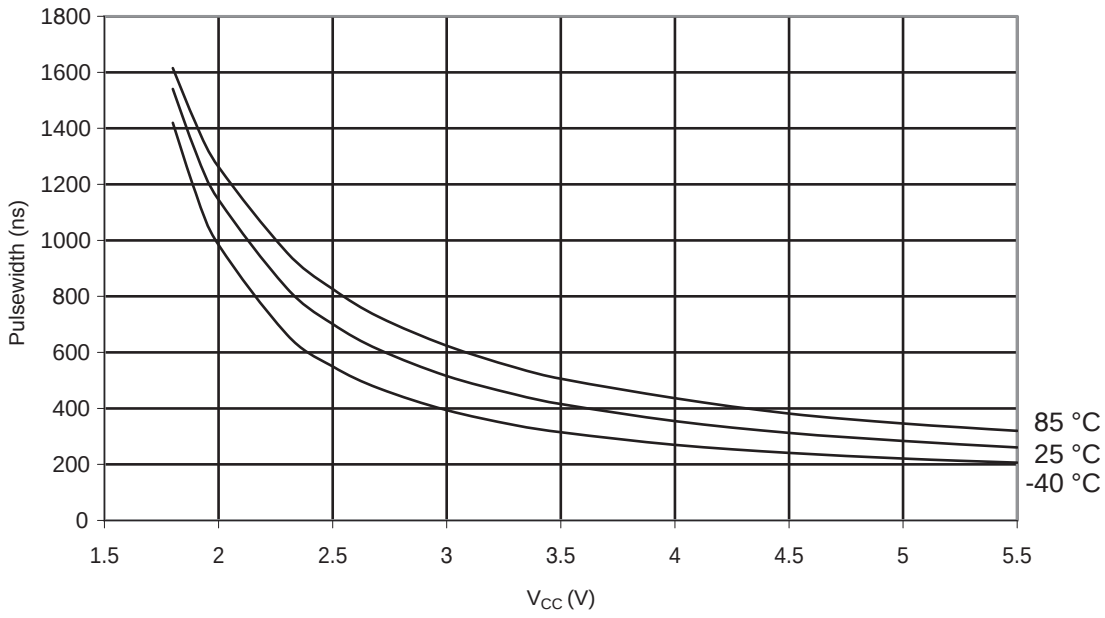


Figure 29-184. ATmega328P: Minimum Reset Pulse width vs. V_{CC}



30. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xFF)	Reserved	–	–	–	–	–	–	–	–	
(0xFE)	Reserved	–	–	–	–	–	–	–	–	
(0xFD)	Reserved	–	–	–	–	–	–	–	–	
(0xFC)	Reserved	–	–	–	–	–	–	–	–	
(0xFB)	Reserved	–	–	–	–	–	–	–	–	
(0xFA)	Reserved	–	–	–	–	–	–	–	–	
(0xF9)	Reserved	–	–	–	–	–	–	–	–	
(0xF8)	Reserved	–	–	–	–	–	–	–	–	
(0xF7)	Reserved	–	–	–	–	–	–	–	–	
(0xF6)	Reserved	–	–	–	–	–	–	–	–	
(0xF5)	Reserved	–	–	–	–	–	–	–	–	
(0xF4)	Reserved	–	–	–	–	–	–	–	–	
(0xF3)	Reserved	–	–	–	–	–	–	–	–	
(0xF2)	Reserved	–	–	–	–	–	–	–	–	
(0xF1)	Reserved	–	–	–	–	–	–	–	–	
(0xF0)	Reserved	–	–	–	–	–	–	–	–	
(0xEF)	Reserved	–	–	–	–	–	–	–	–	
(0xEE)	Reserved	–	–	–	–	–	–	–	–	
(0xED)	Reserved	–	–	–	–	–	–	–	–	
(0xEC)	Reserved	–	–	–	–	–	–	–	–	
(0xEB)	Reserved	–	–	–	–	–	–	–	–	
(0xEA)	Reserved	–	–	–	–	–	–	–	–	
(0xE9)	Reserved	–	–	–	–	–	–	–	–	
(0xE8)	Reserved	–	–	–	–	–	–	–	–	
(0xE7)	Reserved	–	–	–	–	–	–	–	–	
(0xE6)	Reserved	–	–	–	–	–	–	–	–	
(0xE5)	Reserved	–	–	–	–	–	–	–	–	
(0xE4)	Reserved	–	–	–	–	–	–	–	–	
(0xE3)	Reserved	–	–	–	–	–	–	–	–	
(0xE2)	Reserved	–	–	–	–	–	–	–	–	
(0xE1)	Reserved	–	–	–	–	–	–	–	–	
(0xE0)	Reserved	–	–	–	–	–	–	–	–	
(0xDF)	Reserved	–	–	–	–	–	–	–	–	
(0xDE)	Reserved	–	–	–	–	–	–	–	–	
(0xDD)	Reserved	–	–	–	–	–	–	–	–	
(0xDC)	Reserved	–	–	–	–	–	–	–	–	
(0xDB)	Reserved	–	–	–	–	–	–	–	–	
(0xDA)	Reserved	–	–	–	–	–	–	–	–	
(0xD9)	Reserved	–	–	–	–	–	–	–	–	
(0xD8)	Reserved	–	–	–	–	–	–	–	–	
(0xD7)	Reserved	–	–	–	–	–	–	–	–	
(0xD6)	Reserved	–	–	–	–	–	–	–	–	
(0xD5)	Reserved	–	–	–	–	–	–	–	–	
(0xD4)	Reserved	–	–	–	–	–	–	–	–	
(0xD3)	Reserved	–	–	–	–	–	–	–	–	
(0xD2)	Reserved	–	–	–	–	–	–	–	–	
(0xD1)	Reserved	–	–	–	–	–	–	–	–	
(0xD0)	Reserved	–	–	–	–	–	–	–	–	
(0xCF)	Reserved	–	–	–	–	–	–	–	–	
(0xCE)	Reserved	–	–	–	–	–	–	–	–	
(0xCD)	Reserved	–	–	–	–	–	–	–	–	
(0xCC)	Reserved	–	–	–	–	–	–	–	–	
(0xCB)	Reserved	–	–	–	–	–	–	–	–	
(0xCA)	Reserved	–	–	–	–	–	–	–	–	
(0xC9)	Reserved	–	–	–	–	–	–	–	–	
(0xC8)	Reserved	–	–	–	–	–	–	–	–	
(0xC7)	Reserved	–	–	–	–	–	–	–	–	
(0xC6)	UDR0	USART I/O Data Register								195
(0xC5)	UBRR0H					USART Baud Rate Register High				199
(0xC4)	UBRR0L	USART Baud Rate Register Low								199
(0xC3)	Reserved	–	–	–	–	–	–	–	–	
(0xC2)	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01 /UDORD0	UCSZ00 / UCPHA0	UCPOL0	197/212
(0xC1)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	196
(0xC0)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	195

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xBF)	Reserved	–	–	–	–	–	–	–	–	
(0xBE)	Reserved	–	–	–	–	–	–	–	–	
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	–	244
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	241
(0xBB)	TWDR	2-wire Serial Interface Data Register								243
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	244
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	243
(0xB8)	TWBR	2-wire Serial Interface Bit Rate Register								241
(0xB7)	Reserved	–		–	–	–	–	–	–	
(0xB6)	ASSR	–	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	164
(0xB5)	Reserved	–	–	–	–	–	–	–	–	
(0xB4)	OCR2B	Timer/Counter2 Output Compare Register B								162
(0xB3)	OCR2A	Timer/Counter2 Output Compare Register A								162
(0xB2)	TCNT2	Timer/Counter2 (8-bit)								162
(0xB1)	TCCR2B	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	161
(0xB0)	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	158
(0xAF)	Reserved	–	–	–	–	–	–	–	–	
(0xAE)	Reserved	–	–	–	–	–	–	–	–	
(0xAD)	Reserved	–	–	–	–	–	–	–	–	
(0xAC)	Reserved	–	–	–	–	–	–	–	–	
(0xAB)	Reserved	–	–	–	–	–	–	–	–	
(0xAA)	Reserved	–	–	–	–	–	–	–	–	
(0xA9)	Reserved	–	–	–	–	–	–	–	–	
(0xA8)	Reserved	–	–	–	–	–	–	–	–	
(0xA7)	Reserved	–	–	–	–	–	–	–	–	
(0xA6)	Reserved	–	–	–	–	–	–	–	–	
(0xA5)	Reserved	–	–	–	–	–	–	–	–	
(0xA4)	Reserved	–	–	–	–	–	–	–	–	
(0xA3)	Reserved	–	–	–	–	–	–	–	–	
(0xA2)	Reserved	–	–	–	–	–	–	–	–	
(0xA1)	Reserved	–	–	–	–	–	–	–	–	
(0xA0)	Reserved	–	–	–	–	–	–	–	–	
(0x9F)	Reserved	–	–	–	–	–	–	–	–	
(0x9E)	Reserved	–	–	–	–	–	–	–	–	
(0x9D)	Reserved	–	–	–	–	–	–	–	–	
(0x9C)	Reserved	–	–	–	–	–	–	–	–	
(0x9B)	Reserved	–	–	–	–	–	–	–	–	
(0x9A)	Reserved	–	–	–	–	–	–	–	–	
(0x99)	Reserved	–	–	–	–	–	–	–	–	
(0x98)	Reserved	–	–	–	–	–	–	–	–	
(0x97)	Reserved	–	–	–	–	–	–	–	–	
(0x96)	Reserved	–	–	–	–	–	–	–	–	
(0x95)	Reserved	–	–	–	–	–	–	–	–	
(0x94)	Reserved	–	–	–	–	–	–	–	–	
(0x93)	Reserved	–	–	–	–	–	–	–	–	
(0x92)	Reserved	–	–	–	–	–	–	–	–	
(0x91)	Reserved	–	–	–	–	–	–	–	–	
(0x90)	Reserved	–	–	–	–	–	–	–	–	
(0x8F)	Reserved	–	–	–	–	–	–	–	–	
(0x8E)	Reserved	–	–	–	–	–	–	–	–	
(0x8D)	Reserved	–	–	–	–	–	–	–	–	
(0x8C)	Reserved	–	–	–	–	–	–	–	–	
(0x8B)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								138
(0x8A)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								138
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								138
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								138
(0x87)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								138
(0x86)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								138
(0x85)	TCNT1H	Timer/Counter1 - Counter Register High Byte								138
(0x84)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								138
(0x83)	Reserved	–	–	–	–	–	–	–	–	
(0x82)	TCCR1C	FOC1A	FOC1B	–	–	–	–	–	–	137
(0x81)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	136
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	134
(0x7F)	DIDR1	–	–	–	–	–	–	AIN1D	AIN0D	249
(0x7E)	DIDR0	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	266

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7D)	Reserved	–	–	–	–	–	–	–	–	
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	262
(0x7B)	ADCSRB	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	265
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	263
(0x79)	ADCH	ADC Data Register High byte								265
(0x78)	ADCL	ADC Data Register Low byte								265
(0x77)	Reserved	–	–	–	–	–	–	–	–	
(0x76)	Reserved	–	–	–	–	–	–	–	–	
(0x75)	Reserved	–	–	–	–	–	–	–	–	
(0x74)	Reserved	–	–	–	–	–	–	–	–	
(0x73)	Reserved	–	–	–	–	–	–	–	–	
(0x72)	Reserved	–	–	–	–	–	–	–	–	
(0x71)	Reserved	–	–	–	–	–	–	–	–	
(0x70)	TIMSK2	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	163
(0x6F)	TIMSK1	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	139
(0x6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	111
(0x6D)	PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	74
(0x6C)	PCMSK1	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	74
(0x6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	74
(0x6A)	Reserved	–	–	–	–	–	–	–	–	
(0x69)	EICRA	–	–	–	–	ISC11	ISC10	ISC01	ISC00	71
(0x68)	PCICR	–	–	–	–	–	PCIE2	PCIE1	PCIE0	
(0x67)	Reserved	–	–	–	–	–	–	–	–	
(0x66)	OSCCAL	Oscillator Calibration Register								37
(0x65)	Reserved	–	–	–	–	–	–	–	–	
(0x64)	PRR	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC	42
(0x63)	Reserved	–	–	–	–	–	–	–	–	
(0x62)	Reserved	–	–	–	–	–	–	–	–	
(0x61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	37
(0x60)	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	54
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	9
0x3E (0x5E)	SPH	–	–	–	–	–	(SP10) [5.]	SP9	SP8	12
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
0x3C (0x5C)	Reserved	–	–	–	–	–	–	–	–	
0x3B (0x5B)	Reserved	–	–	–	–	–	–	–	–	
0x3A (0x5A)	Reserved	–	–	–	–	–	–	–	–	
0x39 (0x59)	Reserved	–	–	–	–	–	–	–	–	
0x38 (0x58)	Reserved	–	–	–	–	–	–	–	–	
0x37 (0x57)	SPMCSR	SPMIE	(RWWSB)[5.]	–	(RWWSRE)[5.]	BLBSET	PGWRT	PGERS	SELFPRGEN	292
0x36 (0x56)	Reserved	–	–	–	–	–	–	–	–	
0x35 (0x55)	MCUCR	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	44/68/92
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	54
0x33 (0x53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE	40
0x32 (0x52)	Reserved	–	–	–	–	–	–	–	–	
0x31 (0x51)	Reserved	–	–	–	–	–	–	–	–	
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	247
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–	
0x2E (0x4E)	SPDR	SPI Data Register								175
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	174
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	173
0x2B (0x4B)	GPIOR2	General Purpose I/O Register 2								25
0x2A (0x4A)	GPIOR1	General Purpose I/O Register 1								25
0x29 (0x49)	Reserved	–	–	–	–	–	–	–	–	
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B								
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A								
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)								
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	PSRASY	PSRSYNC	143/165
0x22 (0x42)	EEARH	(EEPROM Address Register High Byte) [5.]								21
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte								21
0x20 (0x40)	EEDR	EEPROM Data Register								21
0x1F (0x3F)	EECR	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	21
0x1E (0x3E)	GPIOR0	General Purpose I/O Register 0								25
0x1D (0x3D)	EIMSK	–	–	–	–	–	–	INT1	INT0	72
0x1C (0x3C)	EIFR	–	–	–	–	–	–	INTF1	INTF0	72

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1B (0x3B)	PCIFR	–	–	–	–	–	PCIF2	PCIF1	PCIF0	
0x1A (0x3A)	Reserved	–	–	–	–	–	–	–	–	
0x19 (0x39)	Reserved	–	–	–	–	–	–	–	–	
0x18 (0x38)	Reserved	–	–	–	–	–	–	–	–	
0x17 (0x37)	TIFR2	–	–	–	–	–	OCF2B	OCF2A	TOV2	163
0x16 (0x36)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	139
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–	
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–	
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–	
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–	
0x10 (0x30)	Reserved	–	–	–	–	–	–	–	–	
0x0F (0x2F)	Reserved	–	–	–	–	–	–	–	–	
0x0E (0x2E)	Reserved	–	–	–	–	–	–	–	–	
0x0D (0x2D)	Reserved	–	–	–	–	–	–	–	–	
0x0C (0x2C)	Reserved	–	–	–	–	–	–	–	–	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	93
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	93
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	93
0x08 (0x28)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	92
0x07 (0x27)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	92
0x06 (0x26)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	92
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	92
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	92
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	92
0x02 (0x22)	Reserved	–	–	–	–	–	–	–	–	
0x01 (0x21)	Reserved	–	–	–	–	–	–	–	–	
0x0 (0x20)	Reserved	–	–	–	–	–	–	–	–	

Note:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O Registers within the address range 0x00 - 0x1F 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.
3. Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48PA/88PA/168PA/328P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
5. Only valid for ATmega88PA/168PA.

31. 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 ← 0xFF – Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← 0x00 – 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 • (0xFF - 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 ← 0xFF	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[1]	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[1]	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
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
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
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

Mnemonics	Operands	Description	Operation	Flags	#Clocks
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
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

Note: 1. These instructions are only available in ATmega168PA and ATmega328P.

32. Ordering Information

32.1 ATmega48PA

Speed (MHz)	Power Supply	Ordering Code[2]	Package[1]	Operational Range
20[3]	1.8 - 5.5	ATmega48PA-AU ATmega48PA-MMH[4] ATmega48PA-MU ATmega48PA-PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)

Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
3. See "Speed Grades" on page 316.
4. NiPdAu Lead Finish.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

32.2 ATmega88PA

Speed (MHz)	Power Supply	Ordering Code[2]	Package[1]	Operational Range
20[3]	1.8 - 5.5	ATmega88PA-AU ATmega88PA-MMH[4] ATmega88PA-MU ATmega88PA-PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)

Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
3. See "Speed Grades" on page 316.
4. NiPdAu Lead Finish.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

32.3 ATmega168PA

Speed (MHz)[3]	Power Supply	Ordering Code[2]	Package[1]	Operational Range
20	1.8 - 5.5	ATmega168PA-AU ATmega168PA-MMH[4] ATmega168PA-MU ATmega168PA-PU	32A 28M1 32M1-A 28P3	Industrial (-40°C to 85°C)

Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
3. See "Speed Grades" on page 316.
4. NiPdAu Lead Finish.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28M1	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

32.4 ATmega328P

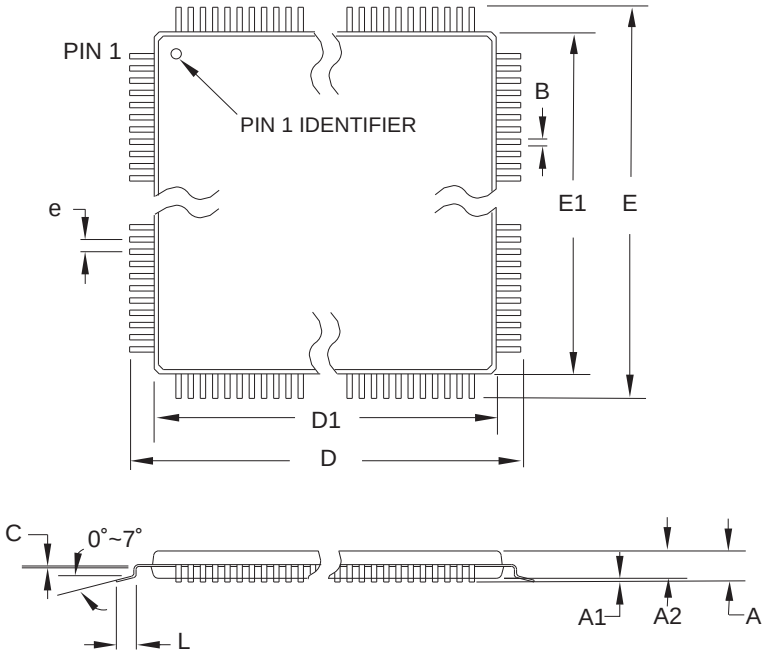
Speed (MHz)	Power Supply	Ordering Code[2]	Package[1]	Operational Range
20[3]	1.8 - 5.5	ATmega328P- AU ATmega328P- MU ATmega328P- PU	32A 32M1-A 28P3	Industrial (-40°C to 85°C)

Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.
3. See Figure 28-1 on page 316.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300” Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

33. Packaging Information

33.1 32A



COMMON DIMENSIONS
(Unit of Measure = mm)

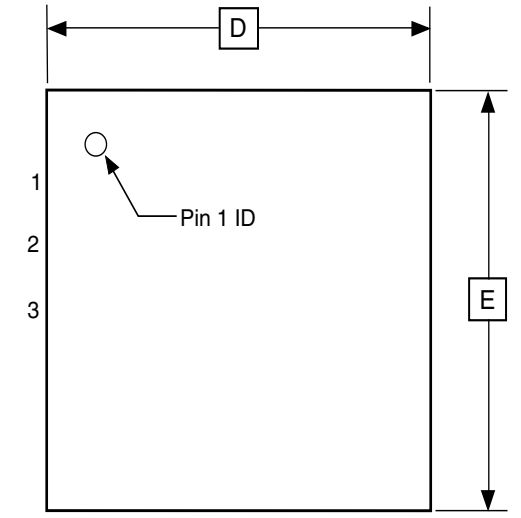
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ABA.
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.

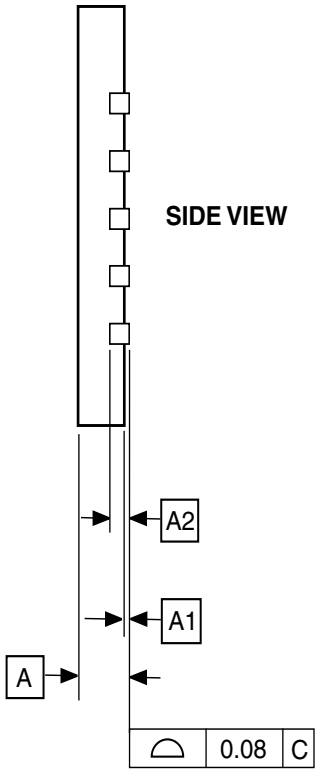
10/5/2001

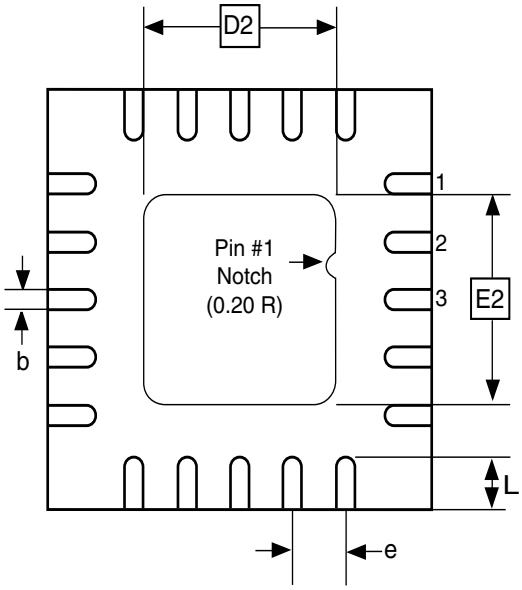
	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**32A,** 32-lead, 7 x 7 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	32A	B

33.2 28M1



TOP VIEW





BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

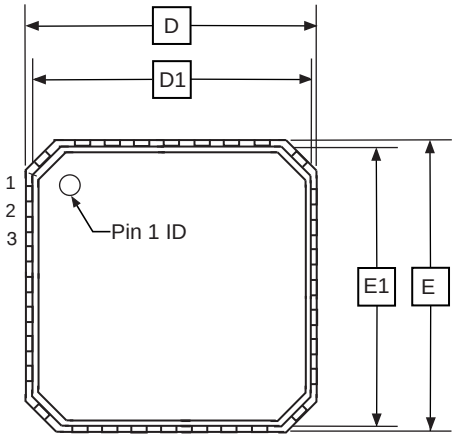
SYMBOL	MIN	NOM	MAX	NOTE
A	0.70	0.75	0.80	
A1	–	0.01	0.05	
A2		0.20 REF		
b	0.18	0.23	0.30	
D		4.00 BSC		
D2	2.45	2.60	2.75	
E		4.00 BSC		
E2	2.45	2.60	2.75	
e		0.50 BSC		
L	0.35	0.40	0.55	

Note: Reference JEDEC Standard MO-220, Fig. 1 (SAW Singulation) WGGD-5.

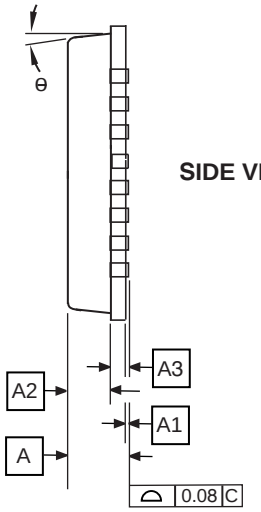
10/27/04

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway	**20M1**, 20-pad, 4 x 4 x 0.8 mm Body, Lead Pitch 0.50 mm,	20M1	A

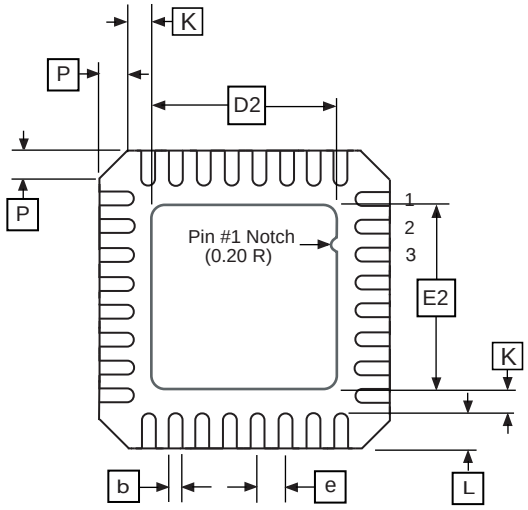
33.3 32M1-A



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

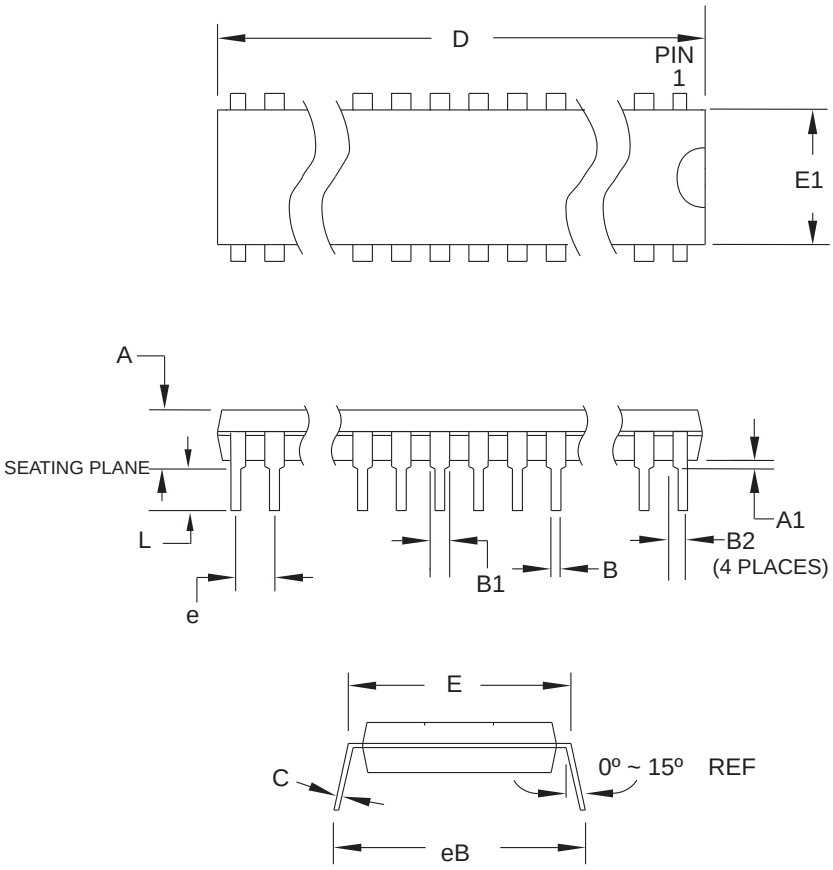
SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A2	–	0.65	1.00	
A3		0.20 REF		
b	0.18	0.23	0.30	
D	4.90	5.00	5.10	
D1	4.70	4.75	4.80	
D2	2.95	3.10	3.25	
E	4.90	5.00	5.10	
E1	4.70	4.75	4.80	
E2	2.95	3.10	3.25	
e		0.50 BSC		
L	0.30	0.40	0.50	
P	–	–	0.60	
θ	–	–	12^{o}	
K	0.20	–	–	

Note: JEDEC Standard MO-220, Fig. 2 (Anvil Singulation), VHHD-2.

5/25/06

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**32M1-A**, 32-pad, 5 x 5 x 1.0 mm Body, Lead Pitch 0.50 mm, 3.10 mm Exposed Pad, Micro Lead Frame Package (MLF)	32M1-A	E

33.4 28P3



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.5724	
A1	0.508	–	–	
D	34.544	–	34.798	Note 1
E	7.620	–	8.255	
E1	7.112	–	7.493	Note 1
B	0.381	–	0.533	
B1	1.143	–	1.397	
B2	0.762	–	1.143	
L	3.175	–	3.429	
C	0.203	–	0.356	
eB	–	–	10.160	
e	2.540 TYP			

Note: 1. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**28P3**, 28-lead (0.300"/7.62 mm Wide) Plastic Dual Inline Package (PDIP)	28P3	B

34. Errata

34.1 Errata ATmega48PA

The revision letter in this section refers to the revision of the ATmega48PA device.

34.1.1 Rev. D

No known errata.

34.2 Errata ATmega88PA

The revision letter in this section refers to the revision of the ATmega88PA device.

34.2.1 Rev. F

No known errata.

34.3 Errata ATmega168PA

The revision letter in this section refers to the revision of the ATmega168PA device.

34.3.1 Rev E

No known errata.

34.4 Errata ATmega328P

The revision letter in this section refers to the revision of the ATmega328P device.

34.4.1 Rev D

No known errata.

34.4.2 Rev C

Not sampled.

34.4.3 Rev B

- **Unstable 32 kHz Oscillator**

1. **Unstable 32 kHz Oscillator**
   The 32 kHz oscillator does not work as system clock.

   The 32 kHz oscillator used as asynchronous timer is inaccurate.

   Problem Fix/ Workaround
   None

34.4.4 Rev A

- **Unstable 32 kHz Oscillator**

1. **Unstable 32 kHz Oscillator**
   The 32 kHz oscillator does not work as system clock.

   The 32 kHz oscillator used as asynchronous timer is inaccurate.

   Problem Fix/ Workaround
   None

35. 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.

35.1 Rev. 8161D – 10/09

1. Inserted Table on page 32, Capacitance for Low-frequency Oscillator.

35.2 Rev. 8161C – 05/09

1. Updated "Features" on page 1 for ATmega48PA/88PA/168PA/328P.
2. Updated "Overview" on page 5 included the Table 2-1 on page 6.
3. Updated "AVR Memories" on page 16 included "Register Description" on page 21 and inserted Figure 7-1 on page 17.
4. Updated "Register Description" on page 44.
5. Updated "System Control and Reset" on page 46.
6. Updated "Interrupts" on page 57.
7. Updated "External Interrupts" on page 70.
8. Updated "Boot Loader Support – Read-While-Write Self-Programming, ATmega88PA, ATmega168PA and ATmega328P" on page 277.
9. Inserted "ATmega168PA DC Characteristics" on page 315.
10. Inserted "ATmega328P DC Characteristics" on page 316.
11. Inserted "ATmega168PA Typical Characteristics" on page 375.
12. Inserted "ATmega328P Typical Characteristics" on page 399.
13. Inserted Ordering Information for "ATmega168PA" on page 432.
14. Inserted Ordering Information for "ATmega328P" on page 433.
15. Inserted "Errata ATmega328P" on page 438.
16. Editing updates.

35.3 Rev. 8161B – 01/09

1. Updated "Features" on page 1 for ATmega48PA and updated the book accordingly.
2. Updated "Overview" on page 5 included the Table 2-1 on page 6.
3. Updated "AVR Memories" on page 16 included "Register Description" on page 21 and inserted Figure 7-1 on page 17.
4. Updated "Register Description" on page 44.
5. Updated "System Control and Reset" on page 46.

6. Updated "Interrupts" on page 57.
7. Updated "External Interrupts" on page 70.
8. Inserted Typical characteristics for "ATmega48PA Typical Characteristics" on page 327.
9. Updated figure names in Typical characteristics for "ATmega88PA Typical Characteristics" on page 351.
10. Inserted "ATmega48PA DC Characteristics" on page 314.
11. Updated Table 28-1 on page 317 by removing the footnote from Vcc/User calibration
12. Updated Table 28-7 on page 323 by removing Max value (2.5 LSB) from Absolute accuracy, V_{REF} = 4V, V_{CC} = 4V, ADC clock = 200 kHz.
13. Inserted Ordering Information for "ATmega48PA" on page 430.

35.4 Rev. 8161A – 11/08

1. Initial revision (Based on the ATmega48P/88P/168P/328P datasheet 8025F-AVR-08/08).
2. Changes done compared to ATmega48P/88P/168P/328P datasheet 8025F-AVR-08/08:
   - Updated "DC Characteristics" on page 313 with new typical values for I_{CC}.
   - Updated "Speed Grades" on page 316.
   - New graphics in "Typical Characteristics" on page 326.
   - New "Ordering Information" on page 430.

Table of Contents

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

XBee/XBee-PRO®DigiMesh 2.4 RF Modules

XBee-PRO® DigiMesh 2.4 RF Modules

RF Module Operation

RF Module Configuration

Appendices

RF Modules by Digi International
Firmware version:

8x6x XBee/XBee-PRO DigiMesh 2.4

Digi International Inc.
11001 Bren Road East
Minnetonka, MN 55343
877 912-3444 or 952 912-3444
http://www.digi.com

90000991_F
March 8, 2013

Technical Support:	Phone:	(866) 765-9885 toll-free U.S.A. & Canada (801) 765-9885 Worldwide 8:00 am - 5:00 pm [U.S. Mountain Time]
	Online Support:	http://www.digi.com/support/eservice/login.jsp
	Email:	rf-experts@digi.com

Contents

Contents

1. XBee® DigiMesh RF Modules

The XBee/XBee-PRO® DigiMesh 2.4 RF Modules were engineered to support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices.

The modules operate within the ISM 2.4 MHz frequency band.

Key Features

High Performance, Low Cost

XBee

- Indoor/Urban: up to 100 ft (30 m)
- Outdoor line-of-sight: up to 300 ft (100 m)
- Transmit Power Output: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300 ft (100 m)
- Outdoor line-of-sight: up to 1 mile (1500 m)
- Transmit Power Output: 100 mW (20 dBm) EIRP
- Receiver Sensitivity: -100 dBm
- RF Data Rate: 250 kbps

Advanced Networking & Security

- Retries and Acknowledgements
- Optional self-routing, self-healing mesh networking available
- DSSS (Direct Sequence Spread Spectrum)

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down current (cyclic sleep): <50 µA
- Power-down current (pin sleep): <10 µA

XBee-PRO

- TX Peak Current: 250 mA (150 mA for international variant)
- TX Peak Current (RPSMA module only): 340 mA (180 mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down current: (cyclic sleep): <50 µA
- Power-down current: (pin sleep): <10 µA

Easy-to-Use

- No configuration necessary for out-of box RF communications
- AT and API Command Modes for configuring module parameters
- Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix B for FCC Requirements. Systems that contain XBee-PRO® DigiMesh 2.4 RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee-PRO DigiMesh 2.4 RF Modules are optimized for use in **US** and **Canada**, (contact Digi for complete list of agency approvals).

Specifications

Specifications of the XBee/XBee-PRO® DigiMesh 2.4 RF Module

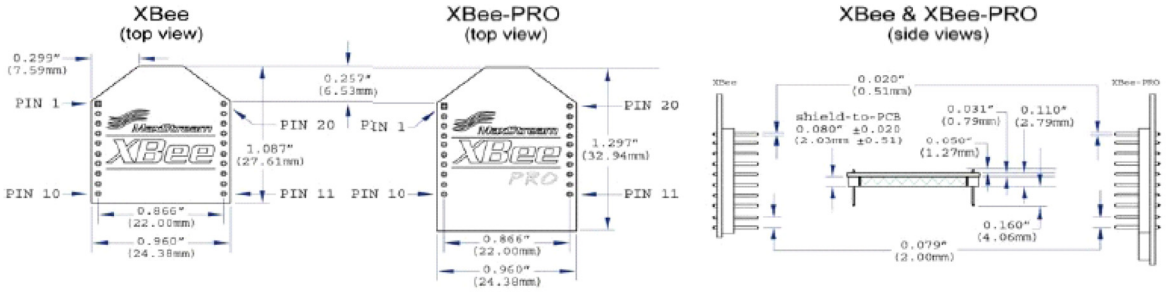
Specification	XBee	XBee-PRO®
Performance		
Indoor/Urban Range	up to 100 ft (30 m)	up to300 ft (90 m), up to 200 ft (60 m) intl. variant
Outdoor RF line-of-sight Range	up to 300 ft (90 m)	up to1 mile (1.5 km) w/2.0 dB dipole antenna up to 6 miles (10 km) w/high gain antenna
Transmit Power Output	1 mW (0 dBm)	63 mW (18 dBm)* 10 mW (10 dBm) for international variant
RF Data Rate	250 kbps	250 kbps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 - 3.4 VDC	2.8 - 3.4 VDC
Transmit Current	45 mA (@ 3.3 V)	250 mA (@ 3.3 V) (150 mA for international variant) RPSMA module only: 340 mA (@ 3.3 V) (180 mA for international variant)
Idle / Receive Current	50 mA (@ 3.3 V)	55 mA (@ 3.3 V)
Power-down Current (pin sleep)	<10 µA	<10 µA
Power-down Current (cyclic sleep)	<50 µA	<50 µA
General		
Operating Frequency Band	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438 cm x 2.761 cm)	0.960" x 1.297" (2.438 cm x 3.294 cm)
Operating Temperature	-40 to 85 C (Industrial), 0 to 95% non-condensing	-40 to 85 C (Industrial), 0 to 95% non-condensing
Antenna Options	1/4 wave wire antenna, embedded PCB antenna, RPSMA RF connector, U.FI RF connector	1/4 wave wire antenna, embedded PCB antenna, RPSMA RF connector, U.FI RF connector
Networking & Security		
Supported Network Topologies	Mesh, Point-to-point, point-to-multipoint, peer-to-peer	Mesh, Point-to-point, point-to-multipoint, peer-to-peer
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and 64-bit addresses	PAN ID, Channel and 64-bit addresses
Encryption	128 bit AES	128 bit AES
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max 10dBm transmit power output)*
RoHS	Lead-free and RoHS compliant	Lead-free and RoHS compliant
Japan	R201WW07215214	R201WW08215111" (Max. 10 dBm transmit power output)** Wire, chip, RPMSA, and U.FL versions are certified for Japan. PCB antenna version is not.
Australia	C-Tick	C-Tick

*See Appendix B for region-specific certification requirements

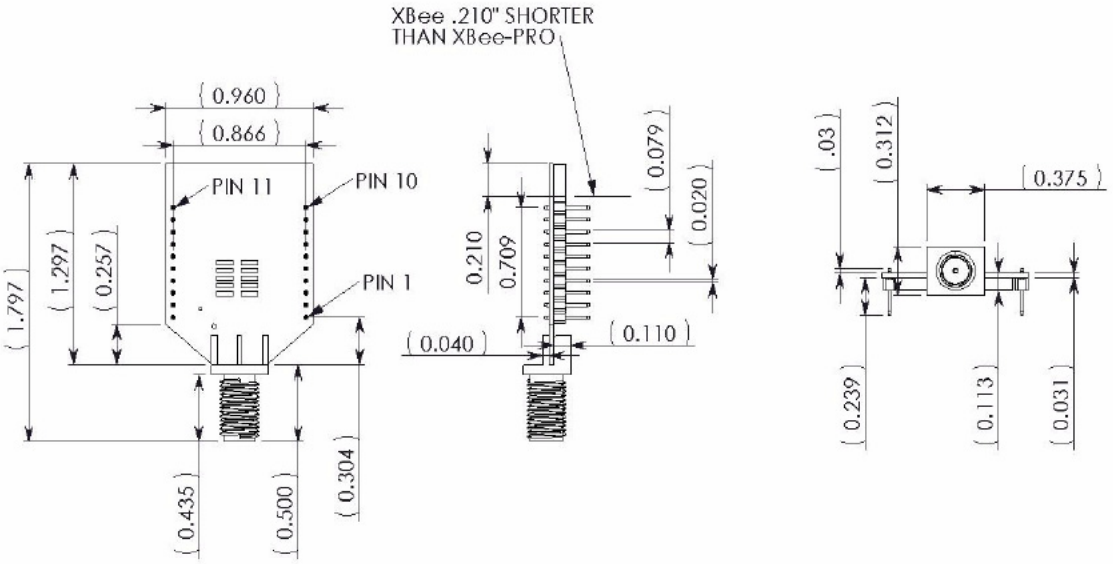
Mechanical Drawings

Mechanical drawings of the XBee/XBee-PRO®RF Modules (antenna opstions not shown)

.

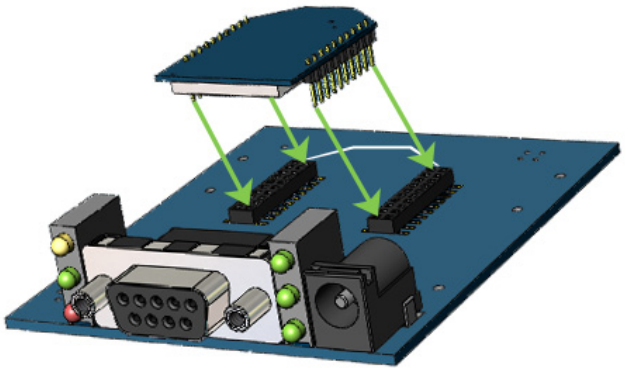


Mechanical Drawings for the RPSMA Variant



Mounting Considerations

The XBee/XBee-PRO® DigiMesh 2.4 RF Module (through-hole) was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.



The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- Through-hole single-row receptacles - Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles - Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles - Samtec P/N: SMM-110-02-SM-S

Digi also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.

Pin Signals

Pin Assignments for the XBee/XBee-PRO® DigiMesh 2.4 RF Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	Vcc	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / **CONFIG**	Input	UART Data In
4	DIO12	Either	Digital I/O 12
5	**RESET**	Input/Open drain output	Module Reset (reset pulse must be at least 100 us. This must be driven as an open drain/collector. The module will drive this line low when a reset occurs. This line should never be driven high.)
6	PWM0 / RSSI / DIO10	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	PWM Output 1 / Digital I/O 11
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ/ DIO8	Either	Pin Sleep Control Line or Digital IO 8
10	GND	-	Ground
11	AD4/DIO4	Either	Analog Input 4 or Digital I/O 4
12	**CTS** / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / **SLEEP**	Output	Module Status Indicator or Digital I/O 9
14	VREF	-	This line must be connected if analog IO sampling is desired. Must be between 2.6 V and Vcc.
15	Associate / DIO5/AD5	Either	Associated Indicator, Digital I/O 5
16	**RTS** / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Either	Analog Input 0, Digital I/O 0, or Commissioning Button

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DOUT, DIN, RTS & DTR
- Signal direction is specified with respect to the module
- Module includes a 50 kΩ pull-up resistor attached to $\overline{\text{RESET}}$
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

Design Notes

The XBee modules do not specifically require any external circuitry or specific connections for proper operation. However, there are some general design guidelines that are recommended for help in troubleshooting and building a robust design.

Power Supply Design

Poor power supply can lead to poor radio performance especially if the supply voltage is not kept within tolerance or is excessively noisy. To help reduce noise a 1.0 uF and 8.2pF capacitor are recommended to be placed as near to pin1 on the PCB as possible. If using a switching regulator for your power supply, switching frequencies above 500kHz are preferred. Power supply ripple should be limited to a maximum 100mV peak to peak.

Recommended Pin Connections

The only required pin connections are VCC, GND, DOUT and DIN. To support serial firmware updates, VCC, GND, DOUT, DIN, RTS, and DTR should be connected.

All unused pins should be left disconnected. All inputs on the radio can be pulled high with internal pull-up resistors using the PR software command. No specific treatment is needed for unused outputs.

Other pins may be connected to external circuitry for convenience of operation including the Associate LED pin (pin 15) and the commissioning button pin (pin 20). The Associate LED pin will flash differently depending on the state of the module, and a pushbutton attached to pin 20 can enable various deployment and troubleshooting functions without having to send UART commands.

If analog sampling is desired the VRef pin (pin 14) should be attached to a voltage reference.

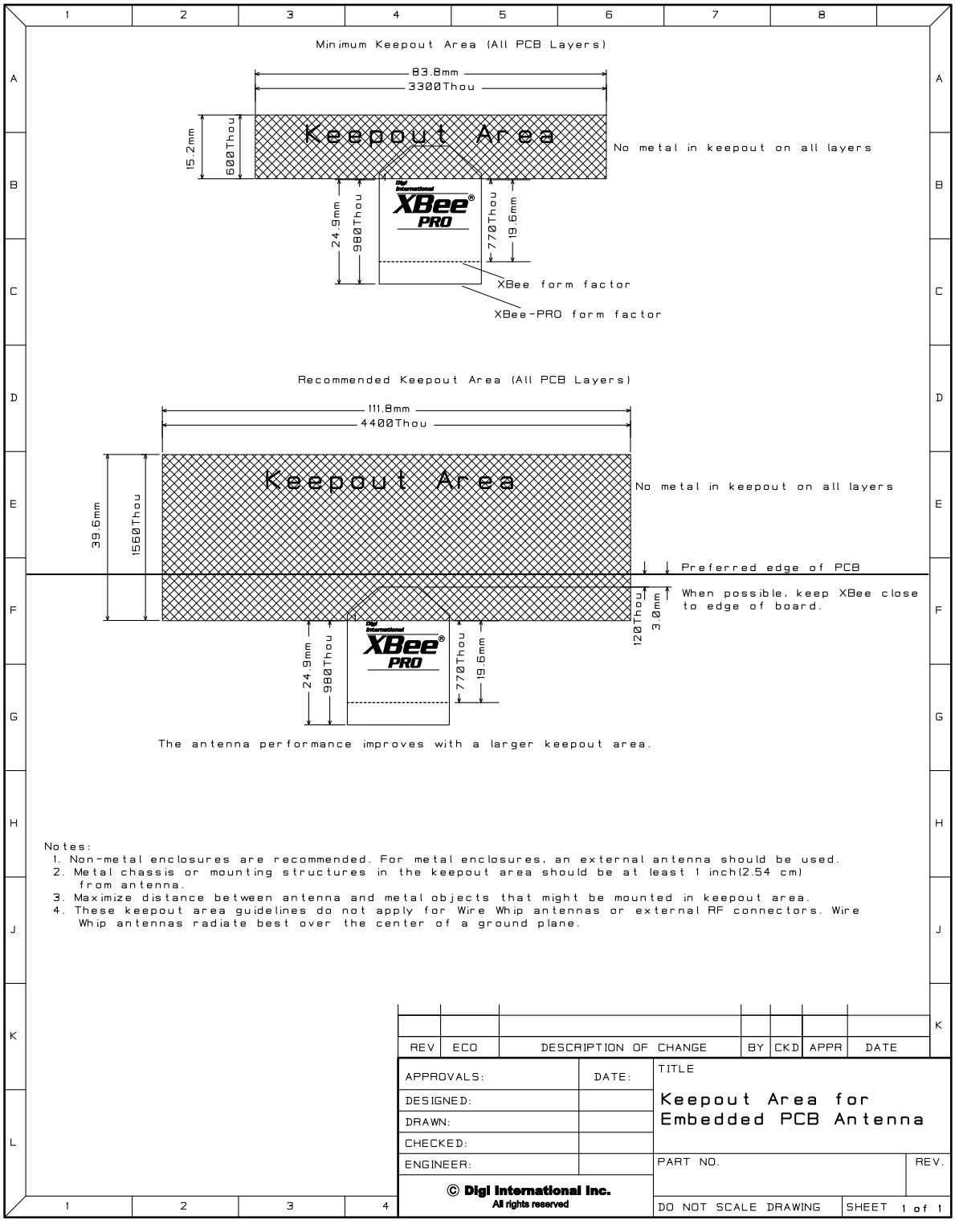
Board Layout

XBee modules are designed to be self sufficient and have minimal sensitivity to nearby processors, crystals or other PCB components. As with all PCB designs, Power and Ground traces should be thicker than signal-traces and able to comfortably support the maximum current specifications. No other special PCB design considerations are required for integrating XBee radios except in the antenna section.

Antenna Performance

Antenna location is an important consideration for optimal performance. In general, antennas radiate and receive best perpendicular to the direction they point. Thus a vertical antenna's radiation pattern is strongest across the horizon. Metal objects near the antenna may impede the radiation pattern. Metal objects between the transmitter and receiver can block the radiation path or reduce the transmission distance, so antennas should be positioned away from them when possible. Some objects that are often overlooked are metal poles, metal studs or beams in structures, concrete (it is usually reinforced with metal rods), vehicles, elevators, ventilation ducts, refrigerators, microwave ovens, batteries, and tall electrolytic capacitors. If the XBee is to be placed inside a metal enclosure, an external antenna should be used.

XBees with the Embedded PCB Antenna should not be placed inside a metal enclosure or have any ground planes or metal objects above or below the antenna. For best results, place the XBee at the edge of the host PCB on which it is mounted. Ensure that the ground, power and signal planes are vacant immediately below the antenna section. Digi recommends allowing a "keepout" area, which is shown in detail on the next page.



Electrical Characteristics

DC Characteristics of the XBee-PRO® (VCC =3.0-3.6VDC)

Symbol	Parameter	Condition	Min	Typical	Max	Units
V_{IL}	Input Low Voltage	All Digital Inputs	-	-	0.2 * VCC	V
V_{IH}	Input High Voltage	All Digital Inputs	0.8 * VCC	-	-	V
V_{OL}	Output Low Voltage	I_{OL} = 2 mA, VCC >= 3.0 V	-	-	0.18*VCC	V
V_{OH}	Output High Voltage	I_{OH} = 2 mA, VCC >= 3.0 V	0.82*VCC	-	-	V
II_{IN}	Input Leakage Current	V_{IN} = VCC or GND, all inputs, per pin	-	-	0.5	µA

ADC Characteristics (Operating)

Symbol	Parameter	Condition	Min	Typical	Max	Units
V_{REFH}	VREF - Analog-to-Digital converter reference range		2.08	-	V_{DDAD}	V
I_{REF}	VREF - Reference Supply Current	Enabled	-	200	-	µA
		Disabled or sleep mode	-	< 0.01	0.02	µA
V_{INDC}	Analog Input Voltage		V_{SSAD} - 0.3	-	V_{SSAD} + 0.3	V

1 Maximum electrical operating range, not valid conversion range

ADC Timing/Performance Characteristics[1]

Symbol	Parameter	Condition	Min	Typical	Max	Units
R_{AS}	Source Impedance at Input[2]		-	-	10	kΩ
V_{AIN}	Analog Input Voltage[3]		V_{REFL}		V_{REFH}	V
RES	Ideal Resolution (1 LSB)[4]	2.08V > V_{DDAD} > 3.6V	2.031		3.516	mV
DNL	Differential Non-linearity[5]		-	±0.5	±1.0	LSB
INL	Integral Non-linearity[6]		-	±0.5	±1.0	LSB
E_{ZS}	Zero-scale Error[7]		-	±0.4	±1.0	LSB
F_{FS}	Full-scale Error[8]		-	±0.4	±1.0	LSB
E_{IL}	Input Leakage Error[9]		-	±0.05	±5.0	LSB
E_{TU}	Total Unadjusted Error		-	±1.1	±2.5	LSB

1 All ACCURACY numbers are based on processor and system being in WAIT state (very little activity and no IO switching) and that adequate low-pass filtering is present on analog input pins (filter with 0.01 µF to 0.1 µF capacitor between analog input and V_{REFL}). Failure to observe these guidelines may result in system or microcontroller noise causing accuracy errors which will vary based on board layout and the type and magnitude of the activity. Data transmission and reception during data conversion may cause some degradation of these specifications, depending on the number and timing of packets. It is advisable to test the ADCs in your installation if best accuracy is required.

2 R_{AS} is the real portion of the impedance of the network driving the analog input pin. Values greater than this amount may not fully charge the input circuitry of the ATD resulting in accuracy error.

3 Analog input must be between V_{REFL} and V_{REFH} for valid conversion. Values greater than V_{REFH} will convert to $3FF.

4 The resolution is the ideal step size or 1LSB = (V_{REFH}–V_{REFL})/1024

5 Differential non-linearity is the difference between the current code width and the ideal code width (1LSB). The current code width is the difference in the transition voltages to and from the current code.

6 Integral non-linearity is the difference between the transition voltage to the current code and the adjusted ideal transition voltage for the current code. The adjusted ideal transition voltage is (Current Code.1/2)*(1/((V_{REFH}+E_{FS}).(V_{REFL}+E_{ZS}))).

7 Zero-scale error is the difference between the transition to the first valid code and the ideal transition to that code. The Ideal transition voltage to a given code is (Code.1/2)*(1/(V_{REFH}.V_{REFL})).

8 Full-scale error is the difference between the transition to the last valid code and the ideal transition to that code. The ideal transition voltage to a given code is (Code.1/2)*(1/(V_{REFH}.V_{REFL})).

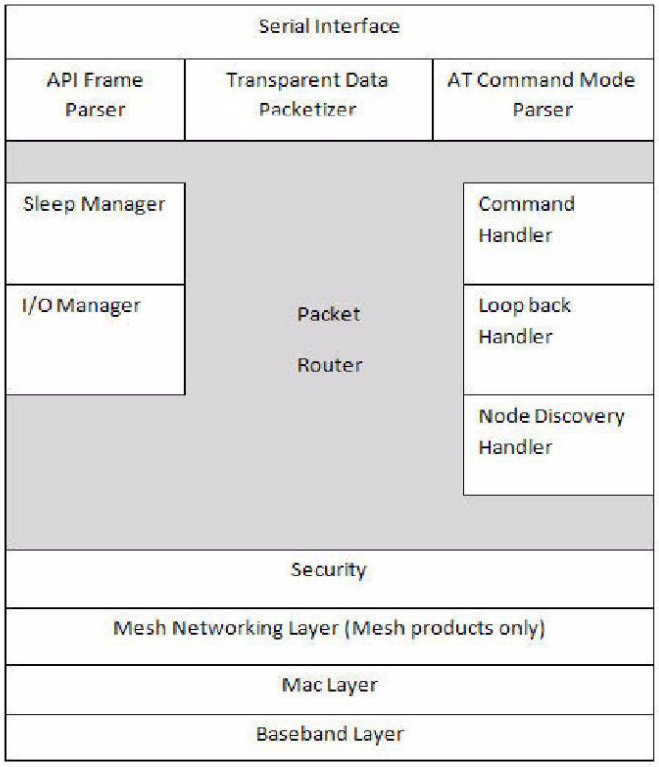
9 Input leakage error is error due to input leakage across the real portion of the impedance of the network driving the analog pin. Reducing the impedance of the network reduces this error.

10 Total unadjusted error is the difference between the transition voltage to the current code and the ideal straight-line transfer function. This measure of error includes inherent quantization error (1/2LSB) and circuit error (differential, integral, zero-scale, and full-scale) error. The specified value of E_{TU} assumes zero E_{IL} (no leakage or zero real source impedance).

2. RF Module Operation

Overview

The XBee module provides a serial interface to an RF link. The XBee module can convert serial data to RF data that can be sent to any device in an RF network. In addition to RF data communication devices, the XBee module provides a software interface for interacting with a variety of peripheral functions, including I/O sampling, commissioning and management devices. The following diagram illustrates the functionality of the XBee module.



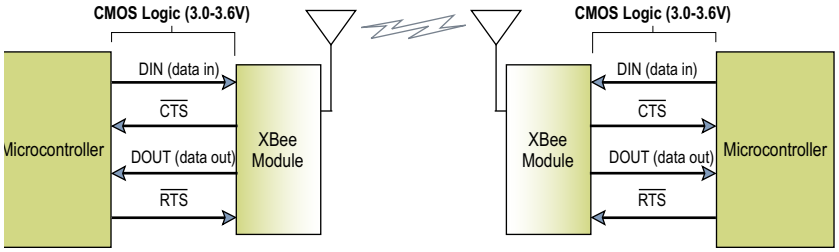
Serial Communications

The XBee-PRO® RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Digi proprietary RS-232 or USB interface board).

UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

System Data Flow Diagram in a UART-interfaced environment
(Low-asserted signals distinguished with horizontal line over signal name.)

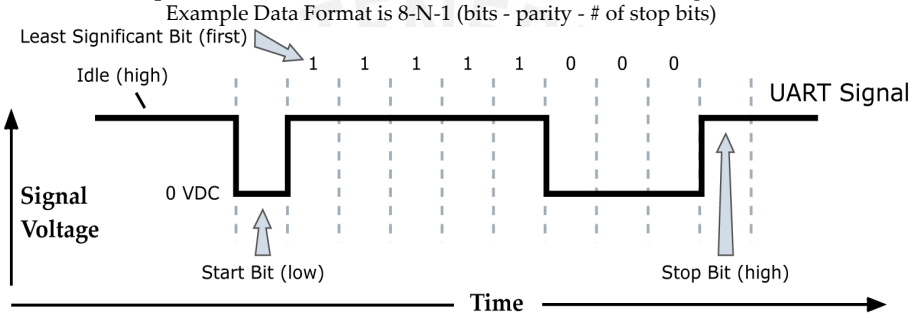


Serial Data

Data enters the module UART through the DIN (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

UART data packet 0x1F (decimal number "31") as transmitted through the RF module
Example Data Format is 8-N-1 (bits - parity - # of stop bits)

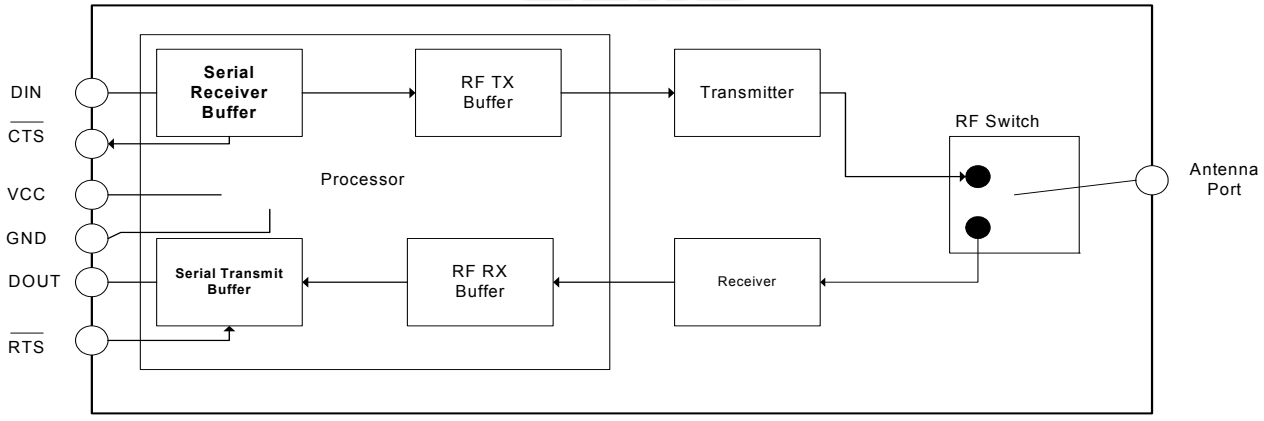


The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

Serial Buffers

The XBee-PRO modules maintain buffers to collect received serial and RF data, which is illustrated in the figure below. The serial receive buffer collects incoming serial characters and holds them until they can be processed. The serial transmit buffer collects data that is received via the RF link that will be transmitted out the UART.

Internal Data Flow Diagram



Serial Receive Buffer

When serial data enters the RF module through the DIN Pin (pin 3), the data is stored in the serial receive buffer until it can be processed. Under certain conditions, the module may not be able to process data in the serial receive buffer immediately. If large amounts of serial data are sent to the module, CTS flow control may be required to avoid overflowing the serial receive buffer.

Cases in which the serial receive buffer may become full and possibly overflow:

- If the module is receiving a continuous stream of RF data, the data in the serial receive buffer will not be transmitted until the module is no longer receiving RF data.
- For mesh networking firmware, if the module is transmitting an RF data packet, the module may need to discover the destination address or establish a route to the destination. After transmitting the data, the module may need to retransmit the data if an acknowledgment is not received, or if the transmission is a broadcast. These issues could delay the processing of data in the serial receive buffer.

Serial Transmit Buffer

When RF data is received, the data is moved into the serial transmit buffer and is sent out the serial port. If the serial transmit buffer becomes full enough such that all data in a received RF packet won't fit in the serial transmit buffer, the entire RF data packet is dropped.

Cases in which the serial transmit buffer may become full resulting in dropped RF packets

- If the RF data rate is set higher than the interface data rate of the module, the module could receive data faster than it can send the data to the host. Even occasional transmissions from a large number of modules can quickly add up and overflow the transmit buffer.
- If the host does not allow the module to transmit data out from the serial transmit buffer because of being held off by hardware flow control.

Serial Flow Control

The $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ module pins can be used to provide $\overline{\text{RTS}}$ and/or $\overline{\text{CTS}}$ flow control. $\overline{\text{CTS}}$ flow control provides an indication to the host to stop sending serial data to the module. $\overline{\text{RTS}}$ flow control allows the host to signal the module to not send data in the serial transmit buffer out the UART. $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ flow control are enabled using the D6 and D7 commands.

$\overline{\text{CTS}}$ Flow Control

If $\overline{\text{CTS}}$ flow control is enabled (D7 command), when the serial receive buffer is filled with FT bytes, the module de-asserts $\overline{\text{CTS}}$ (sets it high) to signal to the host device to stop sending serial data. $\overline{\text{CTS}}$ is re-asserted when less than FT - 16 bytes are in the UART receive buffer. (See command description for the FT command.)

$\overline{\text{RTS}}$ Flow Control

If $\overline{\text{RTS}}$ flow control is enabled (D6 command), data in the serial transmit buffer will not be sent out the DOUT pin as long as $\overline{\text{RTS}}$ is de-asserted (set high). The host device should not de-assert $\overline{\text{RTS}}$ for long periods of time to avoid filling the serial transmit buffer. If an RF data packet is received, and the serial transmit buffer does not have enough space for all of the data bytes, the entire RF data packet will be discarded.

Serial Interface Protocols

The XBee modules support both transparent and API (Application Programming Interface) serial interfaces.

Transparent Operation

When operating in transparent mode, the modules act as a serial line replacement. All UART data received through the DIN pin is queued up for RF transmission. When RF data is received, the data is sent out through the DOUT pin. The module configuration parameters are configured using the AT command mode interface.

Data is buffered in the serial receive buffer until one of the following causes the data to be packetized and transmitted:

- No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.

- The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the serial receive buffer before the sequence is transmitted.
- The maximum number of characters that will fit in an RF packet is received

API Operation

API operation is an alternative to transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and leaving the module's UART is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DIN pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DOUT pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, sync status, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

- Transmitting data to multiple destinations without entering Command Mode
- Receive success/failure status of each transmitted RF packet
- Identify the source address of each received packet.

The following table compares the advantages of transparent and API modes of operation:

Transparent Operation Features	
Simple Interface	All received serial data is transmitted unless the module is in command mode.
Easy to support	It is easier for an application to support transparent operation and command mode
API Operation Features	
Easy to manage data transmissions to multiple destinations	Transmitting RF data to multiple remotes only requires changing the address in the API frame. This process is much faster than in transparent operation where the application must enter AT command mode, change the address, exit command mode, and then transmit data. Each API transmission can return a transmit status frame indicating the success or reason for failure.
Received data frames indicate the sender's address	All received RF data API frames indicate the source address.
Advanced addressing support	API transmit and receive frames can expose addressing fields including source and destination endpoints, cluster ID and profile ID.
Advanced networking diagnostics	API frames can provide indication of IO samples from remote devices, and node identification messages.
Remote Configuration	Set / read configuration commands can be sent to remote devices to configure them as needed using the API.

As a general rule of thumb, API firmware is recommended when a device:

- sends RF data to multiple destinations
- sends remote configuration commands to manage devices in the network
- receives IO samples from remote devices
- receives RF data packets from multiple devices, and the application needs to know which device sent which packet.

If the above conditions do not apply (i.e. a sensor node, router, or a simple application), then AT firmware might be suitable. It is acceptable to use a mixture of devices running API and AT firmware in a network.

> To implement API operations, refer to the API Operation chapter (Chapter 7).

Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. During Idle Mode, the RF module is checking for valid RF data. The module shifts into the other modes of operation under the following conditions:

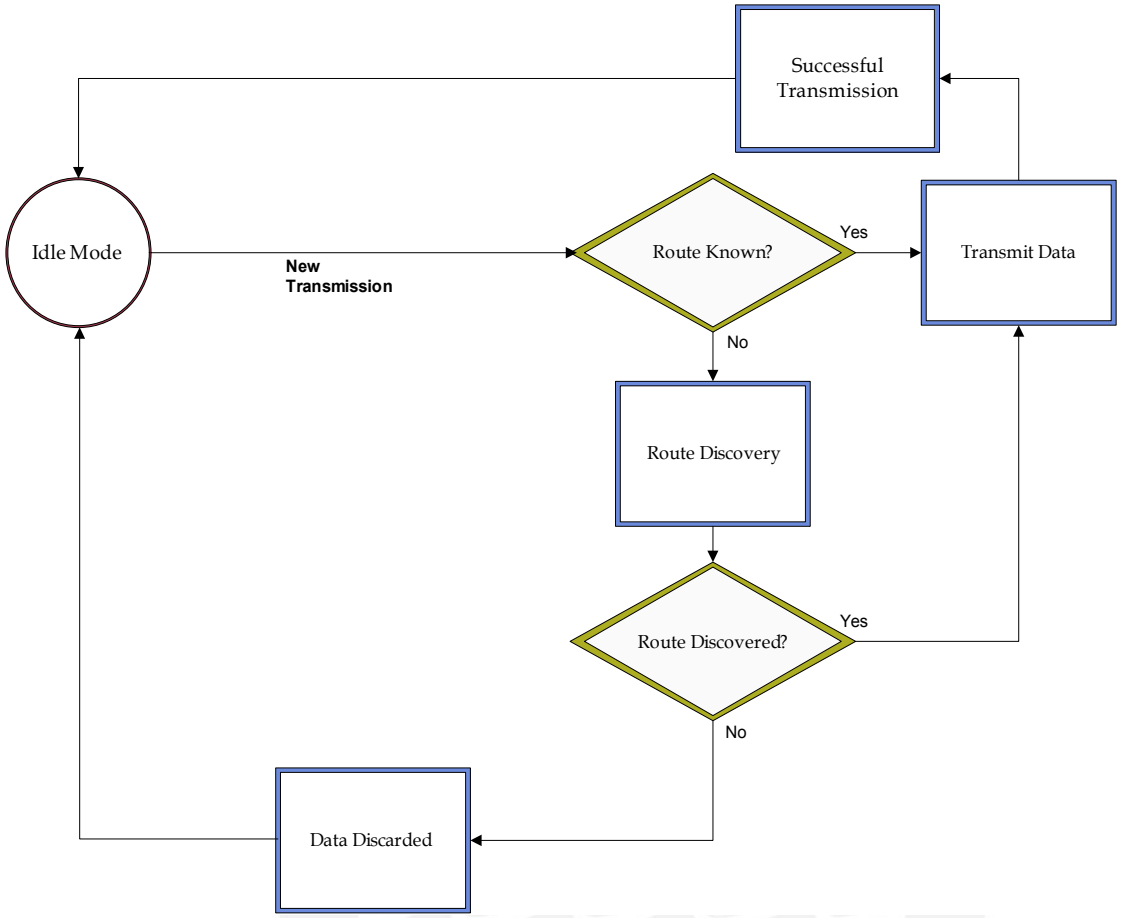
- Transmit Mode (Serial data in the serial receive buffer is ready to be packetized)
- Receive Mode (Valid RF data is received through the antenna)
- Command Mode (Command Mode Sequence is issued)
- Sleep Mode (A device is configured for sleep)

Transmit Mode

When serial data is received and is ready for packetization, the RF module will exit Idle Mode and attempt to transmit the data. The destination address determines which node(s) will receive the data.

For mesh firmware, if a route is not known, route discovery will take place for the purpose of establishing a route to the destination node. If a module with a matching network address is not discovered, the packet is discarded. The data will be transmitted once a route is established. Route discovery will be attempted only once per packet.

Figure 4-01. Transmit Mode Sequence



When data is transmitted from one node to another, a network-level acknowledgement is transmitted back across the established route to the source node. This acknowledgement packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgement is not received, the source node will re-transmit the data. See Data Transmission and Routing in chapter 4 for more information.

Receive Mode

If a valid RF packet is received, the data is transferred to the serial transmit buffer.

Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands. Refer to the API Mode section for an alternate means of configuring modules.

AT Command Mode

To Enter AT Command Mode:
Send the 3-character command sequence "+++" and observe guard times before and after the command characters. [Refer to the "Default AT Command Mode Sequence" below.]

Default AT Command Mode Sequence (for transition to Command Mode):

• No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

Once the AT command mode sequence has been issued, the module sends an "OK\r" out the DOUT pin. The "OK\r" characters can be delayed if the module has not finished transmitting received serial data.

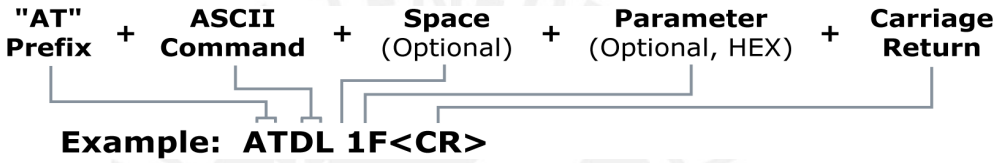
When command mode has been entered, the command mode timer is started (CT command), and the module is able to receive AT commands on the DIN pin.

All of the parameter values in the sequence can be modified to reflect user preferences.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. When using XCTU, ensure that the 'Baud' setting on the "PC Settings" tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (9600 bps).

To Send AT Commands:
Send AT commands and parameters using the syntax shown below.

Syntax for sending AT Commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

Command Response. When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an "OK" message. If execution of a command results in an error, the module returns an "ERROR" message.

Applying Command Changes

Any changes made to the configuration command registers through AT commands will not take effect until the changes are applied. For example, sending the BD command to change the baud rate will not change the actual baud rate until changes are applied. Changes can be applied in one of the following ways:

- The AC (Apply Changes) command is issued.
- AT command mode is exited.

To Exit AT Command Mode:

1. Send the ATCN (Exit Command Mode) command (followed by a carriage return).

   [OR]

2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

For an example of programming the RF module using AT Commands and descriptions of each configurable parameter, refer to the Command Reference Tables chapter.

Sleep Mode

Sleep modes allow the RF module to enter states of low power consumption when not in use. The XBee RF Modules support both pin sleep (sleep mode entered on pin transition) and cyclic sleep

(Module sleeps for a fixed time). The XBee DigiMesh modules support a network synchronized sleep to conserve power. XBee sleep modes are discussed in detail in Chapter 5.

7. Advanced Application Features

Remote Configuration Commands

A module in API mode has provisions to send configuration commands to remote devices using the Remote Command Request API frame (See API Operations chapter.) This API frame can be used to send commands to a remote module to read or set command parameters.

Sending a Remote Command

To send a remote command, the Remote Command Request frame should be populated with the 64-bit address of the remote device, the correct command options value, and the command and parameter data (optional). If a command response is desired, the Frame ID should be set to a non-zero value. Only unicasts of remote commands are supported. Remote commands cannot be broadcast.

Applying Changes on Remote Devices

When remote commands are used to change command parameter settings on a remote device, parameter changes do not take effect until the changes are applied. For example, changing the BD parameter will not change the actual serial interface rate on the remote until the changes are applied. Changes can be applied using remote commands in one of three ways:

- Set the apply changes option bit in the API frame
- Issue an AC command to the remote device
- Issue a WR + FR command to the remote device to save changes and reset the device.

Remote Command Responses

If the remote device receives a remote command request transmission, and the API frame ID is non-zero, the remote will send a remote command response transmission back to the device that sent the remote command. When a remote command response transmission is received, a device sends a remote command response API frame out its UART. The remote command response indicates the status of the command (success, or reason for failure), and in the case of a command query, it will include the register value. The device that sends a remote command will not receive a remote command response frame if:

- The destination device could not be reached
- The frame ID in the remote command request is set to 0.

Network Commissioning and Diagnostics

Network commissioning is the process whereby devices in a network are discovered and configured for operation. The XBee modules include several features to support device discovery and configuration. In addition to configuring devices, a strategy must be developed to place devices to ensure reliable routes.

To accommodate these requirements, the XBee modules include various features to aid in device placement, configuration, and network diagnostics.

Device Configuration

XBee modules can be configured locally through serial commands (AT or API), or remotely through remote API commands. API devices can send configuration commands to set or read the configuration settings of any device in the network.

Building Aggregate Routes

In many applications it is necessary for many or all of the nodes in the network to transmit data to a central aggregator node. In a new DigiMesh network the overhead of these nodes discovering routes to the aggregator node can be extensive and taxing on the network. To eliminate this overhead the AG command can be used to automatically build routes to an aggregate node in a DigiMesh network.

To send a unicast, modules configured for transparent mode (AP=0) must set their DH/DL registers to the MAC address of the node to which they need to transmit to. In networks of transparent mode modules which transmit to an aggregator node it is necessary to set every module's DH/DL registers to the MAC address of the aggregator node. This can be a tedious process.

Upon deploying a DigiMesh network the AG command can be issued on the desired aggregator node to cause all nodes in the network to build routes to the aggregator node. The command can optionally be used to automatically update the DH/DL registers to match the MAC address of the aggregator node. The AG command requires a 64-bit parameter. The parameter indicates the current value of the DH/DL registers on a module which should be replaced by the 64-bit address of the node sending the AG broadcast. If it is not desirable to update the DH/DL of the module receiving the AG broadcast then the invalid address of 0xFFFE can be used. API enabled modules will output an Aggregator Update API frame if they update their DH/DL address (see the API section of this manual for a description of the frame). All modules which receive an AG broadcast will update their routing table information to build a route to the sending module, regardless of whether or not their DH/DL address is updated. This routing information will be used for future transmissions of DigiMesh unicasts.

Example 1: To update the DH/DL registers of all modules in the network to be equal to the MAC address of an aggregator node with a MAC address of 0x0013a2004052c507 after network deployment the following technique could be employed:

- Deploy all modules in the network with the default DH/DL of 0xFFFF.
- Issue an ATAGFFFF command on the aggregator node.

Following the preceding sequence would result in all of the nodes in the network which received the AG broadcast to have a DH of 0x0013a200 and a DL of 0x4052c507. These nodes would have automatically built a route to the aggregator.

Example 2: To cause all nodes in the network to build routes to an aggregator node with a MAC address of 0x0013a2004052c507 without affecting the DH/DL of any nodes in the network the ATAGFFFE command should be issued on the aggregator node. This will cause an AG broadcast to be sent to all nodes in the network. All of the nodes will update their internal routing table information to contain a route to the aggregator node. None of the nodes will update their DH/DL registers (because none of the registers are set to an address of 0xFFFE).

Node Replacement

The AG command can also be used to update the routing table and DH/DL registers in the network after a module is replaced. The DH/DL registers of nodes in the network can also be updated. To update only the routing table information without affecting the DH/DL registers then the process of Example 2 above can be used. To update the DH/DL registers of the network then the method of Example 3 below can be used.

Example 3: The module with serial number 0x0013a2004052c507 was being used as a network aggregator. It was replaced with a module with serial number 0x0013a200f5e4d3b2. The AG0013a2004052c507 command should be issued on the new module. This will cause all modules which had a DH/DL register setting of 0x0013a2004052c507 to update their DH/DL register setting to the MAC address of the sending module (0x0013a200f5e4d3b2).

Device Placement

For a network installation to be successful, the installer must be able to determine where to place individual XBee devices to establish reliable links throughout the network.

Link Testing

A good way to measure the performance of a network is to send unicast data through the network from one device to another to determine the success rate of many transmissions. To simplify link testing, the modules support a loopback cluster ID (0x12) on the data endpoint (0xE8). Any data sent to this cluster ID on the data endpoint will be transmitted back to the sender.

The configuration steps to send data to the loopback cluster ID depend on the AP setting:

AT Configuration (AP=0)

To send data to the loopback cluster ID on the data endpoint of a remote device, set the CI command value to 0x12. The SE and DE commands should be set to 0xE8 (default value). The DH and DL commands should be set to the 64-bit address of the remote. After exiting command mode, any received serial characters will be transmitted to the remote device, and returned to the sender.

API Configuration (AP=1 or AP=2)

Send an Explicit Addressing Command API frame (0x11) using 0x12 as the cluster ID, 0xC105 as the profile ID and 0xE8 as the source and destination endpoint. Data packets received by the remote will be echoed back to the sender.

RSSI Indicators

It is possible to measure the received signal strength on a device using the DB command. DB returns the RSSI value (measured in –dBm) of the last received packet. The dB value only indicates the received signal strength of the last hop. If a transmission spans multiple hops, the dB value provides no indication of the overall transmission path, or the quality of the worst link – it only indicates the quality of the last link and should be used sparingly.

The DB value can be determined in hardware using the RSSI/PWM module pin (pin 6). If the RSSI PWM functionality is enabled (P0 command), when the module receives data, the RSSI PWM is set to a value based on the RSSI of the received packet. (Again, this value only indicates the quality of the last hop.) This pin could potentially be connected to an LED to indicate if the link is stable or not.

Network Discovery

The network discovery command can be used to discover all Digi modules that have joined a network. Issuing the ND command sends a broadcast network discovery command throughout the network. All devices that receive the command will send a response that includes the device's addressing information, node identifier string (see NI command), and other relevant information. This command is useful for generating a list of all module addresses in a network.

When a device receives the network discovery command, it waits a random time before sending its own response. The maximum time delay is set on the ND sender with the NT command. The ND originator includes its NT setting in the transmission to provide a delay window for all devices in the network. Large networks may need to increase NT to improve network discovery reliability. The default NT value is 0x82 (13 seconds).

Neighbor Polling

The neighbor poll command can be used to discover the modules which are immediate neighbors (within RF range) of a particular node. This command is useful in determining network topology and determining possible routes. The command is issued using the FN command. The FN command can be initiated locally on a node using AT command mode or by using a local AT command request frame. The command can also be initiated remotely by sending the target node an FN command using a remote AT command request API frame.

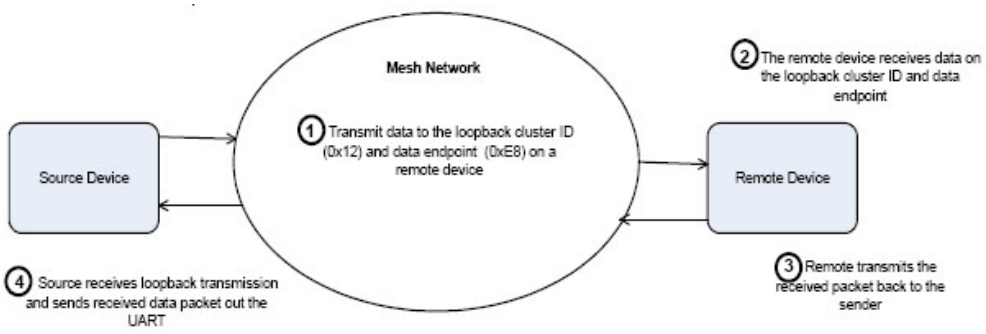
A node which executes an FN command will send a broadcast to all of its immediate neighbors. All radios which receive this broadcast will send an RF packet to the node that initiated the FN command. In the case where the command is initiated remotely this means that the responses are sent directly to the node which sent the FN command to the target node. The response packet is output on the initiating radio in the same format as a network discovery frame.

Link Reliability

For a mesh network installation to be successful, the installer must be able to determine where to place individual XBee devices to establish reliable links throughout the mesh network.

Network Link Testing

A good way to measure the performance of a mesh network is to send unicast data through the network from one device to another to determine the success rate of many transmissions. To simplify link testing, the modules support a loopback cluster ID (0x12) on the data endpoint (0xE8). Any data sent to this cluster ID on the data endpoint will be transmitted back to the sender. This is shown in the figure below:



Demonstration of how the loopback cluster ID and data endpoint can be used to measure the link quality in a mesh network

The configuration steps to send data to the loopback cluster ID depend on the AP setting:

AT Configuration (AP=0)

To send data to the loopback cluster ID on the data endpoint of a remote device, set the CI command value to 0x12. The SE and DE commands should be set to 0xE8 (default value). The DH and DL commands should be set to the address of the remote. After exiting command mode, any received serial characters will be transmitted to the remote device, and returned to the sender.

API Configuration (AP=1 or AP=2)

Send an Explicit TX Request API frame (0x11) using 0x12 as the cluster ID and 0xE8 as the source and destination endpoint. Data packets received by the remote will be echoed back to the sender.

Link Testing Between Adjacent Devices

It is often advantageous to test the quality of a link between two adjacent nodes in a network. The Test Link Request Cluster ID can be used to send a number of test packets between any two nodes in a network.

A link test can be initiated using an Explicit TX Request frame. The command frame should be addressed to the Test Link Request Cluster ID (0x0014) on destination endpoint 0xE6 on the radio which should execute the test link. The Explicit TX Request frame should contain a 12 byte payload with the following format:

Number of Bytes	Field Name	Description
8	Destination address	The address with which the radio should test its link
2	Payload size	The size of the test packet. The maximum payload size the radio can support can be queried with the NP command.
2	Iterations	The number of packets which should be sent. This should be a number between 1 and 4000.

After completing the transmissions of the test link packets the executing radio will send the following data packet to the requesting radio's Test Link Result Cluster (0x0094) on endpoint (0xE6). If the requesting radio is configured to operate in API mode then the following information will be output as an Explicit RX Indicator API Frame:

Number of Bytes	Field Name	Description
8	Destination address	The address with which the radio tested its link
2	Payload size	The size of the test packet that was sent to test the link.
2	Iterations	The number of packets which were sent.
2	Success	The number of packets successfully acknowledged.
2	Retries	The total number of MAC retries used to transfer all the packets.
1	Result	0x00 - command was successful 0x03 - invalid parameter used
1	RR	The maximum number of MAC retries allowed.
1	maxRSSI	The strongest RSSI reading observed during the test.
1	minRSSI	The weakest RSSI reading observed during the test.
1	avgRSSI	The average RSSI reading observed during the test.

Example:

Suppose that the link between radio A (SH/SL = 0x0013a20040521234) and radio B (SH/SL=0x0013a2004052abcd) is to be tested by transmitting 1000 40 byte packets. The following API packet should be sent to the serial interface of the radio on which the results should be output, radio C. Note that radio C can be the same radio as radio A or B (whitespace used to delineate fields, bold text is the payload portion of the packet):

7E 0020 11 01 0013A20040521234 FFFE E6 E6 0014 C105 00 00 **0013A2004052ABCD 0028 03E8** EB

And the following is a possible packet that could be returned:

7E 0027 91 0013A20040521234 FFFE E6 E6 0094 C105 00 **0013A2004052ABCD 0028 03E8 03E7 0064 00 0A 50 53 52** 9F

(999 out of 1000 packets successful, 100 retries used, RR=10, maxRSSI=-80dBm, minRSSI=-83dBm, avgRSSI=-82dBm)

If the result field is not equal to zero then an error has occurred. The other fields in the packet should be ignored. If the Success field is equal to zero then the RSSI fields should be ignored.

Trace Routing

In many applications it is useful to determine the route which a DigiMesh unicast takes to its destination. This information is especially useful when setting up a network or diagnosing problems within a network. The Trace Route API option of Tx Request Packets (see the API section of this manual for a description of the API frames) causes routing information packets to be transmitted to the originator of a DigiMesh unicast by the intermediate nodes.

When a unicast is sent with the Trace Route API option enabled, the unicast is sent to its destination radios which forward the unicast to its eventual destination will transmit a Route Information (RI) packet back along the route to the unicast originator. A full description of Route Information API packets can be found in the API section of this manual. In general they contain addressing information for the unicast and the intermediate hop for which the trace route packet was generated, RSSI information, and other link quality information.

Example:

Suppose that a data packet with trace route enabled was successfully unicast from radio A to radio E, through radios B, C, and D. The following sequence would occur:

- After the successful MAC transmission of the data packet from A to B, A would output a RI Packet indicating that the transmission of the data packet from A to E was successfully forwarded one hop from A to B.
- After the successful MAC transmission of the data packet from B to C, B would transmit a RI Packet to A. A would output this RI packet out its serial interface upon reception.
- After the successful MAC transmission of the data packet from C to D, C would transmit a RI Packet to A (through B). A would output this RI packet out its serial interface upon reception.
- After the successful MAC transmission of the data packet from D to E, D would transmit a RI Packet to A (through C and B). A would output this RI packet out its serial interface upon reception.

It is important to note that Route Information packets are not guaranteed to arrive in the same order as the unicast packet took. It is also possible for the transmission of Route Information packets on a weak route to fail before arriving at the unicast originator. Because of the large number of Route Information packets which can be generated by a unicast with Trace Route enabled it is suggested that the Trace Route option only be used for occasional diagnostic purposes and not for normal operations.

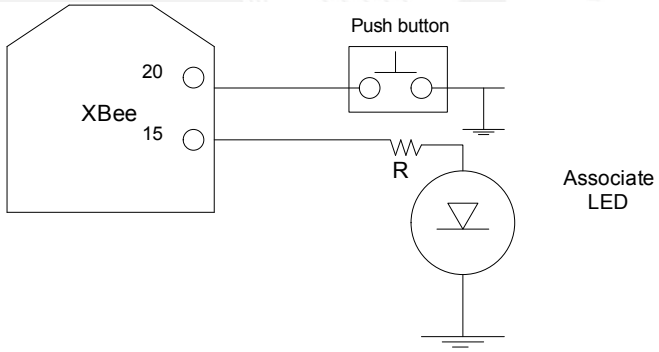
NACK Messages

The NACK API option of Tx Request Packets (see the API section of this manual for a description of the API frames) provides the option to have a Route Information packet generated and sent to the originator of a unicast when a MAC acknowledgment failure occurs on one of the hops to the destination. This information is useful because it allows marginal links to be identified and repaired.

Commissioning Pushbutton and Associate LED

The XBee modules support a set of commissioning and LED behaviors to aid in device deployment and commissioning. These include the commissioning push button definitions and associate LED behaviors. These features can be supported in hardware as shown below.

Commissioning Pushbutton and Associate LED Functionalities



A pushbutton and an LED can be connected to module pins 20 and 15 respectively to support the commissioning pushbutton and associated LED functionalities.

Commissioning Pushbutton

The commissioning pushbutton definitions provide a variety of simple functions to aid in deploying devices in a network. The commissioning button functionality on pin 20 is enabled by setting the D0 command to 1 (enabled by default).

Button Presses	Sleep Configuration and Sync Status	Action
1	Not configured for sleep	Immediately sends a Node Identification broadcast transmission. All devices that receive this transmission will blink their Associate LED rapidly for 1 second. All API devices that receive this transmission will send a Node Identification frame out their UART (API ID 0x95)
1	Configured for asynchronous sleep	Wakes the module for 30 seconds. Immediately sends a Node Identification broadcast transmission. All devices that receive this transmission will blink their Associate LED rapidly for 1 second. All API devices that receive this transmission will send a Node Identification frame out their UART (API ID 0x95).
1	Configured for synchronous sleep	Wakes the module for 30 seconds (or until the entire module goes to sleep). Queues a Node Identification broadcast transmission to be sent at the beginning of the next network wake cycle. All devices that receive this transmission will blink their Associate LEDs rapidly for 1 second. All API devices that receive this transmission will send a Node Identification frame out their UART (API ID 0x95).
2	Not configured for synchronous sleep	No effect.
2	Configured for synchronous sleep	Causes a node which is configured with sleeping router nomination enabled (see description of the ATSO – sleep options command in the XBee module's Product Manual) to immediately nominate itself as the network sleep coordinator.
4	Any	Issues an ATRE to restore module parameters to default values.

Button presses may be simulated in software using the ATCB command. ATCB should be issued with a parameter set to the number of button presses to execute. (i.e. sending ATCB1 will execute the action(s) associated with a single button press.)

The node identification frame is similar to the node discovery response frame – it contains the device's address, node identifier string (NI command), and other relevant data. All API devices that receive the node identification frame send it out their UART as an API Node Identification Indicator frame (0x95).

Having the commissioning button enabled during sleep will increase the sleeping current draw (especially in SM1 mode). It is necessary to hold down the commissioning button for up to 2 seconds to wake the module from sleep.

Associate LED

The Associate pin (pin 15) can provide indication of the device's sleep status and diagnostic information. To take advantage of these indications, an LED can be connected to the Associate pin as shown in the figure above. The Associate LED functionality is enabled by setting the D5 command to 1 (enabled by default). If enabled, the Associate pin is configured as an output and will behave as described in the following sections.

The Associate pin indicates the synchronization status of a sleep compatible node. On a non-sleep compatible node the pin functions as a power indicator. The following table describes this functionality.

The LT command can be used to override the blink rate of the Associate pin. When set to 0, the device uses the default blink time (500ms for sleep coordinator, 250ms otherwise).

Sleep mode	LED Status	Meaning
0	On, blinking	The device is powered and operating properly.
1, 4, 5	Off	The device is in a low power mode.
1, 4, 5	On, blinking	The device is powered, awake and is operating properly.
7	On, solid	The network is asleep or the device has not synchronized with the network or has lost synchronization with the network.
7, 8	On, slow blinking (500 ms blink time)	The device is acting as the network sleep coordinator and is operating properly.
7, 8	On, fast blinking (250 ms blink time)	The device is properly synchronized with the network.
8	Off	The device is in a low power mode.
8	On, solid	The deviced has not synchronized or has lost synchronization with the network.

Diagnostics Support

The Associate pin works with the commissioning pushbutton to provide additional diagnostics behaviors to aid in deploying and testing a network. If the commissioning push button is pressed once the device transmits a broadcast node identification packet at the beginning of the next wake cycle if sleep compatible, or immediately if not sleep compatible. If the Associate LED functionality is enabled (D5 command), a device that receive this transmission will blink its Associate pin rapidly for 1 second.

I/O Line Monitoring

I/O Samples

The XBee modules support both analog input and digital IO line modes on several configurable pins.

Queried Sampling

Parameters for the pin configuration commands typically include the following:

Pin Command Parameter	Description
0	Unmonitored digital input
1	Reserved for pin-specific alternate functionalities.
2	Analog input (A/D pins) or PWM output (PWM pins)
3	Digital input, monitored.
4	Digital output, default low.
5	Digital output, default high.
6-9	Alternate functionalities, where applicable.

Setting the configuration command that corresponds to a particular pin will configure the pin:

Module Pin Names	Module Pin Number	Configuration Command
CD / DIO12	4	P2
PWM0 / RSSI / DIO10	6	P0
PWM1 / DIO11	7	P1

Module Pin Names	Module Pin Number	Configuration Command
DTR / SLEEP_RQ / DIO8	9	D8
AD4 / DIO4	11	D4
CTS / DIO7	12	D7
ON_SLEEP / DIO9	13	D9
ASSOC / AD5 / DIO5	15	D5
RTS / DIO6	16	D6
AD3 / DIO3	17	D3
AD2 / DIO2	18	D2
AD1 / DIO1	19	D1
AD0 / DIO0 / CommissioningButton	20	D0

See the command table for more information. Pullup resistors for each digital input can be enabled using the PR command.

1	Sample Sets	Number of sample sets in the packet. (Always set to 1.)
2	Digital Channel Mask	Indicates which digital IO lines have sampling enabled. Each bit corresponds to one digital IO line on the module. • bit 0 = AD0/DIO0 • bit 1 = AD1/DIO1 • bit 2 = AD2/DIO2 • bit 3 = AD3/DIO3 • bit 4 = DIO4 • bit 5 = ASSOC/DIO5 • bit 6 = RTS/DIO6 • bit 7 = CTS/GPIO7 • bit 8 = DTR / SLEEP_RQ / DIO8 • bit 9 = ON_SLEEP / DIO9 • bit 10 = RSSI/DIO10 • bit 11 = PWM/DIO11 • bit 12 = CD/DIO12 For example, a digital channel mask of 0x002F means DIO0,1,2,3, and 5 are enabled as digital IO.
1	Analog Channel Mask	Indicates which lines have analog inputs enabled for sampling. Each bit in the analog channel mask corresponds to one analog input channel. • bit 0 = AD0/DIO0 • bit 1 = AD1/DIO1 • bit 2 = AD2/DIO2 • bit 3 = AD3/DIO3 • bit 4 = AD4/DIO4 • bit 5 = ASSOC/AD5/DIO5
Variable	Sampled Data Set	If any digital IO lines are enabled, the first two bytes of the data set indicate the state of all enabled digital IO. Only digital channels that are enabled in the Digital Channel Mask bytes have any meaning in the sample set. If no digital IO are enabled on the device, these 2 bytes will be omitted. Following the digital IO data (if any), each enabled analog channel will return 2 bytes. The data starts with AIN0 and continues sequentially for each enabled analog input channel up to AIN5.

If the IS command is issued from AT command mode then a carriage return delimited list will be returned containing the above-listed fields. If the command is issued via an API frame then the module will return an AT command response API frame with the IO data included in the command data portion of the packet.

Example	Sample AT Response
0x01\r	[1 sample set]
0x0C0C\r	[Digital Inputs: DIO 2, 3, 10, 11 enabled]
0x03\r	[Analog Inputs: A/D 0, 1 enabled]
0x0408\r	[Digital input states: DIO 3, 10 high, DIO 2, 11 low]
0x03D0\r	[Analog input ADIO 0= 0x3D0]
0x0124\r	[Analog input ADIO 1=0x120]

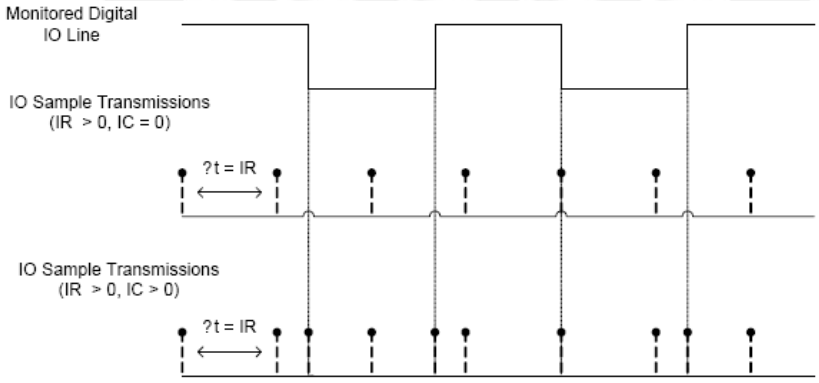
Periodic I/O Sampling

Periodic sampling allows an XBee-PRO module to take an I/O sample and transmit it to a remote device at a periodic rate. The periodic sample rate is set by the IR command. If IR is set to 0, periodic sampling is disabled. For all other values of IR, data will be sampled after IR milliseconds have elapsed and transmitted to a remote device. The DH and DL commands determine the destination address of the IO samples. Only devices with API mode enabled will send IO data samples out their UART. Devices not in API mode will discard received IO data samples.

A module with sleep enabled will transmit periodic I/O samples at the IR rate until the ST time expires and the device can resume sleeping. See the sleep section for more information on sleep.

Digital I/O Change Detection

Modules can be configured to transmit a data sample immediately whenever a monitored digital I/O pin changes state. The IC command is a bitmask that can be used to set which digital I/O lines should be monitored for a state change. If one or more bits in IC is set, an I/O sample will be transmitted as soon as a state change is observed in one of the monitored digital I/O lines. The figure below shows how edge detection can work with periodic sampling.



Enabling Edge Detection will force an immediate sample of all monitored digital IO lines if any digital IO lines change state.

3. Sleep Mode

A number of low-power modes exist to enable modules to operate for extended periods of time on battery power. These sleep modes are enabled with the SM command. The sleep modes are characterized as either asynchronous (SM = 1, 4, 5) or synchronous (SM = 7,8). Asynchronous sleeping modes should not be used in a synchronous sleeping network, and vice versa.

Asynchronous sleep modes can be used to control the sleep state on a module by module basis. Modules operating in an asynchronous sleep mode should not be used to route data. Digi strongly encourages users to set asynchronous sleeping modules as end-devices using the CE command. This will prevent the node from attempting to route data.

The synchronous sleep feature of DigiMesh makes it possible for all nodes in the network to synchronize their sleep and wake times. All synchronized cyclic sleep nodes enter and exit a low power state at the same time. This forms a cyclic sleeping network. Nodes synchronize by receiving a special RF packet called a sync message which is sent by a node acting as a sleep coordinator. A node in the network can become a coordinator through a process called nomination. The sleep coordinator will send one sync message at the beginning of each wake period. The sync message is sent as a broadcast and repeated by every node in the network. The sleep and wake times for the entire network can be changed by locally changing the settings on an individual node. The network will use the most recently set sleep settings.

Sleep Modes

Normal Mode (SM=0)

Normal mode is the default for a newly powered-on node. In this mode, a node will not sleep. Normal mode nodes should be mains-powered.

A normal mode module will synchronize to a sleeping network, but will not observe synchronization data routing rules (it will route data at any time, regardless of the wake state of the network). When synchronized, a normal node will relay sync messages generated by sleep-compatible nodes but will not generate sync messages. Once a normal node has synchronized with a sleeping network, it can be put into a sleep-compatible sleep mode at any time.

Asynchronous Pin Sleep Mode (SM=1)

Pin sleep allows the module to sleep and wake according to the state of the **Sleep_RQ** pin (pin 9). Pin sleep mode is enabled by setting the SM command to 1. When **Sleep_RQ** is asserted (high), the module will finish any transmit or receive operations and enter a low-power state. The module will wake from pin sleep when the **Sleep_RQ** pin is de-asserted (low).

Asynchronous Cyclic Sleep Mode (SM=4)

Cyclic sleep allows the module to sleep for a specified time and wake for a short time to poll. Cyclic sleep mode is enabled by setting the SM command to 4. In cyclic sleep, the module sleeps for a specified time. If the XBee receives serial or RF data while awake, it will then extend the time before it returns to sleep by the amount specified by the ST command. Otherwise, it will enter sleep mode immediately. The **On_SLEEP** line is asserted (high) when the module wakes, and is de-asserted (low) when the module sleeps. If hardware flow control is enabled (D7 command), the **CTS** pin will assert (low) when the module wakes and can receive serial data, and de-assert (high) when the module sleeps.

Asynchronous Cyclic Sleep with Pin Wake Up Mode (SM=5)

(SM=5) is similar to both the (SM=1) and (SM=4) modes. When the **SLEEP_REQUEST** pin is asserted, the module will enter a cyclic sleep mode similar to (SM=4). When the **SLEEP_REQUEST** pin is de-asserted, the module will immediately wake up. The module will not sleep when the **SLEEP_REQUEST** pin is de-asserted.

Synchronous Sleep Support Mode (SM=7)

A node in synchronous sleep support mode will synchronize itself with a sleeping network but will not itself sleep. At any time, the node will respond to new nodes which are attempting to join the sleeping network with a sync message. A sleep support node will only transmit normal data when the other nodes in the sleeping network are awake. Sleep support nodes are especially useful when used as preferred sleep coordinator nodes and as aids in adding new nodes to a sleeping network.

Note: Because sleep support nodes do not sleep, they should be mains powered.

Synchronous Cyclic Sleep Mode (SM=8)

A node in synchronous cyclic sleep mode sleeps for a programmed time, wakes in unison with other nodes, exchanges data and sync messages, and then returns to sleep. While asleep, it cannot receive RF messages or read commands fromt the UART port. Generally, sleep and wake times are specified by the SP and ST respectively of the network's sleep coordinator. These parameters are only used at start up until the node is synchronized with the network. When a module has synchronized with the network, its sleep and wake times can be queried with the OS and OW commands respectively. If D9 = 1 (**ON_SLEEP** enabled) on a cyclic sleep node, the **ON_SLEEP** line will assert when the module is awake and de-assert when the module is asleep. **CTS** is also de-asserted while asleep (D7 = 1). A newly-powered unsynchronized sleeping node will poll for a synchronized message and then sleep for the period specified by SP, repeating this cycle until it becomes synchronized by receiving a sync message. Once a sync message is received, the node will synchronize itself with the network.

Note: All nodes in a synchronous sleep network should be configured to operate in either Synchronous Sleep Support Mode or Synchronous Cyclic Sleep Mode. Asynchronous sleeping nodes are not compatible with synchronous sleep nodes.

Asynchronous Sleep Operation

Wake Timer

In cyclic sleep mode (SM=4 or SM=5), if serial or RF data is received, the module will start a sleep timer (time until sleep). Any data received serially or by RF link will reset the timer. The timer duration can be set using the ST command. The module returns to sleep when the sleep timer expires.

Sleeping Routers

The Sleeping Router feature of DigiMesh makes it possible for all nodes in the network to synchronize their sleep and wake times. All synchronized cyclic sleep nodes enter and exit a low power state at the same time. This forms a cyclic sleeping network. Nodes synchronize by receiving a special RF packet called a sync message which is sent by a node acting as a sleep coordinator. A node in the network can become a sleep coordinator through a process called nomination. The sleep coordinator will send one sync message at the beginning of each wake period. The sync message is sent as a broadcast and repeated by every node in the network. The sleep and wake times for the entire network can be changed by locally changing the settings on an individual node. The network will use the most recently set sleep settings.

One node in a sleeping network acts as the sleeping coordinator. The process by which a node becomes a sleep coordinator is described later in this document. During normal operations, at the beginning of a wake cycle the sleep coordinator will send a sync message as a broadcast to all nodes in the network. This message contains synchronization information and the wake and sleep times for the current cycle. All cyclic sleep nodes receiving a sync message will remain awake for the wake time and then sleep for the sleep period specified.

The sleep coordinator will send one sync message at the beginning of each cycle with the currently configured wake and sleep times. All router nodes which receive this sync message will relay the message to the rest of the network. If the sleep coordinator does not hear a re-broadcast of the sync message by one of its immediate neighbors then it will re-send the message one additional time. It should be noted that if SP or ST are changed, the network will not apply the new settings until the beginning of the next wake time. See the Changing Sleep Parameters section below for more information.

A sleeping router network is robust enough that an individual node can go several cycles without receiving a sync message (due to RF interference, for example). As a node misses sync messages, the time available for transmitting messages in the wake time is reduced to maintain synchronization accuracy. By default, a module will also reduce its active sleep time progressively as sync messages are missed.

Synchronization Messages

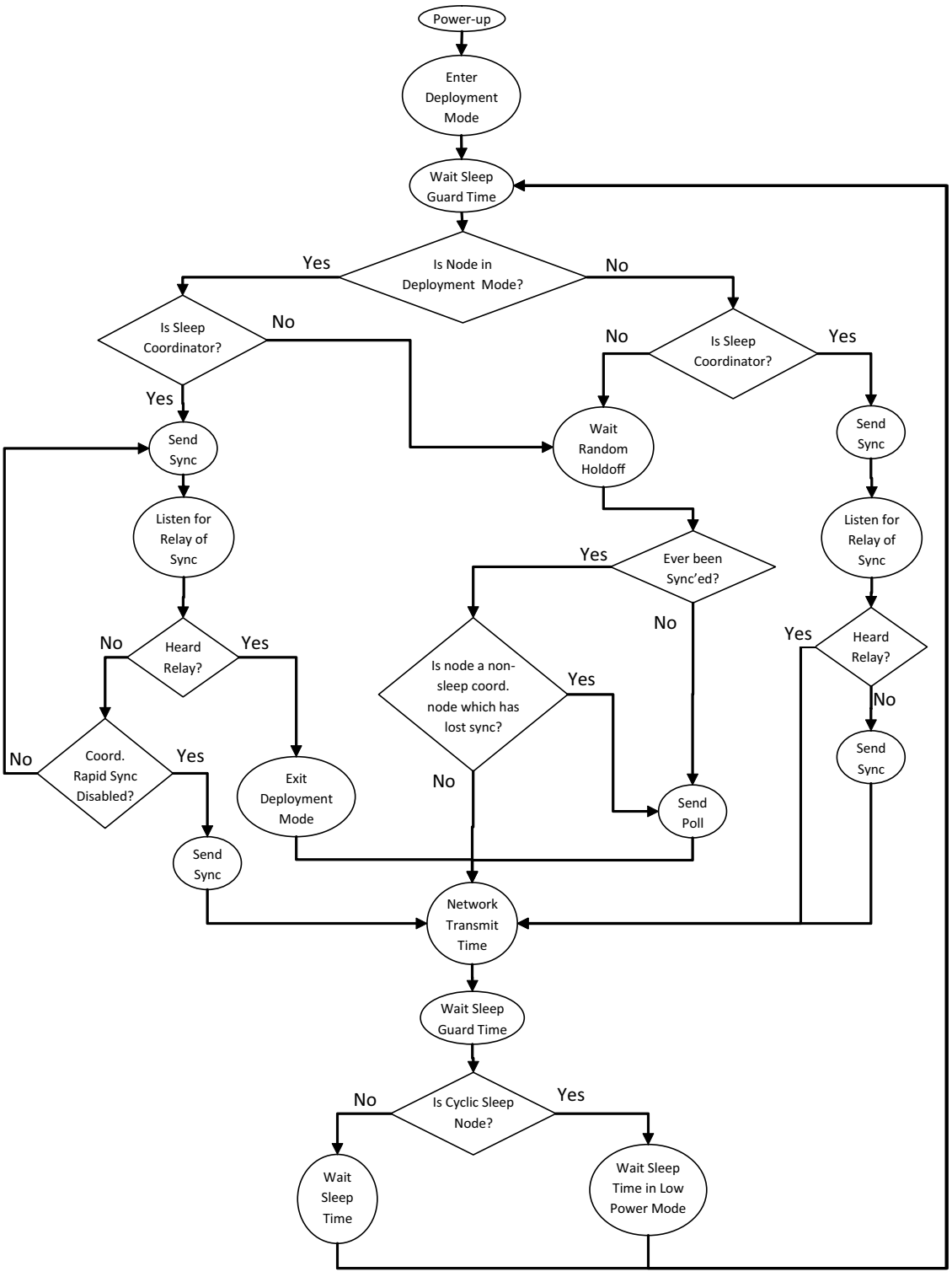
A sleep coordinator will regularly send sync messages to keep the network in sync. Nodes which have not been synchronized or, in some cases, which have lost sync will also send messages requesting sync information.

Deployment mode is used by sleep compatible nodes when they are first powered up and the sync message has not been relayed. A sleep coordinator in deployment mode will rapidly send sync messages until it receives a relay of one of those messages. This allows a network to be deployed more effectively and allows a sleep coordinator which is accidentally or intentionally reset to rapidly re-synchronize with the rest of the network. If a node which has exited deployment mode receives a sync message from a sleep coordinator which is in deployment mode, the sync will be rejected and a corrective sync will be sent to the sleep coordinator. Deployment mode can be disabled using the sleep options command (SO).

A sleep coordinator which is not in deployment mode or which has had deployment mode disabled will send a sync message at the beginning of the wake cycle. The sleep coordinator will then listen for a neighboring node to relay the sync. If the relay is not heard, the sync coordinator will send the sync one additional time.

A node which is not acting as a sleep coordinator which has never been synchronized will send a message requesting sync information at the beginning of its wake cycle. Synchronized nodes which receive one of these messages will respond with a synchronization packet. Nodes which are configured as non-sleep coordinators (using the SO command) which have gone six or more cycles without hearing a sync will also send a message requesting sync at the beginning of their wake period.

The following diagram illustrates the synchronization behavior of sleep compatible modules:


Power-up
Enter Deployment Mode
Wait Sleep Guard Time
Is Node in Deployment Mode?
Yes
No
Is Sleep Coordinator?
No
Yes
Send Sync
Listen for Relay of Sync
Heard Relay?
No
Yes
Coord. Rapid Sync Disabled?
No
Yes
Send Sync
Exit Deployment Mode
Wait Random Holdoff
Is Sleep Coordinator?
No
Yes
Send Sync
Listen for Relay of Sync
Heard Relay?
Yes
No
Send Sync
Ever been Sync'ed?
Yes
No
Is node a non-sleep coord. node which has lost sync?
Yes
No
Send Poll
Network Transmit Time
Wait Sleep Guard Time
Is Cyclic Sleep Node?
No
Yes
Wait Sleep Time
Wait Sleep Time in Low Power Mode

Becoming a Sleep Coordinator

A node can become a sleep coordinator in one of four ways:

Preferred Sleep Coordinator Option

A node can be specified to always act as a sleep coordinator. This is done by setting the preferred sleep coordinator bit (bit 0) in the sleep operations parameter (SO) to 1. A node with the sleep coordinator bit set will always send a sync message at the beginning of a wake cycle. For this reason, it is imperative that no more than one node in the network has this bit set. Although it is not necessary to specify a preferred sleep coordinator, it is often useful to select a node for this purpose to improve network performance. A node which is centrally located in the network can serve as a good sleep coordinator to minimize the number of hops a sync message must take to get across the network. A sleep support node and/or a node which is mains powered may be a good candidate.

The preferred sleep coordinator bit should be used with caution. The advantages of using the option become weaknesses when used on a node that is not positioned or configured properly. The preferred sleep coordinator option can also be used when setting up a network for the first time. When starting a network, a node can be configured as a sleep coordinator so it will begin sending sleep messages. After the network is set up, the preferred sleep coordinator bit can be disabled.

Nomination and Election

Nomination is an optional process that can occur on a node in the event that contact with the network sleep coordinator is lost. By default, this behavior is disabled. This behavior can be enabled with the sleep options command (SO). This process will automatically occur in the event that contact with the previous sleep coordinator is lost. Any sleep compatible node which has this behavior enabled is eligible to become the sleep coordinator for the network. If a sleep compatible node has missed three or more sync messages and is not configured as a non-sleep coordinator (presumably because the sleep coordinator has been disabled) it may become a sleep coordinator. Depending on the platform and other configured options, such a node will eventually nominate itself after a number of cycles without a sync. A nominated node will begin acting as the new network sleep coordinator. It is possible for multiple nodes to nominate themselves as the sleep coordinator. If this occurs, an election will take place to establish seniority among the multiple sleep coordinators. Seniority is determined by four factors (in order of priority):

1. Newer sleep parameters: a node using newer sleep parameters (SP/ST) is considered senior to a node using older sleep parameters. (See the Changing Sleep Parameters section below.)

2. Preferred Sleep Coordinator: a node acting as a preferred sleep coordinator is senior to other nodes.

3. Sleep Support Node: sleep support nodes are senior to cyclic sleep nodes. (This behavior can be modified using the SO parameter.)

4. Serial number: in the event that the above factors do not resolve seniority, the node with the higher serial number is considered senior.

Commissioning Button

The commissioning button can be used to select a module to act as the sleep coordinator. If the commissioning button functionality has been enabled, a node can be immediately nominated as a sleep coordinator by pressing the commissioning button twice or by issuing the CB2 command. A node nominated in this manner is still subject to the election process described above. A node configured as a non-sleep coordinator will ignore commissioning button nomination requests.

Changing Sleep Parameters

Any sleep compatible node in the network which does not have the non-sleep coordinator sleep option set can be used to make changes to the network's sleep and wake times. If a node's SP and/or ST are changed to values different from those that the network is using, that node will become the sleep coordinator. That node will begin sending sync messages with the new sleep parameters at the beginning of the next wake cycle.

Note #1: For normal operations, a module will use the sleep and wake parameters it gets from the sleep sync message, not the ones specified in its SP and ST parameters. The SP and ST parameters are not updated with the values of the sync message. The current network sleep and wake times used by the node can be queried using the OS and OW commands.

Note #2: Changing network parameters can cause a node to become a sleep coordinator and change the sleep settings of the network. The following commands can cause this to occur: NH, NN, NQ, and MR. In most applications, these network parameters should only be configured during deployment.

Sleep Guard Times

To compensate for variations in the timekeeping hardware of the various modules in a sleeping router network, sleep guard times are allocated at the beginning and end of the wake time. The size of the sleep guard time varies based on the sleep and wake times selected and the number of cycles that have elapsed since the last sync message was received. The sleep guard time guarantees that a destination radio will be awake when a transmission is sent. As more and more consecutive sync messages are missed, the sleep guard time increases in duration and decreases the available transmission time.

Auto-Early Wake-Up Sleep Option

Similarly to the sleep guard time, the auto early wake-up option decreases the sleep period based on the number of sync messages missed. This option comes at the expense of battery life. Auto-early wake-up sleep can be disabled using the sleep options (SO) command.

Configuration

Selecting Sleep Parameters

Choosing proper sleep parameters is vital to creating a robust sleep-enabled network with a desireable battery life. To select sleep parameters that will be good for most applications, follow these steps:

1. **Choose NN and NH.** Based on the placement of the nodes in your network, select appropriate values for the Network Hops (NH) and Network Delay Slots (NN) parameters.

Note: the default values of NH and NN have been optimized to work for the majority of deployments. In most cases, we suggest that these parameters not be modified from their default values. Decreasing these parameters for small networks can improve battery life, but care should be taken so that the values are not made too small.

2. **Calculate the Sync Message Propagation Time (SMPT).** This is the maximum amount of time it takes for a sleep synchronization message to propagate to every node in the network. This number can be estimated with the following formula:

SMPT = NN * NH * (MT + 1) 18ms

3. **Select desired duty cycle.** The ratio of sleep time to wake time is the factor that has the greatest effect on the RF module's power consumption. Battery life can be estimated based on the following factors: sleep period, wake time, sleep current, RX current, TX current, and battery capacity.

4. **Choose sleep period and wake time.** The wake time needs to be long enough to transmit the desired data as well as the sync message. The ST parameter will automatically adjust upwards to its minimum value when other AT commands are changed that will affect it (SP, NN, and NH). Use a value larger than this minimum. If a module misses successive sync messages, it reduces its available transmit time to compensate for possible clock drift. Budget a large enough ST time to allow for a few sync messages to be missed and still have time for normal data transmissions.

Starting a Sleeping Network

By default, all new nodes operate in normal (non-sleep) mode. To start a sleeping network, follow these steps:

1. Enable the preferred sleep coordinator option on one of the nodes, and set its SM to a sleep compatible mode (7 or 8) with its SP and ST set to a quick cycle time. The purpose of a quick cycle time is to allow commands to be sent quickly through the network during commissioning.

2. Next, power on the new nodes within range of the sleep coordinator. The nodes will quickly receive a sync message and synchronize themselves to the short cycle SP and ST.

3. Configure the new nodes in their desired sleep mode as cyclic sleeping nodes or sleep support nodes.

4. Set the SP and ST values on the sleep coordinator to the desired values for the deployed network.

5. Wait a cycle for the sleeping nodes to sync themselves to the new SP and ST values.

6. Disable the preferred sleep coordinator option bit on the sleep coordinator (unless a preferred sleep coordinator is desired).

7. Deploy the nodes to their positions.

Alternatively, nodes can be set up with their sleep pre-configured and written to flash (using the WR command) prior to deployment. If this is the case, the commissioning button and associate LED can be used to aid in deployment:

1. If a preferred sleep coordinator is going to be used in the network, deploy it first. If there will be no preferred sleep coordinator, select a node for deployment, power it on and press the commissioning button twice. This will cause the node to begin emitting sync messages.

Verify that the first node is emitting sync messages by watching its associate LED. A slow blink indicates that the node is acting as a sleep coordinator.

2. Next, power on nodes in range of the sleep coordinator or other nodes which have synchronized with the network. If the synchronized node is asleep, it can be woken by pressing the commissioning button once.

3. Wait a cycle for the new node to sync itself.

4. Verify that the node syncs with the network. The associate LED will blink when the module is awake and synchronized.

5. Continue this process until all nodes have been deployed.

Adding a New Node to an Existing Network

To add a new node to the network, the node must receive a sync message from a node already in the network. On power-up, an unsynchronized sleep compatible node will periodically send a broadcast requesting a sync message and then sleep for its SP period. Any node in the network that receives this message will respond with a sync. Because the network can be asleep for extended periods of time, and as such cannot respond to requests for sync messages, there are methods that can be used to sync a new node while the network is asleep.

1. Power the new node on within range of a sleep support node. Sleep support nodes are always awake and will be able to respond to sync requests promptly.

2. A sleeping cyclic sleep node in the network can be woken by the commissioning button. Place the new node in range of the existing cyclic sleep node and wake the existing node by holding down the commissioning button for 2 seconds, or until the node wakes. The existing node stays awake for 30 seconds and will respond to sync requests while it is awake.

If you do not use one of these two methods, you must wait for the network to wake up before adding the new node. The new node should be placed in range of the network with a sleep/wake cycle that is shorter than the wake period of the network. The new node will periodically send sync requests until the network wakes up and it receives a sync message.

Changing Sleep Parameters

Changes to the sleep and wake cycle of the network can be made by selecting any node in the network and changing the SP and/or ST of the node to values different than those the network is currently using. If using a preferred sleep coordinator or if it is known which node is acting as the sleep coordinator, it is suggested that this node be used to make changes to network settings. If

the network sleep coordinator is not known, any node that does not have the non-sleep coordinator sleep option bit set (see the SO command) can be used.

When changes are made to a node's sleep parameters, that node will become the network's sleep coordinator (unless it has the non-sleep coordinator option selected) and will send a sync message with the new sleep settings to the entire network at the beginning of the next wake cycle. The network will immediately begin using the new sleep parameters after this sync is sent.

Changing sleep parameters increases the chances that nodes will lose sync. If a node does not receive the sync message with the new sleep settings, it will continue to operate on its old settings. To minimize the risk of a node losing sync and to facilitate the re-syncing of a node that does lose sync, the following precautions can be taken:

1. Whenever possible, avoid changing sleep parameters.

2. Enable the missed sync early wake up sleep option (SO). This command is used to tell a node to wake up progressively earlier based on the number of cycles it has gone without receiving a sync. This will increase the probability that the un-synced node will be awake when the network wakes up and sends the sync message.

Note: using this sleep option increases reliability but may decrease battery life. Nodes using this sleep option which miss sync messages will have an increased wake time and decreased sleep time during cycles in which the sync message is missed. This will reduce battery conservation.

3. When changing between two sets of sleep settings, choose settings so that the wake periods of the two sleep settings will happen at the same time. In other words, try to satisfy the following equation: (SP1 + ST1) = N * (SP2 + ST2), where SP1/ST1 and SP2/ST2 are the desired sleep settings and N is an integer.

Rejoining Nodes Which Have Lost Sync

Mesh networks get their robustness from taking advantage of routing redundancies which may be available in a network. It is recommended to architect the network with redundant mesh nodes to increase robustness. If a scenario exists such that the only route connecting a subnet to the rest of the network depends on a single node, and that node fails -- or the wireless link fails due to changing environmental conditions (catastrophic failure condition), then multiple subnets may arise while using the same wake and sleep intervals. When this occurs the first task is to repair, replace, and strengthen the weak link with new and/or redundant modules to fix the problem and prevent it from occurring in the future.

When the default DigiMesh sleep parameters are used, separated subnets will not drift out of phase with each other. Subnets can drift out of phase with each other if the network is configured in one of the following ways:

- If multiple modules in the network have had the non-sleep coordinator sleep option bit disabled and are thus eligible to be nominated as a sleep coordinator.
- If the modules in the network are not using the auto early wake-up sleep option.

If a network has multiple subnets that have drifted out of phase with each other, get the subnets back in phase with the following steps:

1. Place a sleep support node in range of both subnets.

2. Select a node in the subnet that you want the other subnet to sync up with. Use this node to slightly change the sleep cycle settings of the network (increment ST, for example).

3. Wait for the subnet's next wake cycle. During this cycle, the node selected to change the sleep cycle parameters will send the new settings to the entire subnet it is in range of, including the sleep support node which is in range of the other subnet.

4. Wait for the out of sync subnet to wake up and send a sync. When the sleep support node receives this sync, it will reject it and send a sync to the subnet with the new sleep settings.

5. The subnets will now be in sync. The sleep support node can be removed. If desired, the sleep cycle settings can be changed back to what they were.

In the case that only a few nodes need to be replaced, this method can also be used:

1. Reset the out of sync node and set its sleep mode to cyclic sleep (SM = 8). Set it up to have a short sleep cycle.

2. Place the node in range of a sleep support node or wake a sleeping node with the commissioning button.

3. The out of sync node will receive a sync from the node which is synchronized to the network and sync to the network sleep settings.

Diagnostics

The following are useful in some applications when managing a sleeping router network:

Query current sleep cycle: the OS and OW command can be used to query the current operational sleep and wake times a module is currently using.

Sleep Status: the SS command can be used to query useful information regarding the sleep status of the module. This command can be used to query if the node is currently acting as a network sleep coordinator, as well as other useful diagnostics.

Missed Sync Messages Command: the MS command can be used to query the number of cycles that have elapsed since the module last received a sync message.

Sleep Status API messages: when enabled with the SO command, a module configured in API mode will output modem status frames immediately after a module wakes up and just prior to a module going to sleep.

4. XBee/XBee-PRO® DigiMesh 2.4

DigiMesh Networking

Mesh networking allows messages to be routed through several different nodes to a final destination. DigiMesh firmware allows manufacturers and system integrators to bolster their networks with the self-healing attributes of mesh networking. In the event that one RF connection between nodes is lost (due to power-loss, environmental obstructions, etc.) critical data can still reach its destination due to the mesh networking capabilities embedded inside the modules.

DigiMesh Feature Set

DigiMesh contains the following features

- **Self-healing**
  Any node may enter or leave the network at any time without causing the network as a whole to fail.
- **Peer-to-peer architecture**
  No hierarchy and no parent-child relationships are needed.
- **Quiet Protocol**
  Routing overhead will be reduced by using a reactive protocol similar to AODV.
- **Route Discovery**
  Rather than maintaining a network map, routes will be discovered and created only when needed.
- **Selective acknowledgements**
  Only the destination node will reply to route requests.
- **Reliable delivery**
  Reliable delivery of data is accomplished by means of acknowledgements.
- **Sleep Modes**
  Low power sleep modes with synchronized wake are supported with variable sleep and wake times.

Networking Concepts

Device Configuration

DigiMesh modules can be configured to act as routers or end devices with the CE command. By default all modules in a DigiMesh network act as routers. Modules configured as routers will actively relay network unicast and broadcast traffic as described below.

Network ID

DigiMesh networks are defined with a unique network identifier. This identifier is set with the ID command. For modules to communicate they must be configured with the same network identifier. The ID parameter allows multiple DigiMesh networks to co-exist on the same physical channel.

Operating Channel

DigiMesh modules utilize direct-sequence spread spectrum modulation and operate on a fixed channel. There are 16 operating channels defined in the 2.4 GHz frequency band. XBee modules support all 16 channels and XBee-PRO modules support 12 of the 16 channels. The operating channel on a module is selected using the CH command.

For modules to communicate, the channel (CH) and network identifier (ID) must be equal on all modules in the network.

Data Transmission and Routing

Unicast Addressing

When transmitting while using Unicast communications, reliable delivery of data is accomplished using retries and acknowledgements. The number of retries is determined by the NR (Network Retries) parameter. RF data packets are sent up to NR + 1 times and ACKs (acknowledgements) are transmitted by the receiving node upon receipt. If a network ACK is not received within the time it would take for a packet to traverse the network twice, a retransmission occurs.

To send Unicast messages, set the DH and DL on the transmitting module to match the corresponding SH and SL parameter values on the receiving module.

Broadcast Addressing

Broadcast transmissions will be received and repeated by all routers in the network. Because ACKs are not used the originating node will send the broadcast multiple times. By default a broadcast transmission is sent four times. Essentially the extra transmissions become automatic retries without acknowledgments. This will result in all nodes repeating the transmission four times as well. In order to avoid RF packet collisions, a random delay is inserted before each router relays the broadcast message. (See NN parameter for details on changing this random delay time.) Sending frequent broadcast transmissions can quickly reduce the available network bandwidth and as such should be used sparingly.

The broadcast address is a 64 bit address with the lowest 16 bits set to 1. The upper bits are set to 0. To send a broadcast transmission set DH to 0 and DL to 0xFFFF. In API mode the destination address would be set to 0x000000000000FFFF.

Routing

A module within a mesh network is able to determine reliable routes using a routing algorithm and table. The routing algorithm uses a reactive method derived from AODV (Ad-hoc On-demand Distance Vector). An associative routing table is used to map a destination node address with its next hop. By sending a message to the next hop address, either the message will reach its destination or be forwarded to an intermediate router which will route the message on to its destination. A message with a broadcast address is broadcast to all neighbors. All routers receiving the message will rebroadcast the message MT+1 times and eventually the message will reach all corners of the network. Packet tracking prevents a node from resending a broadcast message more than MT+1 times.

Route Discovery

If the source node doesn't have a route to the requested destination, the packet is queued to await a route discovery (RD) process. This process is also used when a route fails. A route fails when the source node uses up its network retries without ever receiving an ACK. This results in the source node initiating RD.

RD begins by the source node broadcasting a route request (RREQ). Any router that receives the RREQ that is not the ultimate destination is called an intermediate node.

Intermediate nodes may either drop or forward a RREQ, depending on whether the new RREQ has a better route back to the source node. If so, information from the RREQ is saved and the RREQ is updated and broadcast. When the ultimate destination receives the RREQ, it unicasts a route reply (RREP) back to the source node along the path of the RREQ. This is done regardless of route quality and regardless of how many times an RREQ has been seen before.

This allows the source node to receive multiple route replies. The source node selects the route with the best round trip route quality, which it will use for the queued packet and for subsequent packets with the same destination address.

Throughput

Throughput in a DigiMesh network can vary by a number of variables, including: number of hops, encryption enabled/ disabled, sleeping end devices, failures/route discoveries. Our empirical testing showed the following throughput performance in a robust operating environment (low interference).

Configuration	Data Throughput
1 hop, Encryption Disabled	27.0 kbps
3 hop, Encryption Disabled	10.9 kbps
6 hop, Encryption Disabled	5.78 kbps
1 hop, Encryption Enabled	20.5 kbps
3 hop, Encryption Enabled	9.81 kbps
6 hop, Encryption Enabled	4.7 kbps

Note: Data throughput measurements were made setting the serial interface rate to 115200 bps, and measuring the time to send 100,000 bytes from source to destination. During the test, no route discoveries or failures occurred.

Transmission Timeouts

When a node receives an API TX Request (API configured modules) or an RO timeout occurs (modules configured for Transparent Mode) the time required to route the data to its destination depends on a number of configured parameters, whether the transmission is a unicast or a broadcast, and if the route to the destination address is known. Timeouts or timing information is provided for the following transmission types:

- Transmitting a broadcast
- Transmitting a unicast with a known route
- Transmitting a unicast with an unknown route
- Transmitting a unicast with a broken route.

Note: The timeouts in this section are theoretical timeouts and not precisely accurate. The application should pad the calculated maximum timeouts by a few hundred milliseconds. When using API mode, Tx Status API packets should be the primary method of determining if a transmission has completed.

Unicast One Hop Time

A building block of many of the calculations presented below is the unicastOneHopTime. As its name indicates, it represents the amount of time it takes to send a unicast transmission between two adjacent nodes. It can be queried with the %H command. It is defined as follows:

unicastOneHopTime=%H

Transmitting a broadcast

A broadcast transmission must be relayed by all routers in the network. The maximum delay would be when the sender and receiver are on the opposite ends of the network. The NH and %H parameters define the maximum broadcast delay as follows:

BroadcastTxTime=NH*%H

Transmitting a unicast with a known route

When a route to a destination node is known the transmission time is largely a function of the number of hops and retries. The timeout associated with a unicast assumes the maximum number of hops is necessary (as specified by NH). The timeout can be estimated in the following manner:

knownRouteUnicast=2*NH*MR*unicastOneHopTime

Transmitting a unicast with an unknown route

If the route to the destination is not known the transmitting module will begin by sending a route discovery. If the route discovery is successful and a route is found then the data is transmitted. The timeout associated with the entire operation can be estimated as follows:

unknownRouteUnicast=BroadcastTxTime+(NH*unicastOneHopTime) +knownRouteUnicast

Transmitting a unicast with a broken route

If the route to a destination node has changed since the last time a route discovery was completed a node will begin by attempting to send the data along the previous route. After it fails a route discovery will be initiated and, upon completion of the route discovery, the data will be transmitted along the new route. The timeout associated with the entire operation can be estimated as follows:

brokenRouteUnicast=BroadcastTxTime+(NH*unicastOneHopTime) +(2*knownRouteUnicast)

5. Command Reference Tables

Special

Special Commands

AT Command	Name and Description	Parameter Range	Default
AC	**Apply Changes**. Immediately applies new settings without exiting command mode.	--	--
FR	**Software Reset**. Reset module. Responds immediately with an "OK" then performs a reset 100ms later.	--	--
RE	**Restore Defaults**. Restore module parameters to factory defaults.	--	--
WR	**Write**. Write parameter values to non-volatile memory so that parameter modifications persist through subsequent resets. Note: Once WR is issued, no additional characters should be sent to the module until after the "OK\r" response is received.	--	--

MAC/PHY Level

MAC/PHY-level Commands

AT Command	Name and Description	Parameter Range	Default
CH	**Channel**. Set/Read the channel number used for transmitting and receiving data between RF modules (uses 802.15.4 protocol channel numbers).	0x0B - 0x1A (XBee) 0x0C - 0x17 (XBee-PRO)	0x0C (12d)
ID	**Network ID**. Set or read the user network identifier. Nodes must have the same network identifier to communicate. Changes to ID can be written to non-volatile memory using the WR command.	0x0000 to 0x7FFF	0x7FFF
MT	**Broadcast Multi-Transmit**. Set/Read the number of additional MAC-level broadcast transmissions. All broadcast packets are transmitted MT+1 times to ensure it is received.	0-0xF	3
PL	**Power Level**. Set/Read the power level at which the RF module transmits conducted power.	XBee 0 = -7dBm 1 = -1.7dBm 2 = -0.77dBm 3 = 0.62dBm 4 = 1.42dBm XBee-Pro 0 = 10dBm 1 = 12dBm 2 = 14dBm 3 = 16dBm 4 = 18dBm	4
RR	**Unicast Mac Retries**. Set/Read the maximum number of MAC level packet delivery attempts for unicasts. If RR is non-zero packets sent from the radio will request an acknowledgement, and can be resent up to RR times if no acknowledgements are received.	0-0xF	10
ED	**Energy Detect**. Start an Energy Detect Scan. This parameter is the time in milliseconds to scan all channels. The Module will loop through all the channels until the time elapses. The maximal energy on each channel is returned, and each value is followed by a comma with the list ending with a charriage return. The values returned reflect the detected energy level in units of -dBm.	0-0xFFFF	

Diagnostics

Diagnostics Commands - MAC Statistics and Timeouts

AT Command	Name and Description	Parameter Range	Default
BC	**Bytes Transmitted**. The number of RF bytes transmitted. This count is incremented for every PHY-level byte transmitted. The purpose of this count is to estimate battery life by tracking time spent doing transmissions. This number rolls over to zero from 0xFFFF. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
DB	**Received Signal Strength.** This command reports the received signal strength of the last received RF data packet. The DB command only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link. The DB command value is measured in -dBm. For example if DB returns 0x60, then the RSSI of the last packet received was -96dBm.	n/a	n/a
GD	**Good packets**. Read the number of good frames with valid MAC headers that are received on the RF interface. When the value reaches 0xFFFF, it stays there.	n/a	n/a
EA	**MAC ACK Timeouts**. This count is incremented whenever a MAC ACK timeout occurs on a MAC-level unicast. Once the number reaches 0xFFFF, further events will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
TR	**Transmission Errors**. Read the number of MAC frames that exhaust MAC retries without ever receiving a MAC acknowledgement message from the adjacent node. When the value reaches 0xffff, it stays there.	n/a	n/a
UA	**MAC Unicast Transmission Count**. This count is incremented whenever a MAC unicast transmission occurs for which an ACK is requested. Once the number reaches 0xFFFF, further evens will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
%H	**MAC Unicast One Hop Time**. The MAC unicast one hop timeout in milliseconds. Changing MAC parameters can change this value.	[read-only]	
%8	**MAC Broadcast One Hop Time**. The MAC broadcast one hop timeout in milliseconds. Changing MAC parameters can change this value.	[read-only]	

Network

Network Commands: DigiMesh

AT Command	Name and Description	Parameter Range	Default
CE	**Node Type**. Set/read the node networking type. A module set as an end device will not propagate broadcasts and won't become and intermediate node on a route.	0 - Router 2 - End Device	0
BH	**Broadcast Radius**. Set/read the transmission radius for broadcast data transmissions. Set to 0 for maximum radius. If BH is set greater than NH then the value of NH is used.	0-0x20	0
NH	**Network Hops** Set or read the maximum number of hops expected to be seen in a network route. This value doesn't limit the number of hops allowed, but it is used to calculate timeouts waiting for network acknowledgements.	1 to 0x20	7
NN	**Network Delay Slots** Set or read the maximum random number of network delay slots before rebroadcasting a network packet. One network delay slot is approximately 13ms.	1 to 0x0A	3
MR	**Mesh Network Retries** Set or read the maximum number of network packet delivery attempts. If MR is non-zero, packets sent will request a network acknowledgement, and can be resent up to MR+1 times if no acknowledgements are received.	0 to 7	1

Addressing

Addressing Commands

AT Command	Name and Description	Parameter Range	Default
SH	**Serial Number High**. Read high 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled. This value is read-only and it never changes	0-0xFFFFFFFF	Factory
SL	**Serial Number Low**. Read low 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled. This is read only and it is also the serial number of the node..	0-0xFFFFFFFF	Factory
DH	**Destination Address High**. Set/Get the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the destination address used for transmission.	0 to 0xFFFFFFFF	0
DL	**Destination Address Low**. Set/Get the lower 32 bits of the 64-bit destination address. When combined with DH, DL defines the destination address used for transmission.	0 to 0xFFFFFFFF	0x0000FFFF

Addressing Commands

AT Command	Name and Description	Parameter Range	Default
NI	**Node Identifier**. Stores a string identifier. The string accepts only printable ASCII data In AT Command Mode, the string can not start with a space. A carriage return or comma ends the command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ATND (Network Discover) command. This identifier is also used with the ATDN (Destination Node) command.	up to 20 byte ASCII string	a space character
NT	**Node Discover Timeout.** Set/Read the amount of time a node will spend discovering other nodes when ND or DN is issued.	0 - 0xFC [x 100 msec]	0x82 (130d)
NO	**Network Discovery Options**. Set/Read the options value for the network discovery command. The options bitfield value can change the behavior of the ND (network discovery) command and/or change what optional values are returned in any received ND responses or API node identification frames. Options include: 0x01 = Append DD value (to ND responses or API node identification frames) 0x02 = Local device sends ND response frame when ND is issued. 0x04 = Append RSSI (of the last hop for DigiMesh networks) to ND or FN responses or API node identification frames.	0-0x07 [bitfield]	0
CI	**Cluster Identifier**. Set/read application layer cluster ID value. This value will be used as the cluster ID for all data transmissions. The default value 0x11 (Transparent data cluster ID)	0-0xFFFF	0x11
DE	**Destination Endpoint**. Set/read application layer destination ID value. This value will be used as the destination endpoint for all data transmissions. The default value (0xE8) is the Digi data endpoint.	0-0xFF	0xE8
SE	**Source Endpoint**. Set/read the application layer source endpoint value. This value will be used as the source endpoint for all data transmissions. The default value 0xE8 (Data endpoint) is the Digi data endpoint	0-0xFF	0xE8

Addressing Discovery/Configuration

Addressing Discovery/Configuration Commands

AT Command	Name and Description	Parameter Range	Default
AG	**Aggregator Support.** The AG command sends a broadcast through the network that has the following effects on nodes which receive the broadcast: - The receiving node will establish a DigiMesh route back to the originating node, provided there is space in the rounting table. - The DH and DL of the receiving node will be updated to the address of the originating node if the AG parameter matches the current DH/DL of the receiving node. - For API-enabled modules on which DH and DL are updated, an aggregate Addressing Update frame will be sent out the serial port. Note that the AG command is only available on products that support DigiMesh.	Any 64-bit number	n/a
DN	**Discover Node - Destination Node**. Resolves an NI (Node Identifier) string to a physical address (case sensitive). The following events occur after the destination node is discovered: <AT Firmware> 1. DL & DH are set to the extended (64-bit) address of the module with the matching NI (Node Identifier) string. 2. OK (or ERROR)\r is returned. 3. Command Mode is exited to allow immediate communication <API Firmware> 0xFFFE and 64-bit extended addresses are returned in an API Command Response frame. If there is no response from a module within (NT * 100) milliseconds or a parameter is not specified (left blank), the command is terminated and an "ERROR" message is returned. In the case of an ERROR, Command Mode is not exited.	20 byte ascii string	
ND	**Network Discover**. Discovers and reports all RF modules found. If the ND command is sent through a local API frame, each response is returned as a separate Local or Remote AT Command Response API packet, respectively.		

Addressing Discovery/Configuration Commands

AT Command	Name and Description	Parameter Range	Default
FN	**Find Neighbors**. Discovers and reports all RF modules found within immediate RF range. The following information is reported for each module discovered. MY<CR> (always 0xFFFE) SH<CR> SL<CR> NI<CR> (Variable length) PARENT_NETWORK ADDRESS<CR> (2 Bytes) (always 0xFFFE) DEVICE_TYPE<CR> (1 Byte: 0=Coord, 1=Router, 2=End Device) STATUS<CR> (1 Byte: Reserved) PROFILE_ID<CR> (2 Bytes) MANUFACTURER_ID<CR> (2 Bytes) DIGI DEVICE TYPE<CR> (4 Bytes. Optionally included based on NO settings.) RSSI OF LAST HOP<DR> (1 Byte. Optionally included based on NO settings.) <CR> If the FN command is issued in command mode, after (NT*100) ms + overhead time, the command ends by returning a <CR>. If the FN command is sent through a local API frame, each response is returned as a separate Local or Remote AT Command Response API packet, respectively. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a "0x00" null character.		

Security

Security Commands

AT Command	Name and Description	Parameter Range	Default
EE	**Security Enable** Enables or disables 128-bit AES encryption. This command parameter should be set the same on all devices.	0 to 1	0
KY	**Security Key** Sets the 16 byte network security key value. This command is write-only. Attempts to read KY will return an OK status. This command parameter should be set the same on all devices.	128-bit value	n/a

Serial Interfacing

Serial Interfacing Commands

AT Command	Name and Description	Parameter Range	Default
BD	**Baud rate**. Set or read serial interface rate (speed for data transfer between radio modem and host). Values from 0-8 select preset standard rates. Values at 0x7A and above select the actual baud rate, providing the host supports it. The values from 0 to 8 are interpreted as follows: **0** - 1,200bps **3** - 9,600bps **6** - 57,600bps **1** - 2,400bps **4** - 19,200bps **7** - 111,111bps **2** - 4,800bps **5** - 38,400bps	0 to 7, and 0x39 to 0xF4240	0x03 (9600 bps)
NB	**Parity**. Set or read parity settings for UART communications. The values from 0 to 4 are interpreted as follows: **0** No parity **3** Forced high parity **1** Even parity **4** Forced low parity **2** Odd parity	0 to 4	0 (No parity)
RO	**Packetization Timeout**. Set/Read number of character times of inter-character silence required before packetization. Set (RO=0) to transmit characters as they arrive instead of buffering them into one RF packet.	0 - 0xFF [x character times]	3
FT	**Flow Control Threshhold**. Set or read flow control threshhold. De-assert CTS and/or send XOFF when FT bytes are in the UART receive buffer. Re-assert CTS when less than FT - 16 bytes are in the UART receive buffer.	0x11 - 0xEE	0xBE=190d
AP	**API mode**. Set or read the API mode of the radio. The following settings are allowed: **0** API mode is off. All UART input and output is raw data and packets are delineated using the RO parameter. **1** API mode is on. All UART input and output data is packetized in the API format, without escape sequences. **2** API mode is on with escaped sequences inserted to allow for control characters (XON, XOFF, escape, and the 0x7e delimiter to be passed as data.)	0, 1, or 2	0
AO	**API Output Format**. Enables different API output frames. Options include: **0** Standard Data Frames (0x90 for RF RX) **1** Explicit Addressing Data Frames (0x91 for RF RX)	0, 1	0

I/O Settings

I/O Settings and Commands

AT Command	Name and Description	Parameter Range	Default
CB	**Commissioning Pushbutton**. This command can be used to simulate commissioning button presses in software. The parameter value should be set to the number of button presses to be simulated. For example, sending the ATCB1 command will execute the action associated with 1 commissioning button press.	0-4	n/a
D0	**AD0/DIO0 Configuration**. Configure options for the AD0/DIO0 line of the module. Options include: 0 = Input, unmonitored 1 = Commissioning button enable 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0 - 5	1
D1	**AD1/DIO1 Configuration**. Configure options for the AD1/DIO1 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0

I/O Settings and Commands

AT Command	Name and Description	Parameter Range	Default
D2	**AD2/DIO2 Configuration**. Configure options for the AD2/DIO2 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
D3	**AD3/DIO3 Configuration**. Configure options for the AD3/DIO3 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
D4	**AD4/DIO4 Configuration**. Configure options for the AD4/DIO4 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
D5	**AD5/DIO5 Configuration**. Configure options for the AD5/DIO5 line of the module. Options include: 0 = Input, unmonitored 1 = Associate LED 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0-5	1
D6	**DIO6 Configuration**. Configure options for the DIO6 line of the module. Options include: 0 = Input, unmonitored 1 = RTS flow control 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0-1, 3-5	0
D7	**DIO7 Configuration**. Configure options for the DIO7 line of the module. Options include: 0 = Input, unmonitored 1 = CTS flow control 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high 6 = RS-485 Tx enable, low TX (0V on transmit, high when idle) 7 = RS-485 Tx enable, high TX (high on transmit, 0V when idle)	0-1, 3-7	1
D8	**DIO8/SLEEP_RQ Configuration**. Configure options for the DIO8/SLEEP_RQ line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high When used as SLEEP_RQ, the D8 parameter should be configured in mode 0 or 3.	0,3-5	0
D9	**DIO9 / ON/$\overline{\text{SLEEP}}$.** Configuration. Configure options for the DIO9/ON/$\overline{\text{SLEEP}}$ line of the module. Options include: 0 = Input, unmonitored 1 = ON/$\overline{\text{SLEEP}}$ 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0,1,3-5	

I/O Settings and Commands

AT Command	Name and Description	Parameter Range	Default
P0	**DIO10/PWM0 Configuration**. Configure options for the DIO10/PWM0 line of the module. Options include: 0 = Input, unmonitored 1 = RSSI PWM 2 = PWM0 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0-5	1
P1	**DIO11/PWM1 Configuration**. Configure options for the DIO11/PWM1 line of the module. Options include: 0 = Input, unmonitored 2 = PWM1 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
P2	**DIO12 Configuration**. Configure options for the DIO12 line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 3-5	0
PR	**Pull-up Resistor.** Set/read the bit field that configures the internal pull-up resistor status for the I/O lines. "1" specifies the pull-up resistor is enabled. "0" specifies no pullup. Bits: 0 - DIO4/AD4 (Pin 11) 1 - AD3 / DIO3 (Pin 17) 2 - AD2 / DIO2 (Pin 18) 3 - AD1 / DIO1 (Pin 19) 4 - AD0 / DIO0 (Pin 20) 5 - RTS / DIO6 (Pin 16) 6 - DTR / SLEEP_RQ/DI08 / DIO8 (Pin 9) 7 - DIN / Config (Pin 3) 8 - Associate / DIO5 (Pin 15) 9 - On/Sleep / DIO9 (Pin 13) 10 - DIO12 (Pin 4) 11 - PWM0 / RSSI / DIO10 (Pin 6) 12 - PWM1 / DIO11 (Pin 7) 13 - DIO7/CTS (Pin 12) 14 - DOUT (Pin 2)	0 - 0x1FFF	0x1FFF
M0	**PWM0 Output Level**. Set/read the output level of the PWM0 line. The line should be configured as a PWM output using the P0 command.	0-0x03FF	0
M1	**PWM1 Output Level**. Set/read the output level of the PWM1 line. The line should be configured as a PWM output using the P1 command.	0-0x03FF	0
LT	**Assoc LED Blink Time**. Set/Read the Associate LED blink time. If the Associate LED functionality is enabled (D5 command), this value determines the on and off blink times for the LED. If LT=0, the default blink rate will be used (500ms sleep coordinator, 250ms otherwise). For all other LT values, LT is measured in 10ms	0x14-0xFF (x10ms)	0
RP	**RSSI PWM Timer.** Time RSSI signal will be output after last transmission. When RP = 0xFF, output will always be on.	0 - 0xFF [x 100 ms]	0x28 (4 seconds)

I/O Sampling

I/O Sampling Commands

AT Command	Name and Description	Parameter Range	Default
IC	**I/O Digital Change Detection**. Set/Read the digital I/O pins to monitor for changes in the I/O state. IC works with the individual pin configuration commands (D0-D9, P0-P2). If a pin is enabled as a digital input/output, the IC command can be used to force an immediate I/O sample transmission when the DIO state changes. IC is a bitmask that can be used to enable or disable edge detection on individual channels. Unused bits should be set to 0. Bit (I/O pin): 0 (DIO0) 1 (DIO1) 2 (DIO2) 3 (DIO3) 4 (DIO4) 5 (DIO5) 6 (DIO6) 7 (DIO7) 8 (DIO8) 9 (DIO9) 10 (DIO10) 11 (DIO11) 12 (DIO12)	0-0xFFFF	0
IF	**Sleep Sample Rate**. Set/read the number of sleep cycles that must elapse between periodic I/O samples. This allows I/O samples to be taken only during some wake cycles. During those cycles I/O samples are taken at the rate specified by IR.	1-0xFF	1
IR	**IO Sample Rate**. Set/Read the I/O sample rate to enable periodic sampling. For periodic sampling to be enabled, IR must be set to a non-zero value, and at least one module pin must have analog or digital I/O functionality enabled (see D0-D9, P0-P2 commands). The sample rate is measured in milliseconds.	0 - 0xFFFF (ms)	0
IS	**Force Sample**. Forces a read of all enabled digital and analog input lines.	n/a	n/a
1S	**XBee Sensor Sample**. Forces a sample to be taken on an XBee Sensor device. This command can only be issued to an XBee Sensor device using an API remote command.	-	-

Sleep

Sleep Commands

AT Command	Name and Description	Parameter Range	Default
SM	**Sleep Mode**. Set/read the sleep mode of the module. 0 - No sleep mode enabled 1 - Pin sleep. In this mode, the sleep/wake state of the module is controlled by the SLEEP_RQ line. 4 - Asynchronous cyclic sleep. In this mode, the module periodically sleeps and wakes based on the SP and ST commands. 5 - Asynchronous cyclic sleep with pin wake-up. When the **SLEEP_REQUEST** pin is asserted, the module will enter a cyclic sleep mode similar to (SM=4). When the **SLEEP_REQUEST** pin is de-asserted, the module will immediately wake up. The module will not sleep when the **SLEEP_REQUEST** pin is de-asserted. 7 - Sleep support mode. 8 - Synchronous cyclic sleep mode.	0, 1, 4, 5, 7, 8	0
SO	**Sleep Options**. Set/read the sleep options of the module. This command is a bitmask. For synchronous sleep modules, the following sleep options are defined: bit 0 = Preferred sleep coordinator bit 1 = Non-sleep coordinator bit 2 = Enable API sleep status messages bit 3 = Disable early wake-up bit 4 = Enable node type equality bit 5 = Disable lone coordinator sync repeat For ansynchronous sleep modules, the following sleep options are defined: bit 8 = Always wake for ST time	Any of the available sleep option bits can be set or cleared. Bit 0 and bit 1 cannot be set at the same time.	0x02

AT Command	Name and Description	Parameter Range	Default
SN	**Number of Sleep Periods.** Set/read the number of sleep periods value. This command controls the number of sleep periods that must elapse between assertions of the ON_SLEEP line during the wake time of asynchronous cyclic sleep. During cycles when the ON_SLEEP line is not asserted, the module will wake up and check for any serial or RF data. If any such data is received, then the ON_SLEEP line will be asserted and the module will fully wake up. Otherwise, the module will return to sleep after checking. This command does not work with synchronous sleep modules.	1 - 0xFFFF	1
SP	**Sleep Period**. Set/read the sleep period of the module. This command defines the amount of time the module will sleep per cycle.	1 - 1440000 (x 10 ms)	2 seconds
ST	**Wake Time**. Set/read the wake period of the module. For asynchronous sleep modules, this command defines the amount of time that the module will stay awake after receiving RF or serial data. For synchronous sleep modules, this command defines the amount of time that the module will stay awake when operating in cyclic sleep mode. This value will be adjusted upwards automatically if it is too small to function properly based on other settings.	0x45-0x36EE80	0x7D0 (2 seconds)
WH	**Wake Host**. Set/Read the wake host timer value. If the wake host timer is set to a non-zero value, this timer specifies a time (in millisecond units) that the device should allow after waking from sleep before sending data out the UART or transmitting an I/O sample. If serial characters are received, the WH timer is stopped immediately. When in synchronous sleep, the device will shorten its sleep period by the value specified by the WH command to ensure that it is prepared to communicate when the network wakes up. When in this this sleep mode, the device will always stay awake for the WH time plus the amount of time it takes to transmit a one-hop unicast to another node.	0-0xFFFF (x 1ms)	0

Sleep Diagnostics

Diagnostics - Sleep Status Timing

AT Command	Name and Description	Parameter Range	Default
SS	**Sleep Status** The SS command can be used to query a number of Boolean values describing the status of the module. Bit 0: This bit will be true when the network is in its wake state. Bit 1: This bit will be true if the node is currently acting as a network sleep coordinator. Bit 2: This bit will be true if the node has ever received a valid sync message since the time it was powered on. Bit 3: This bit will be true if the node has received a sync message in the current wake cycle. Bit 4: This bit will be true if the user has altered the sleep settings on the module so that the node will nominate itself and send a sync message with the new settings at the beginning of the next wake cycle. Bit 5: This bit will be true if the user has requested that the node nominate itself as the sleep coordinator (using the commissioning button or the CB2 command). All other bits: Reserved - All non-documented bits can be any value and should be ignored. bit 6 = This bit will be true if the node is currently in deployment mode. All other bits: Reserved - All non-documented bits can be any value and should be ignored.	n/a	n/a
OS	**Operational Sleep Period**. Read the sleep period that the node is currently using. This number will oftentimes be different from the SP parameter if the node has synchronized with a sleeping router network. Units of 10mSec	n/a	n/a
OW	**Operational Wake Period**. Read the wake time that the node is currently using. This number will oftentimes be different from the ST parameter if the node has synchronized with a sleeping router network. Units of 1 ms	n/a	n/a
MS	**Number of Missed Syncs**. Read the number of wake cycles that have elapsed since the last sync message was received.		

AT Command	Name and Description	Parameter Range	Default
SQ	**Missed Sync Count**. Count of the number of syncs that have been missed. This value can be reset by setting ATSQ to 0. When the value reaches 0xFFFF it will not be incremented anymore.	n/a	n/a

AT Command Options

AT Command Options Commands

AT Command	Name and Description	Parameter Range	Default
CC	**Command Character**. Set or read the character to be used between guard times of the AT Command Mode Sequence. The AT Command Mode Sequence causes the radio modem to enter Command Mode (from Idle Mode).	0 - 0xFF	0x2B
CT	**Command Mode Timeout.** Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	2-0x1770	0x64 (100d)
CN	**Exit Command Mode.** Explicitly exit the module from AT Command Mode.	--	--
GT	**Guard Times**. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT + CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	0 to 0xFFFF	0x3E8 (1000d)

Firmware Commands

Firmware Version/Information

AT Command	Name and Description	Parameter Range	Default
VL	**Version Long**. Shows detailed version information including application build date and time.	--	--
VR	**Firmware Version**. Read firmware version of the module.	0 - 0xFFFFFFFF [read-only]	Firmware-set
HV	**Hardware Version**. Read hardware version of the module.	0 - 0xFFFF [read-only]	Factory-set
DD	**Device Type Identifier.** Stores a device type value. This value can be used to differentiate multiple XBee-based products.	0-0xFFFFFFFF [read only]	0x40000
NP	**Maximum RF Payload Bytes**. This value returns the maximum number of RF payload bytes that can be sent in a unicast transmission based on the current configurations.	0-0xFFFF	n/a
CK	**Configuration Code**. Read the configuration code associated with the current AT command configuration.The code returned can be used as a quick check to determine if a node has been configured as desired.	0-0xFFFFFFFF	n/a

7. API Operation

As an alternative to Transparent Operation, API (Application Programming Interface) Operations are available. API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a UART Data Frame.

Please note that Digi may add new frame types to future versions of firmware, so please build into your software interface the ability to filter out additional API frames with unknown Frame Types.

API Frame Specifications

Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

- AP = 1: API Operation
- AP = 2: API Operation (with escaped characters)

API Operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the UART data frame structure is defined as follows:

Figure 4-01. UART Data Frame Structure:

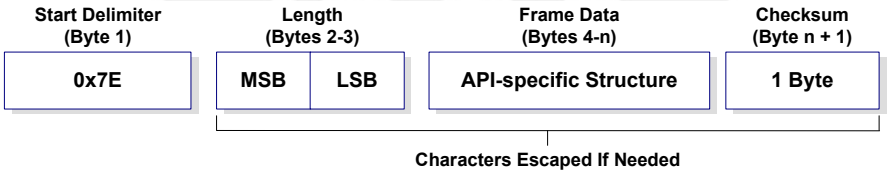
MSB = Most Significant Byte, LSB = Least Significant Byte

Any data received prior to the start delimiter is silently discarded. If the frame is not received correctly or if the checksum fails, the module will discard the packet.

API Operation - with Escape Characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the UART data frame structure is defined as follows:

Figure 4-02. UART Data Frame Structure - with escape control characters:



MSB = Most Significant Byte, LSB = Least Significant Byte

Escape characters. When sending or receiving a UART data frame, specific data values must be escaped (flagged) so they do not interfere with the data frame sequencing. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20.

Data bytes that need to be escaped:

- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Example - Raw UART Data Frame (before escaping interfering bytes):
0x7E 0x00 0x02 0x23 0x11 0xCB

0x11 needs to be escaped which results in the following frame:
0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

Note: In the above example, the length of the raw data (excluding the checksum) is 0x0002 and the checksum of the non-escaped data (excluding frame delimiter and length) is calculated as: 0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB.

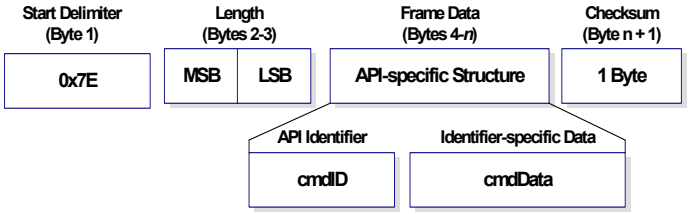
Length

The length field has two-byte value that specifies the number of bytes that will be contained in the frame data field. It does not include the checksum field.

Frame Data

Frame data of the UART data frame forms an API-specific structure as follows:

Figure 4-03. UART Data Frame & API-specific Structure:



The cmdID frame (API-identifier) indicates which API messages will be contained in the cmdData frame (Identifier-specific data). Note that multi-byte values are sent big endian.The XBee modules support the following API frames:

API Frame Names and Values

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
Transmit Request	0x10
Explicit Addressing Command Frame	0x11
Remote Command Request	0x17
AT Command Response	0x88
Modem Status	0x8A
Transmit Status	0x8B
Receive Packet (AO=0)	0x90
Explicit Rx Indicator (AO=1)	0x91
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97

Checksum

To test data integrity, a checksum is calculated and verified on non-escaped data.

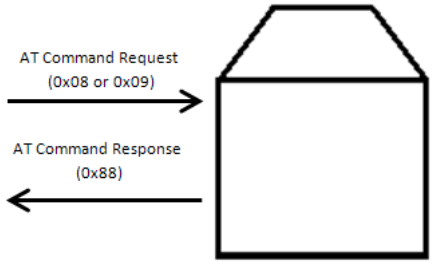
To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.

To verify: Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

API UART Exchanges

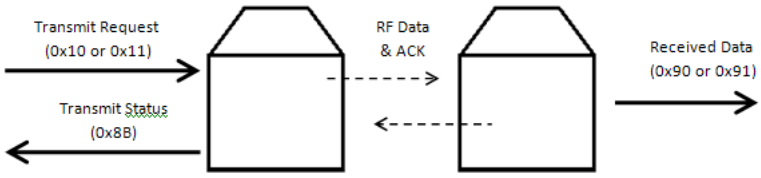
AT Commands

The following image shows the API frame exchange that takes place at the UART when sending an AT command request to read or set a module parameter. The response can be disabled by setting the frame ID to 0 in the request.



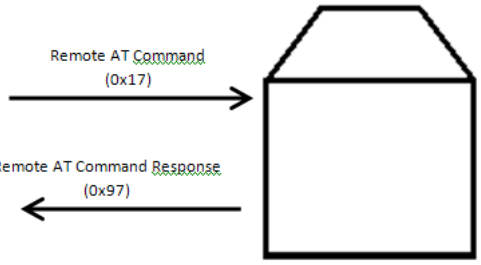
Transmitting and Receiving RF Data

The following image shows the API exchanges that take place at the UART when sending RF data to another device. The transmit status frame is always sent at the end of a data transmission unless the frame ID is set to 0 in the transmit request. If the packet cannot be delivered to the destination, the transmit status frame will indicate the cause of failure. The received data frame (0x90 or 0x91) is set by the AP command.



Remote AT Commands

The following image shows the API frame exchanges that take place at the UART when sending a remote AT command. A remote command response frame is not sent out the UART if the remote device does not receive the remote command.



Supporting the API

Applications that support the API should make provisions to deal with new API frames that may be introduced in future releases. For example, a section of code on a host microprocessor that handles received serial API frames (sent out the module's DOUT pin) might look like this:

```
void XBee_HandleRxAPIFrame(_apiFrameUnion *papiFrame){
   switch(papiFrame->api_id){
      case RX_RF_DATA_FRAME:
         //process received RF data frame
         break;

      case RX_IO_SAMPLE_FRAME:
         //process IO sample frame
         break;

      case NODE_IDENTIFICATION_FRAME:
         //process node identification frame
         break;

      default:
         //Discard any other API frame types that are not being used
         break;
   }
}
```

Frame Data

The following sections illustrate the types of frames encountered while using the API.

AT Command

Frame Type: 0x08
Used to query or set module parameters on the local device. This API command applies changes after executing the command. (Changes made to module parameters take effect once changes are applied.) The API example below illustrates an API frame when modifying the NH parameter value of the module

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x04	
	Frame-specific Data	Frame Type	3	0x08	
		Frame ID	4	0x52 (R)	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
		AT Command	5	0x4E (N)	Command Name - Two ASCII characters that identify the AT Command.
			6	0x48 (H)	
		Parameter Value (optional)			If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
	Checksum		8	0x0F	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

The above example illustrates an AT command when querying an NH value.

AT Command - Queue Parameter Value

Frame Type: 0x09
This API type allows module parameters to be queried or set. In contrast to the "AT Command" API type, new parameter values are queued and not applied until either the "AT Command" (0x08) API type or the AC (Apply Changes) command is issued. Register queries (reading parameter values) are returned immediately.

Example: Send a command to change the baud rate (BD) to 115200 baud, but don't apply changes yet. (Module will continue to operate at the previous baud rate until changes are applied.)

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x05	
	Frame-specific Data	Frame Type	3	0x09	
		Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
		AT Command	5	0x42 (B)	Command Name - Two ASCII characters that identify the AT Command.
			6	0x44 (D)	
		Parameter Value (ATBD7 = 115200 baud)		0x07	If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
	Checksum		8	0x68	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Note: In this example, the parameter could have been sent as a zero-padded 2-byte or 4-byte value.

Transmit Request

Frame Type: 0x10

A Transmit Request API frame causes the module to send data as an RF packet to the specified destination.

The 64-bit destination address should be set to 0x000000000000FFFF for a broadcast transmission (to all devices). For unicast transmissions the 64 bit address field should be set to the address of the desired destination node. The reserved field should be set to 0xFFFE.

This example shows if escaping is disabled (AP=1).

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x16	
	Frame-specific Data	Frame Type	3	0x10	
		Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
		64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF - Broadcast address
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x0A	
			11	0x01	
			LSB 12	0x27	
		Reserved	13	0xFF	Set to 0xFFFE.
			14	0xFE	
		Broadcast Radius	15	0x00	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.
		Transmit Options	16	0x00	Bitfield: bit 0: Disable ACK bit 1: Don't attempt route Discovery. All other bits must be set to 0.
		RF Data	17	0x54	Data that is sent to the destination device
			18	0x78	
			19	0x44	
			20	0x61	
			21	0x74	
			22	0x61	
			23	0x30	
			24	0x41	
	Checksum		25	0x13	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: The example above shows how to send a transmission to a module where escaping is disabled (AP=1) with destination address 0x0013A200 400A0127, payload "TxData0A". If escaping is enabled (AP=2), the frame should look like:

0x7E 0x00 0x16 0x10 0x01 0x00 0x7D 0x33 0xA2 0x00 0x40 0x0A 0x01 0x27

0xFF 0xFE 0x00 0x00 0x54 0x78 0x44 0x61 0x74 0x61 0x30 0x41 0x7D 0x33

The checksum is calculated (on all non-escaped bytes) as [0xFF - (sum of all bytes from API frame type through data payload)].

Explicit Addressing Command Frame

Frame Type: 0x11

Allows application layer fields (endpoint and cluster ID) to be specified for a data transmission. Similar to the Transmit Request, but also requires application layer addressing fields to be specified (endpoints, cluster ID, profile ID). An Explicit Addressing Request API frame causes the module to send data as an RF packet to the specified destination, using the specified source and destination endpoints, cluster ID, and profile ID.

The 64-bit destination address should be set to 0x000000000000FFFF for a broadcast transmission (to all devices). For unicast transmissions the 64 bit address field should be set to the address of the desired destination node. The reserved field should be set to 0xFFFE.

The broadcast radius can be set from 0 up to NH to 0xFF. If the broadcast radius exceeds the value of NH then the value of NH will be used as the radius. This parameter is only used for broadcast transmissions.

The maximum number of payload bytes can be read with the NP command.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x1A	
	Frame-specific Data	Frame Type	3	0x11	
		Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
		64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF - Broadcast address
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x01	
			10	0x23	
			11	0x84	
			LSB12	0x00	
		Reserved	13	0xFF	Set to 0xFFFE.
			14	0xFE	
		Source Endpoint	15	0xA0	Source endpoint for the transmission.
		Destination Endpoint	16	0xA1	Destination endpoint for the transmission.
		Cluster ID	17	0x15	Cluster ID used in the transmission
			18	0x54	
		Profile ID	19	0xC1	Profile ID used in the transmission
			20	0x05	
		Broadcast Radius	21	0x00	Sets the maximum number of hops a broadcast transmission can traverse. If set to 0, the transmission radius will be set to the network maximum hops value.
		Transmit Options	22	0x00	Bitfield: bit 0: Disable ACK bit 1: Don't attempt route Discovery. All other bits must be set to 0.
		Data Payload	23	0x54	
			24	0x78	
			25	0x44	
			26	0x61	
			27	0x74	
			28	0x61	
	Checksum		29	0xDD	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: The above example sends a data transmission to a radio with a 64 bit address of 0x0013A20001238400 using a source endpoint of 0xA0, destination endpoint 0xA1, cluster ID =0x1554, and profile ID 0xC105. Payload will be "TxData".

Remote AT Command Request

Frame Type: 0x17

Used to query or set module parameters on a remote device. For parameter changes on the remote device to take effect, changes must be applied, either by setting the apply changes options bit, or by sending an AC command to the remote.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x10	
	Frame-specific Data	Frame Type	3	0x17	
		Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
		64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF - Broadcast address
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x40	
			11	0x11	
			LSB 12	0x22	
		Reserved	13	0xFF	Set to 0xFFFE.
			14	0xFE	
		Remote Command Options	15	0x02 (apply changes)	0x02 - Apply changes on remote. (If not set, AC command must be sent before changes will take effect.) All other bits must be set to 0.
		AT Command	16	0x42 (B)	Name of the command
			17	0x48 (H)	
		Command Parameter	18	0x01	If present, indicates the requested parameter value to set the given register. If no characters present, the register is queried.
	Checksum		18	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: The above example sends a remote command to change the broadcast hops register on a remote device to 1 (broadcasts go to 1-hop neighbors only), and apply changes so the new configuration value immediately takes effect. In this example, the 64-bit address of the remote is 0x0013A200 40401122.

AT Command Response

Frame Type: 0x88
In response to an AT Command message, the module will send an AT Command Response message. Some commands will send back multiple frames (for example, the ND (Node Discover) command).

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x05	
	Frame-specific Data	Frame Type	3	0x88	
		Frame ID	4	0x01	Identifies the UART data frame being reported. Note: If Frame ID = 0 in AT Command Mode, no AT Command Response will be given.
		AT Command	5	'B' = 0x42	Command Name - Two ASCII characters that identify the AT Command.
			6	'D' = 0x44	
		Command Status	7	0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter
		Command Data			Register data in binary format. If the register was set, then this field is not returned, as in this example.
	Checksum		8	0xF0	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: Suppose the BD parameter is changed on the local device with a frame ID of 0x01. If successful (parameter was valid), the above response would be received.

Modem Status

Frame Type: (0x8A)
RF module status messages are sent from the module in response to specific conditions.

Example: The following API frame is returned when an API device powers up.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x02	
	Frame-specific Data	Frame Type	3	0x8A	
		Status	4	0x00	0x00 = Hardware reset 0x01= Watchdog timer reset 0x0B = Network Woke Up 0x0C = Network Went To Sleep
	Checksum		5	0x75	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Transmit Status

Frame Type: 0x8B

When a TX Request is completed, the module sends a TX Status message. This message will indicate if the packet was transmitted successfully or if there was a failure.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x07	
	Frame-specific Data	Frame Type	3	0x8B	
		Reserved	4	0x47	Reserved.
			5	0xFF	
			6	0xFE	
		Transmit Retry Count	7	0x00	The number of application transmission retries that took place.
		Delivery Status	8	0x00	0x00 = Success 0x01 = MAC ACK Failure 0x15 = Invalid destination endpoint 0x21 = Network ACK Failure 0x25 = Route Not Found 0x74 = Data Payload Too Large
		Discovery Status	9	0x02	0x00 = No Discovery Overhead 0x02 = Route Discovery
	Checksum		10	0x2E	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: In the above example, a unicast data transmission was sent successfully to a destination device.

Receive Packet

Frame Type: (0x90)

When the module receives an RF packet, it is sent out the UART using this message type.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x12	
	Frame-specific Data	Frame Type	3	0x90	
		64-bit Source Address	MSB 4	0x00	64-bit address of sender
			5	0x13	
			6	0xA2	
			7	0x00	
			8	0x40	
			9	0x52	
			10	0x2B	
			LSB 11	0xAA	
		Reserved	12	0xFF	Reserved
			13	0xFE	
		Receive Options	14	0x01	0x01 - Packet Acknowledged 0x02 - Packet was a broadcast packet
		Received Data	15	0x52	Received RF data
			16	0x78	
			17	0x44	
			18	0x61	
			19	0x74	
			20	0x61	
	Checksum		21	0x11	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: In the above example, a device with a 64-bit address of 0x0013A200 40522BAA sends a unicast data transmission to a remote device with payload "RxData". If AO=0 on the receiving device, it would send the above frame out its UART.

Explicit Rx Indicator

Frame Type:0x91

When the modem receives an RF packet it is sent out the UART using this message type (when AO=1).

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x18	
	Frame-specific Data	Frame Type	3	0x91	
		64-bit Source Address	MSB 4	0x00	64-bit address of sender
			5	0x13	
			6	0xA2	
			7	0x00	
			8	0x40	
			9	0x52	
			10	0x2B	
			LSB 11	0xAA	
		Reserved	12	0xFF	Reserved.
			13	0xFE	
		Source Endpoint	14	0xE0	Endpoint of the source that initiated the transmission
		Destination Endpoint	15	0xE0	Endpoint of the destination the message is addressed to.
		Cluster ID	16	0x22	Cluster ID the packet was addressed to.
			17	0x11	
		Profile ID	18	0xC1	Profile ID the packet was addressed to.
			19	0x05	
		Receive Options	20	0x02	0x01 – Packet Acknowledged 0x02 – Packet was a broadcast packet
		Received Data	21	0x52	Received RF data
			22	0x78	
			23	0x44	
			24	0x61	
			25	0x74	
			26	0x61	
	Checksum		27	0x56	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: In the above example, a device with a 64-bit address of 0x0013A200 40522BAA sends a broadcast data transmission to a remote device with payload "RxData". Suppose the transmission was sent with source and destination endpoints of 0xE0, cluster ID=0x2211, and profile ID=0xC105. If AO=1 on the receiving device, it would send the above frame out its UART.

Data Sample Rx Indicator

Frame Type:0x92

When the modem receives an RF packet it is sent out the UART using this message type (when AO=1).

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x14	
	Frame-specific Data	Frame Type	3	0x92	
		64-bit Source Address	MSB 4	0x00	64-bit address of sender
			5	0x13	
			6	0xA2	
			7	0x00	
			8	0x40	
			9	0x52	
			10	0x2B	
			LSB 11	0xAA	
		16-bit Source Network Address	MSB 12	0x7D	16-bit address of sender.
			LSB 13	0x84	
		Receive Options	14	0x01	0x01 - Packet Acknoledged 0x02 - Packet was a broadcast packet
		Number of samples	15	0x01	Number of sample sets included in the payload. (Always set to 1.)
		Digital Channel Mask*	16	0x00	Bitmask field that indicates which digital IO lines on the remote have sampling enabled (if any).
			17	0x1C	
		Analog Channel Maski**	18	0x02	Bitmask field that indicates which analog IO lines on the remote have sampling enabled (if any).
		Digital Samples (if included)	19	0x00	If the sample set includes any digital IO lines (Digital Channel Mask > 0), these two bytes contain samples for all enabled digital IO lines. DIO lines that do not have sampling enabled return 0. Bits in these 2 bytes map the same as they do in the Digital Channels Mask field.
			20	0x14	
		Analog Sample	21	0x02	If the sample set includes any analog input lines (Analog Channel Mask > 0), each enabled analog input returns a 2-byte value indicating the A/D measurement of that input. Analog samples are ordered sequentially from AD0/DIO0 to AD3/DIO3, to the supply voltage.
			22	0x25	
	Checksum		23	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Node Identification Indicator

Frame Type:0x95

This frame is received when a module transmits a node identification message to identify itself (when AO=0). The data portion of this frame is similar to a network discovery response frame (see ND command).

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x20	
	Frame-specific Data	Frame Type	3	0x95	
		64-bit Source Address	MSB 4	0x00	64-bit address of sender
			5	0x13	
			6	0xA2	
			7	0x00	
			8	0x40	
			9	0x52	
			10	0x2B	
			LSB 11	0xAA	
		16-bit Source Network Address	MSB 12	0xFF	Set to 0xFFFE.
			LSB 13	0xFE	
		Receive Options	14	0x02	0x01 - Packet Acknowledged 0x02 - Packet was a broadcast packet
		Source 16-bit address	15	0xFF	Set to 0xFFFE.
			16	0xFE	
		64-bit Network address	17	0x00	Indicates the 64-bit address of the remote module that transmitted the node identification frame.
			18	0x13	
			19	0xA2	
			20	0x00	
			21	0x40	
			22	0x52	
			23	0x2B	
			24	0xAA	
		NI String	25	0x20	Node identifier string on the remote device. The NI-String is terminated with a NULL byte (0x00).
			26	0x00	
		Parent 16-bit address	27	0xFF	Set to 0xFFFE.
			28	0xFE	
	Checksum		29	0xF4	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: If the commissioning push button is pressed on a remote router device with 64-bit address 0x0013A200 40522BAA and default NI string, the following node identification indicator would be received.

Remote Command Response

Frame Type: 0x97

If a module receives a remote command response RF data frame in response to a Remote AT Command Request, the module will send a Remote AT Command Response message out the UART. Some commands may send back multiple frames--for example, Node Discover (ND) command.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x13	
	Frame-specific Data	Frame Type	3	0x97	
		Frame ID	4	0x55	This is the same value passed in to the request.
		64-bit Source (remote) Address	MSB 5	0x00	The address of the remote radio returning this response.
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x52	
			11	0x2B	
			LSB 12	0xAA	
		Reserved	13	0xFF	Reserved
			14	0xFE	
		AT Commands	15	0x53	Name of the command
			16	0x4C	
		Command Status	17	0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter 4 = Transmit Failure
		Command Data	18	0x40	The value of the required register
			19	0x52	
			20	0x2B	
			21	0xAA	
	Checksum		22	0xF4	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: If a remote command is sent to a remote device with 64-bit address 0x0013A200 40522BAA to query the SL command, and if the frame ID=0x55, the response would look like the above example.

Appendix A: Definitions

Definitions

Terms and Definitions

PAN	Personal Area Network - A data communication network that includes a coordinator and one or more routers/end devices. Synonymous with Network ID
Network Address	The 16-bit address is used for compatibility and has an address of 0xFFFE
Route Request	Broadcast transmission sent by a coordinator or router throughout the network in attempt to establish a route to a destination node.
Route Reply	Unicast transmission sent back to the originator of the route request. It is initiated by a node when it receives a route request packet and its address matches the Destination Address in the route request packet.
Route Discovery	The process of establishing a route to a destination node when one does not exist in the Routing Table. It is based on the AODV (Ad-hoc On-demand Distance Vector routing) protocol.
DigiMesh Protocol	
Election	An election takes place to resolve which node will function as the network's sleep coordinator if multiple nodes nominate themselves at the same time,
Hopping	One direct host-to-host connection forming part of the route between hosts
Network Identifier	A user configurable string used to identify a node apart from its address
Network Address	The 64-bit address assigned to a node after it has joined to another node.
Nomination	Nomination is the process where a node becomes a sleep coordinator.
Route Request	Broadcast transmission sent by a coordinator or router throughout the network in attempt to establish a route to a destination node.
Route Reply	Unicast transmission sent back to the originator of the route request. It is initiated by a node when it receives a route request packet and its address matches the Destination Address in the route request packet.
Route Discovery	The process of establishing a route to a destination node when one does not exist in the Routing Table. It is based on the AODV (Ad-hoc On-demand Distance Vector routing) protocol.
Sleep coordinator	Node used to send sync messages in a cyclic sleeping network.

Sync message	A transmission used in a cyclic sleeping network to maintain synchronization.

Appendix B: Agency Certifications

United States (FCC)

XBee®/XBee-PRO® RF Modules comply with Part 15 of the FCC rules and regulations. Compliance with the labeling requirements, FCC notices and antenna usage guidelines is required.

To fulfill FCC Certification requirements, the OEM must comply with the following regulations:

1. The system integrator must ensure that the text on the external label provided with this device is placed on the outside of the final product [Figure A-01].
2. XBee®/XBee-PRO® RF Modules may only be used with antennas that have been tested and approved for use with this module [refer to the antenna tables in this section].

OEM Labeling Requirements

WARNING: The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product enclosure that displays the contents shown in the figure below.

Figure 6-01. Required FCC Label for OEM products containing the XBee®/XBee-PRO® RF Module

Contains FCC ID: OUR-XBEE/OUR-XBEEPRO**

The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*i.*) this device may not cause harmful interference and (*ii.*) this device must accept any interference received, including interference that may cause undesired operation.

* The FCC ID for the XBee is "OUR-XBEE". The FCC ID for the XBee-PRO is "OUR-XBEEPRO".

FCC Notices

IMPORTANT: The XBee®/XBee-PRO® RF Module has been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Modifications not expressly approved by Digi could void the user's authority to operate the equipment.

IMPORTANT: OEMs must test final product to comply with unintentional radiators (FCC section 15.107 & 15.109) before declaring compliance of their final product to Part 15 of the FCC Rules.

IMPORTANT: The RF module has been certified for remote and base radio applications. If the module will be used for portable applications, please take note of the following rules:

- For XBee modules where the antenna gain is less than 13.8 dBi, no additional SAR testing is required. The 20 cm separation distance is not required for antenna gain less than 13.8 dBi.
- For XBee modules where the antenna gain is greater than 13.8 dBi and for all XBee-PRO modules, the device must undergo SAR testing.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: Re-orient or relocate the receiving antenna,

Increase the separation between the equipment and receiver, Connect equipment and receiver to outlets on different circuits, or Consult the dealer or an experienced radio/TV technician for help.

FCC-Approved Antennas (2.4 GHz)

XBee/XBee-PRO RF Modules can be installed using antennas and cables constructed with standard connectors (Type-N, SMA, TNC, etc.) if the installation is performed professionally and according to FCC guidelines. For installations not performed by a professional, non-standard connectors (RPSMA, RPTNC, etc) must be used.

The modules are FCC-approved for fixed base station and mobile applications on channels 0x0B - 0x1A (XBee) and 0x0C - 0x17 (XBee-PRO). If the antenna is mounted at least 20cm (8 in.) from nearby persons, the application is considered a mobile application. Antennas not listed in the table must be tested to comply with FCC Section 15.203 (Unique Antenna Connectors) and Section 15.247 (Emissions).

XBee RF Modules (1 mW): XBee Modules have been tested and approved for use with the antennas listed in the first and second tables below (cable loss is required as shown).

XBee-PRO RF Modules (60 mW): XBee-PRO Modules have been tested and approved for use with the antennas listed in the first and third tables below (cable loss is required as shown).

The antennas in the tables below have been approved for use with this module. Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

Antennas approved for use with the XBee®/XBee-PRO® RF Modules (Cable-loss is not required.)

Part Number	Type (Description)	Gain	Application*	Min. Separation
A24-HASM-450	Dipole (Half-wave articulated RPSMA - 4.5")	2.1 dBi	Fixed/Mobile	20 cm
A24-HABSM	Dipole (Articulated RPSMA)	2.1 dBi	Fixed	20 cm
A24-HABUF-P5I	Dipole (Half-wave articulated bulkhead mount U.FL. w/ 5" pigtail)	2.1 dBi	Fixed	20 cm
A24-HASM-525	Dipole (Half-wave articulated RPSMA - 5.25")	2.1 dBi	Fixed/Mobile	20 cm
A24-QI	Monopole (Integrated whip)	1.5 dBi	Fixed	20 cm
A24-C1	Surface Mount	-1.5 dBi	Fixed/Mobile	20 cm
29000430	Integrated PCB Antenna	-0.5 dBi	Fixed/Mobile	20 cm

Antennas approved for use with the XBee RF Modules (Cable-loss is required)

Part Number	Type (Description)	Gain	Application*	Min. Separation	Required Cable-loss
Yagi Class Antennas					
A24-Y4NF	Yagi (4-element)	6.0 dBi	Fixed	2 m	
A24-Y6NF	Yagi (6-element)	8.8 dBi	Fixed	2 m	1.7 dB
A24-Y7NF	Yagi (7-element)	9.0 dBi	Fixed	2 m	1.9 dB
A24-Y9NF	Yagi (9-element)	10.0 dBi	Fixed	2 m	2.9 dB
A24-Y10NF	Yagi (10-element)	11.0 dBi	Fixed	2 m	3.9 dB
A24-Y12NF	Yagi (12-element)	12.0 dBi	Fixed	2 m	4.9 dB
A24-Y13NF	Yagi (13-element)	12.0 dBi	Fixed	2 m	4.9 dB
A24-Y15NF	Yagi (15-element)	12.5 dBi	Fixed	2 m	5.4 dB
A24-Y16NF	Yagi (16-element)	13.5 dBi	Fixed	2 m	6.4 dB
A24-Y16RM	Yagi (16-element, RPSMA connector)	13.5 dBi	Fixed	2 m	6.4 dB
A24-Y18NF	Yagi (18-element)	15.0 dBi	Fixed	2 m	7.9 dB
Omni-Directional Class Antennas					
A24-F2NF	Omni-directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	
A24-F3NF	Omni-directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	
A24-F5NF	Omni-directional (Fiberglass base station)	5.0 dBi	Fixed/Mobile	20 cm	
A24-F8NF	Omni-directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	
A24-F9NF	Omni-directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	0.2 dB
A24-F10NF	Omni-directional (Fiberglass base station)	10.0 dBi	Fixed	2 m	0.7 dB
A24-F12NF	Omni-directional (Fiberglass base station)	12.0 dBi	Fixed	2 m	2.7 dB
A24-F15NF	Omni-directional (Fiberglass base station)	15.0 dBi	Fixed	2 m	5.7 dB
A24-W7NF	Omni-directional (Base station)	7.2 dBi	Fixed	2 m	
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	
Panel Class Antennas					
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	1.5 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	2 m	1.5 dB

Part Number	Type (Description)	Gain	Application*	Min. Separation	Required Cable-loss
A24-P13NF	Flat Panel	13.0 dBi	Fixed	2 m	6 dB
A24-P14NF	Flat Panel	14.0 dBi	Fixed	2 m	7 dB
A24-P15NF	Flat Panel	15.0 dBi	Fixed	2 m	8 dB
A24-P16NF	Flat Panel	16.0 dBi	Fixed	2 m	9 dB

Antennas approved for use with the XBee®/XBee-PRO® RF Modules (Cable-loss is required)

Part Number	Type (Description)	Gain	Application*	Min. Separation	Required Cable-loss
Yagi Class Antennas					
A24-Y4NF	Yagi (4-element)	6.0 dBi	Fixed	2 m	8.1 dB
A24-Y6NF	Yagi (6-element)	8.8 dBi	Fixed	2 m	10.9 dB
A24-Y7NF	Yagi (7-element)	9.0 dBi	Fixed	2 m	11.1 dB
A24-Y9NF	Yagi (9-element)	10.0 dBi	Fixed	2 m	12.1 dB
A24-Y10NF	Yagi (10-element)	11.0 dBi	Fixed	2 m	13.1 dB
A24-Y12NF	Yagi (12-element)	12.0 dBi	Fixed	2 m	14.1 dB
A24-Y13NF	Yagi (13-element)	12.0 dBi	Fixed	2 m	14.1 dB
A24-Y15NF	Yagi (15-element)	12.5 dBi	Fixed	2 m	14.6 dB
A24-Y16NF	Yagi (16-element)	13.5 dBi	Fixed	2 m	15.6 dB
A24-Y16RM	Yagi (16-element, RPSMA connector)	13.5 dBi	Fixed	2 m	15.6 dB
A24-Y18NF	Yagi (18-element)	15.0 dBi	Fixed	2 m	17.1 dB
Omni-Directional Class Antennas					
A24-F2NF	Omni-directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	4.2 dB
A24-F3NF	Omni-directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	5.1 dB
A24-F5NF	Omni-directional (Fiberglass base station)	5.0 dBi	Fixed/Mobile	20 cm	7.1 dB
A24-F8NF	Omni-directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	10.1 dB
A24-F9NF	Omni-directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	11.6 dB
A24-F10NF	Omni-directional (Fiberglass base station)	10.0 dBi	Fixed	2 m	12.1 dB
A24-F12NF	Omni-directional (Fiberglass base station)	12.0 dBi	Fixed	2 m	14.1 dB
A24-F15NF	Omni-directional (Fiberglass base station)	15.0 dBi	Fixed	2 m	17.1 dB
A24-W7NF	Omni-directional (Base station)	7.2 dBi	Fixed	2 m	9.3 dB
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	9.3 dB
Flat Panel Class Antennas					
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	8.6 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	2 m	8.6 dB
A24-P13NF	Flat Panel	13.0 dBi	Fixed	2 m	13.1 dB
A24-P14NF	Flat Panel	14.0 dBi	Fixed	2 m	14.1 dB
A24-P15NF	Flat Panel	15.0 dBi	Fixed	2 m	15.1 dB
A24-P16NF	Flat Panel	16.0 dBi	Fixed	2 m	16.1 dB
A24-P19NF	Flat Panel	19.0 dBi	Fixed	2 m	19.1 dB
Waveguide Class Antennas					
RSM	Waveguide	7.1 dBi	Fixed	2 m	1.5 dB
Helical Class Antennas					
A24-H3UF	Helical	3.0 dBi	Fixed/Mobile	20 cm	0 dB

*** If using the RF module in a portable application** (For example - If the module is used in a handheld device and the antenna is less than 20cm from the human body when the device is operation): The integrator may be responsible for passing additional SAR (Specific Absorption Rate) testing based on FCC rules 2.1091 and FCC Guidelines for Human Exposure to Radio Frequency Electromagnetic Fields, OET Bulletin and Supplement C. See the note under FCC Notices for more information. The testing results will be submitted to the FCC for approval prior to selling the integrated unit. The required SAR testing measures emissions from the module and how they affect the person.

RF Exposure

WARNING: To satisfy FCC RF exposure requirements for mobile transmitting devices, a separation distance of 20 cm or more should be maintained between the antenna of this device and persons during device operation. To ensure compliance, operations at closer than this distance is not recommended. The antenna used for this transmitter must not be co-located in conjunction with any other antenna or transmitter.

The preceding statement must be included as a CAUTION statement in OEM product manuals in order to alert users of FCC RF Exposure compliance.

Europe (ETSI)

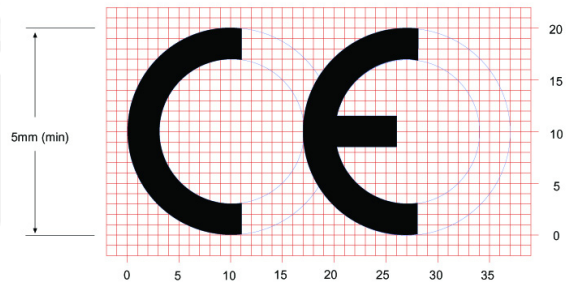
The XBee RF Modules have been certified for use in several European countries. For a complete list, refer to www.digi.com

If the XBee RF Modules are incorporated into a product, the manufacturer must ensure compliance of the final product to the European harmonized EMC and low-voltage/safety standards. A Declaration of Conformity must be issued for each of these standards and kept on file as described in Annex II of the R&TTE Directive.

Furthermore, the manufacturer must maintain a copy of the XBee user manual documentation and ensure the final product does not exceed the specified power ratings, antenna specifications, and/ or installation requirements as specified in the user manual. If any of these specifications are exceeded in the final product, a submission must be made to a notified body for compliance testing to all required standards.

OEM Labeling Requirements

The 'CE' marking must be affixed to a visible location on the OEM product.



CE Labeling Requirements

The CE mark shall consist of the initials "CE" taking the following form:

- If the CE marking is reduced or enlarged, the proportions given in the above graduated drawing must be respected.
- The CE marking must have a height of at least 5mm except where this is not possible on account of the nature of the apparatus.
- The CE marking must be affixed visibly, legibly, and indelibly.

Restrictions

Power Output: When operating in Europe, XBee-PRO 802.15.4 modules must operate at or below a transmit power output level of 10dBm. Customers have two choices for transmitting at or below 10dBm:

a. Order the standard XBee-PRO module and change the PL command to 0 (10dBm)

b. Order the International variant of the XBee-PRO module, which has a maximum transmit output power of 10dBm (@ PL=4).

Additionally, European regulations stipulate an EIRP power maximum of 12.86 dBm (19 mW) for the XBee-PRO and 12.11 dBm for the XBee when integrating antennas.

France: Outdoor use limited to 10 mW EIRP within the band 2454-2483.5 MHz.

Norway: Norway prohibits operation near Ny-Alesund in Svalbard. More information can be found at the Norway Posts and Telecommunications site (www.npt.no).

Declarations of Conformity

Digi has issued Declarations of Conformity for the XBee RF Modules concerning emissions, EMC and safety. Files can be obtained by contacting Digi Support.

Important Note:

Digi does not list the entire set of standards that must be met for each country. Digi customers assume full responsibility for learning and meeting the required guidelines for each country in their distribution market. For more information relating to European compliance of an OEM product incorporating the XBee RF Module, contact Digi, or refer to the following web sites:

CEPT ERC 70-03E - Technical Requirements, European restrictions and general requirements: Available at www.ero.dk/.

R&TTE Directive - Equipment requirements, placement on market: Available at www.ero.dk/.

Approved Antennas

When integrating high-gain antennas, European regulations stipulate EIRP power maximums. Use the following guidelines to determine which antennas to design into an application.

XBee-PRO RF Module

The following antenna types have been tested and approved for use with the XBee Module:

Antenna Type: Yagi
RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss (EIRP Maximum of 14 dBm). Any Yagi type antenna with 14 dBi gain or less can be used with no cable-loss.

Antenna Type: Omni-directional
RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss (EIRP Maximum of 14 dBm). Any Omni-directional type antenna with 14 dBi gain or less can be used with no cable-loss.

Antenna Type: Flat Panel
RF module was tested and approved with 19 dBi antenna gain with 4.8 dB cable-loss (EIRP Maximum of 14.2 dBm). Any Flat Panel type antenna with 14.2 dBi gain or less can be used with no cable-loss.

XBee-PRO RF Module (@ 10 dBm Transmit Power, PL parameter value must equal 0, or use International variant)

The following antennas have been tested and approved for use with the embedded XBee-PRO RF Module:

- Dipole (2.1 dBi, Omni-directional, Articulated RPSMA, Digi part number A24-HABSM)
- Chip Antenna (-1.5 dBi)
- Attached Monopole Whip (1.5 dBi)
- Integrated PCB Antenna (-0.5 dBi)

The RF modem encasement was designed to accommodate the RPSMA antenna option.

Canada (IC)

Labeling Requirements

Labeling requirements for Industry Canada are similar to those of the FCC. A clearly visible label on the outside of the final product enclosure must display the following text:

Contains Model XBee Radio, IC: 4214A-XBEE
Contains Model XBee-PRO Radio, IC: 4214A-XBEEPRO

The integrator is responsible for its product to comply with IC ICES-003 & FCC Part 15, Sub. B - Unintentional Radiators. ICES-003 is the same as FCC Part 15 Sub. B and Industry Canada accepts FCC test report or CISPR 22 test report for compliance with ICES-003.

Japan

In order to use the XBee-PRO in Japan, you must order the International version. The International XBee-PRO RF Modules are limited to a transmit power output of 10 dBm.

Labeling Requirements

A clearly visible label on the outside of the final product enclosure must display the following text:

R201WW07215214 (XBee)

R201WW08215111 (XBee-PRO)

Appendix C. Additional Information

1-Year Warranty

XBee®/XBee-PRO® RF Modules from Digi Intenational, Inc. (the "Product") are warranted against defects in materials and workmanship under normal use, for a period of 1-year from the date of purchase. In the event of a product failure due to materials or workmanship, Digi will repair or replace the defective product. For warranty service, return the defective product to Digi, shipping prepaid, for prompt repair or replacement.

The foregoing sets forth the full extent of Digi's warranties regarding the Product. Repair or replacement at Digi's option is the exclusive remedy. THIS WARRANTY IS GIVEN IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, AND DIGI SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL DIGI, ITS SUPPLIERS OR LICENSORS BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, FOR ANY LOSS OF USE, LOSS OF TIME, INCONVENIENCE, COMMERCIAL LOSS, LOST PROFITS OR SAVINGS, OR OTHER INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES. THEREFORE, THE FOREGOING EXCLUSIONS MAY NOT APPLY IN ALL CASES. This warranty provides specific legal rights. Other rights which vary from state to state may also apply.

Digital relative humidity & temperature sensor RHT03

1. Feature & Application:

*High precision
*Capacitive type
*Full range temperature compensated
*Relative humidity and temperature measurement
*Calibrated digital signal
*Outstanding long-term stability
*Extra components not needed
*Long transmission distance, up to 100 meters
*Low power consumption
*4 pins packaged and fully interchangeable

2. Description:

RHT03 output calibrated digital signal. It applys exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(100m) enable RHT03 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

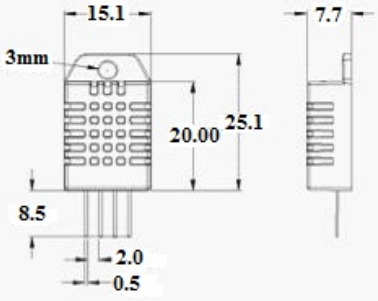
Model	RHT03	
Power supply	3.3-6V DC	
Output signal	digital signal via MaxDetect 1-wire bus	
Sensing element	Polymer humidity capacitor	
Operating range	humidity 0-100%RH;	temperature -40~80Celsius
Accuracy	**humidity +-2%RH**(Max +-5%RH);	temperature +-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-0.2Celsius

MaxDetect Technology Co., Ltd. **http://www.humiditycn.com**

Thomas Liu (Sales Manager)
Tel: [illegible] , sales@humiditycn.com

Humidity hysteresis	+-0.3%RH
Long-term Stability	+-0.5%RH/year
Interchangeability	fully interchangeable

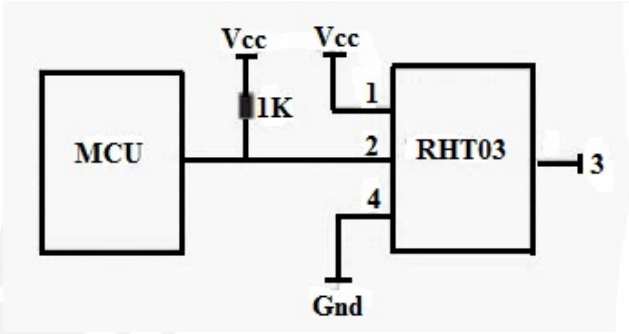
4. Dimensions: (unit----mm)



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD—power supply
2	DATA--signal
3	NULL
4	GND

5. Electrical connection diagram:



6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

MaxDetect 1-wire bus is used for communication between MCU and RHT03. (MaxDetect 1-wire bus is specially designed by MaxDetect Technology Co., Ltd. , it's different from Maxim/Dallas 1-wire bus, so it's incompatible with Dallas 1-wire bus.)

Illustration of MaxDetect 1-wire bus:

Data is comprised of integral and decimal part, the following is the formula for data.

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum

MaxDetect Technology Co., Ltd. **http://www.humiditycn.com**

Thomas Liu (Sales Manager)

Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com

If the data transmission is right, check-sum should be:
Check sum=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data

Example: MCU has received 40 bits data from RHT03 as

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

16 bits RH data 16 bits T data check sum

Check sum=0000 0010+1000 1100+0000 0001+0101 1111=1110 1110

RH= (0000 0010 1000 1100)/10=65.2%RH

T=(0000 0001 0101 1111)/10=35.1□

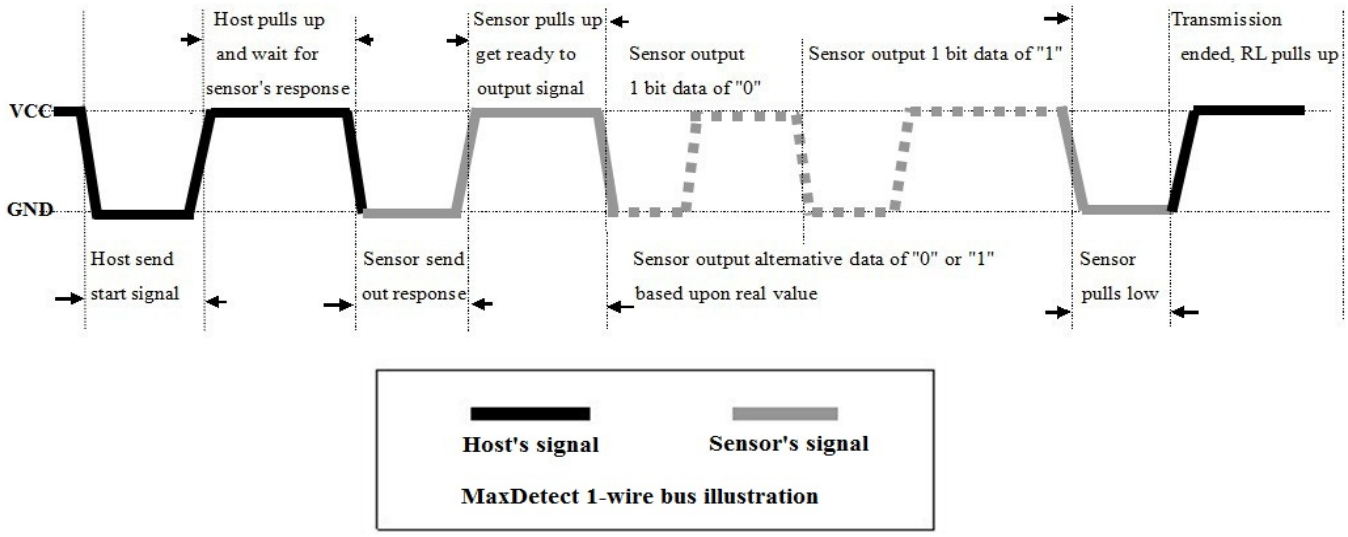
When highest bit of temperature is 1, it means the temperature is below 0 degree Celsius.

Example: 1000 0000 0110 0101, T= minus 10.1□

16 bits T data

When MCU send start signal, RHT03 change from standby-status to running-status. When MCU finishs sending the start signal, RHT03 will send response signal of 40-bit data that reflect the relative humidity and temperature to MCU. Without start signal from MCU, RHT03 will not give response signal to MCU. One start signal for one response data from RHT03 that reflect the relative humidity and temperature. RHT03 will change to standby status when data collecting finished if it don't receive start signal from MCU again.

See below figure for overall communication process, **the interval of whole process must beyond 2 seconds.**



MaxDetect Technology Co., Ltd. **http://www.humiditycn.com**

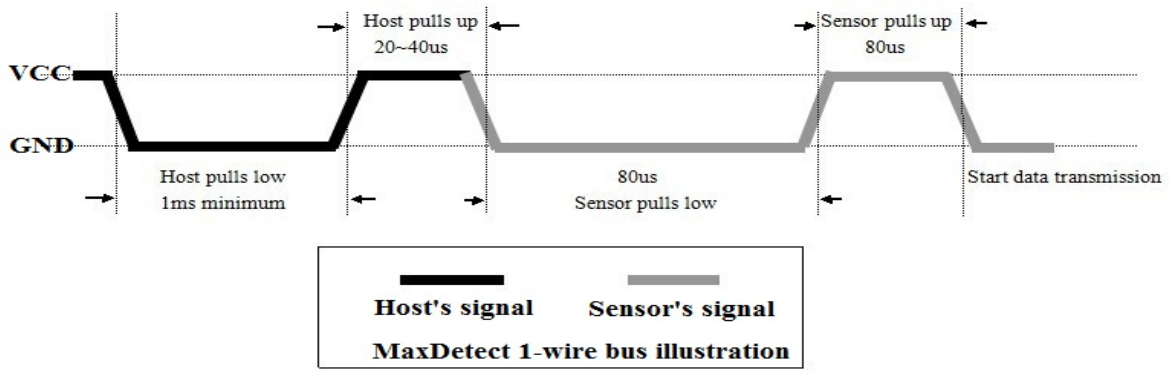
Thomas Liu (Sales Manager)
Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com

---

1) Step 1: MCU send out start signal to RHT03 and RHT03 send response signal to MCU

Data-bus's free status is high voltage level. When communication between MCU and RHT03 begins, MCU will pull low data-bus and this process must beyond at least 1~10ms to ensure RHT03 could detect MCU's signal, then MCU will pulls up and wait 20-40us for RHT03's response.

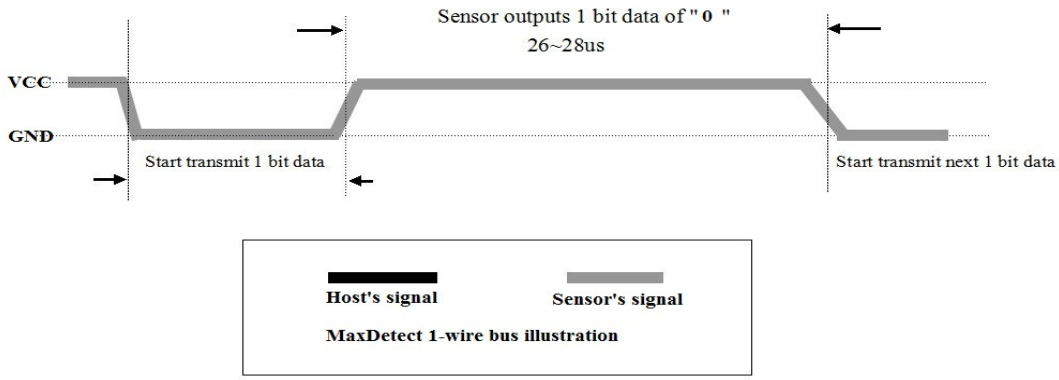
When RHT03 detect the start signal, RHT03 will pull low the bus 80us as response signal, then RHT03 pulls up 80us for preparation to send data. See below figure:



---

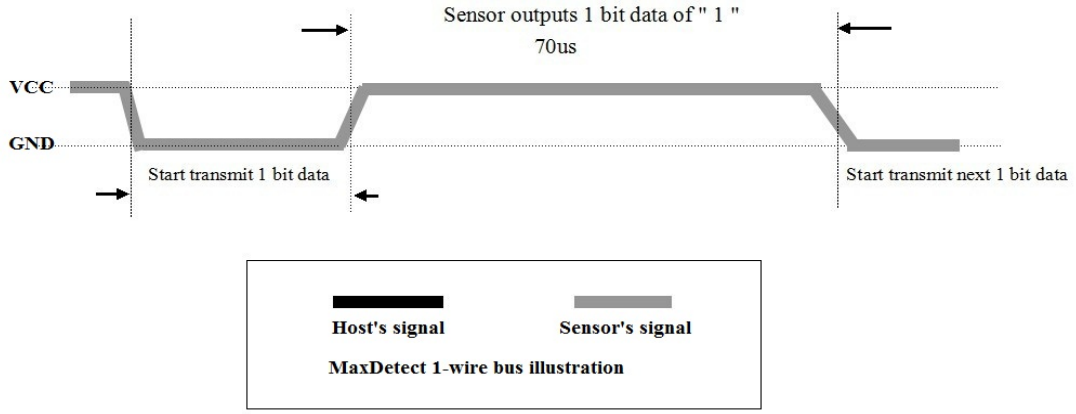
2). Step 2: RHT03 send data to MCU

When RHT03 is sending data to MCU, every bit's transmission begin with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0". See below figures:



MaxDetect Technology Co., Ltd. **http://www.humiditycn.com**

Thomas Liu (Sales Manager)
Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com



Attention:

If signal from RHT03 is always high-voltage-level, it means RHT03 is not working properly, please check the electrical connection status.

7. Electrical Characteristics:

Items	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

8. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The RHT03 sensor can recover after working in abnormal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere RHT03's sensitive-elements and debase RHT03's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the RHT03 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the RHT03 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

MaxDetect Technology Co., Ltd. **http://www.humiditycn.com**

Thomas Liu (Sales Manager)

Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

RHT03 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase RHT03's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius.

* Avoid using the sensor under dew condition.

* Don't use this product in safety or emergency stop devices or any other occasion that failure of RHT03 may cause personal injury.

Thomas Liu (Sales Manager)
Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com

X-CTU
Configuration & Test Utility Software

User's Guide

Contents

Technical Support:

Online support: http://www.digi.com/support/eservice/login.jsp

Phone: (801) 765-9885

90001003_A

2008.08.20

Introduction

This User's Guide is intended to discuss the functions of Digi's X-CTU software utility. Each function will be discussed in detail allowing a better understanding of the program and how it can be used.

X-CTU is a Windows-based application provided by Digi. This program was designed to interact with the firmware files found on Digi's RF products and to provide a simple-to-use graphical user interface to them.

X-CTU is designed to function with all Windows-based computers running Microsoft Windows 98 SE and above. X-CTU can either be downloaded from Digi's Web site or an installation CD. When properly installed it can be launched by clicking on the icon on the PC desktop (see Figure 1) or selecting from the Start menu (see Figure 2).

Figure 1

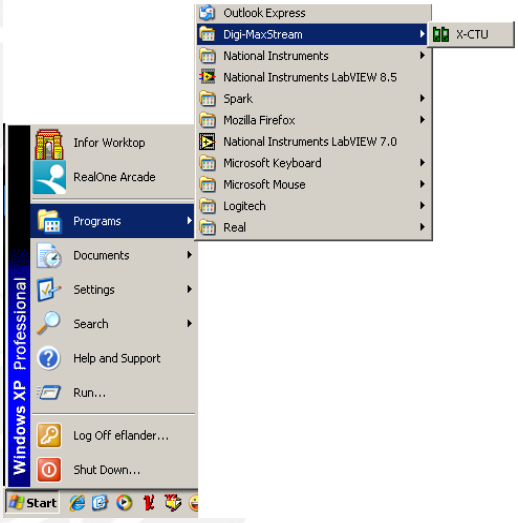


Figure 2

When launched, you will see four tabs across the top of the program (see Figure 3). Each of these tabs has a different function. The four tabs are:

PC Settings: Allows a customer to select the desired COM port and configure that port to fit the radios settings.

Range Test: Allows a customer to perform a range test between two radios.

Terminal: Allows access to the computers COM port with a terminal emulation program. This tab also allows the ability to access the radios' firmware using AT commands (for a complete listing of the radios' AT commands, please see the product manuals available online).

Modem Configuration: Allows the ability to program the radios' firmware settings via a graphical user interface. This tab also allows customers the ability to change firmware versions.

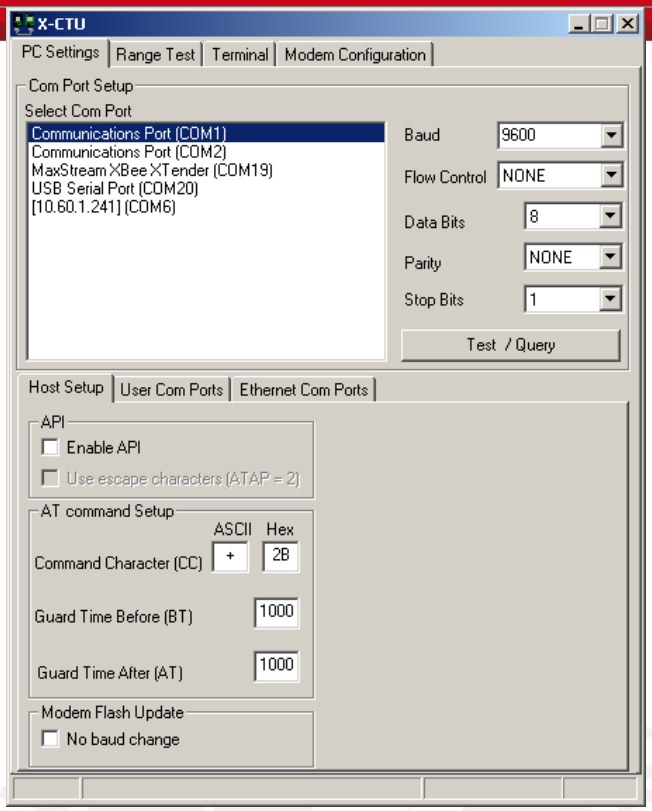


Figure 3

PC Settings Tab

When the program is launched, the default tab selected is the "PC Settings" tab. The PC Settings tab is broken down into three basic areas: The COM port setup, the Host Setup, and the User Com ports.

COM port setup:

The PC settings tab allows the user to select a COM port and configure the selected COM port settings when accessing the port. Some of these settings include:

Baud Rate:	Both standard and non-standard
Flow Control:	Hardware, Software (Xon/Xoff), None
Data bits:	4, 5, 6, 7, and 8 data bits
Parity:	None, Odd, Even, Mark and Space
Stop bit:	1, 1.5, and 2

To change any of the above settings, select the pull down menu on the left of the value and select the desired setting. To enter a non-standard baud rate, type the baud rate into the baud rate box to the left.

The **Test / Query** button is used to test the selected COM port and PC settings. If the settings and COM port are correct, you will receive a response similar to the one depicted in Figure 4 below.

Figure 4

Host Setup:

The Host Setup tab allows the user to configure how the X-CTU program is to interface with a radio's firmware. This includes determining whether API or AT command mode will be used to access the module's firmware as well as the proper command mode character and sequence.

By default, the Host Settings are as follows:

API mode:	not enabled (Not checked)
Command mode Character:	+ (ACSII) 2B (Hex).
Before Guard Time:	1000 (1 Sec)
After Guard Time:	1000 (1 Sec)

This is the default value of our radios. If this is not the value of the AT, BT, or GT commands of the connected radio, enter the respective value here.

User COM ports:

The user COM port option allows the user to "Add" or "Delete" a user-created COM port. This is only for temporary use. Once the program has closed, the user-created COM port will disappear and is no longer accessible to the program.

Range Test Tab

The range test tab is designed to verify the range of the radio link by sending a user-specified data packet and verifying the response packet is the same, within the time specified. For performing a standard range test, please follow the steps found in most Quick Start or Getting Started Guides that ship with the product.

Packet Data and Size

By default, the size of the data packet sent is 32 bytes. This data packet specified can be adjusted in either size or the text sent.

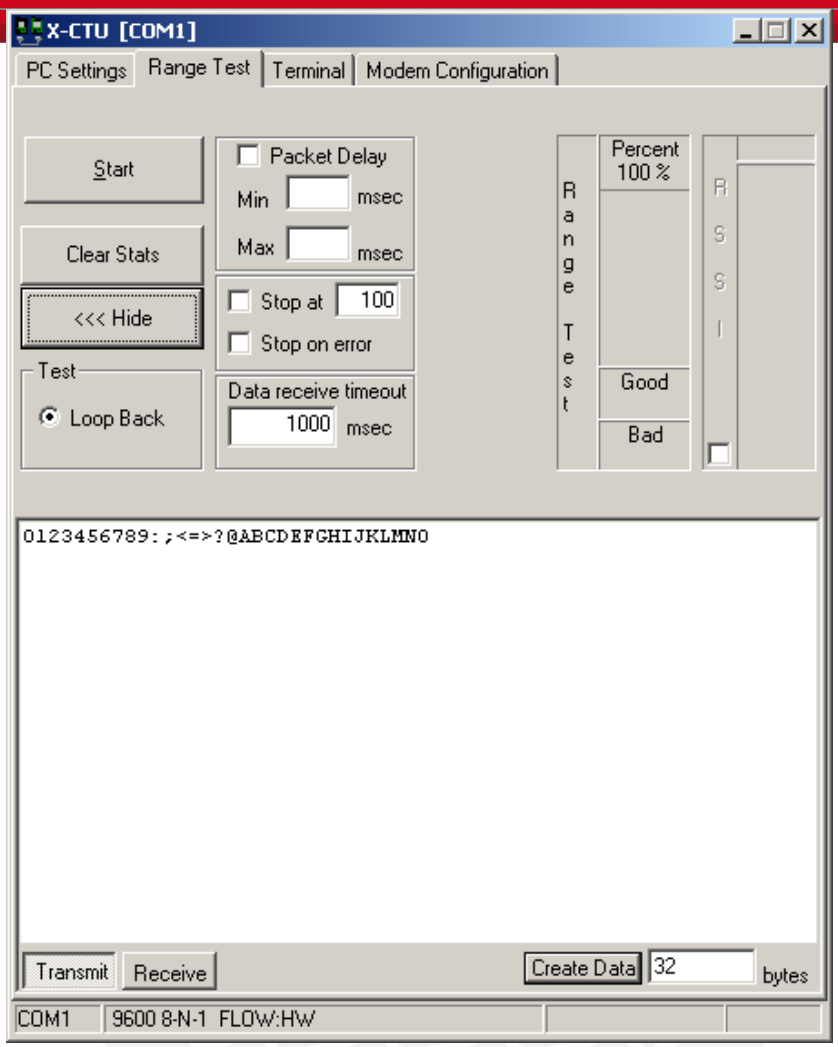


Figure 5

To modify the size of the packet sent, change the value next to the "Create Data" box and click on the "Create Data" button (see Figure 5). If you want to change the data sent, delete the text in the transmit window and place in your desired text.

By modifying the text, data packet size, packet delay and the data receive timeout; the user is able to simulate a wide range of scenarios.

RSSI:

The RSSI option of the X-CTU allows the user to see the RSSI (Received Signal Strength Indicator) of a received packet when performing a range test.

API Function:

The X-CTU also allows the user to test the API function of a radio during a range test.

To perform a range test with the API function of the radio, follow the steps outlined below:

1: Configure the Base with API enabled and a unique 16 bit or 64 bit source address.
2: Configure the remote radio with a unique source address and set the Destination address to equal the Base radio's source address.
3: Enable the API option of the X-CTU on the PC Settings tab and connect the base radio to the PC (See Figure 3).
4: Connect the red loopback adapter to the remote radio and place them a distance apart.
5: Enter either the 16 bit or 64 bit destination address of the remote radio into the Destination Address box on the Range Test tab (See figure 6).
6: Create a data packet of your choosing by typing in the data in the Transmit box
7: To start a Range test, click on Start.

You will notice the TX failures, Purge, CCA, and ACK messages will increment accordingly while the range test is performed.

To stop a range test, click on the Stop button.

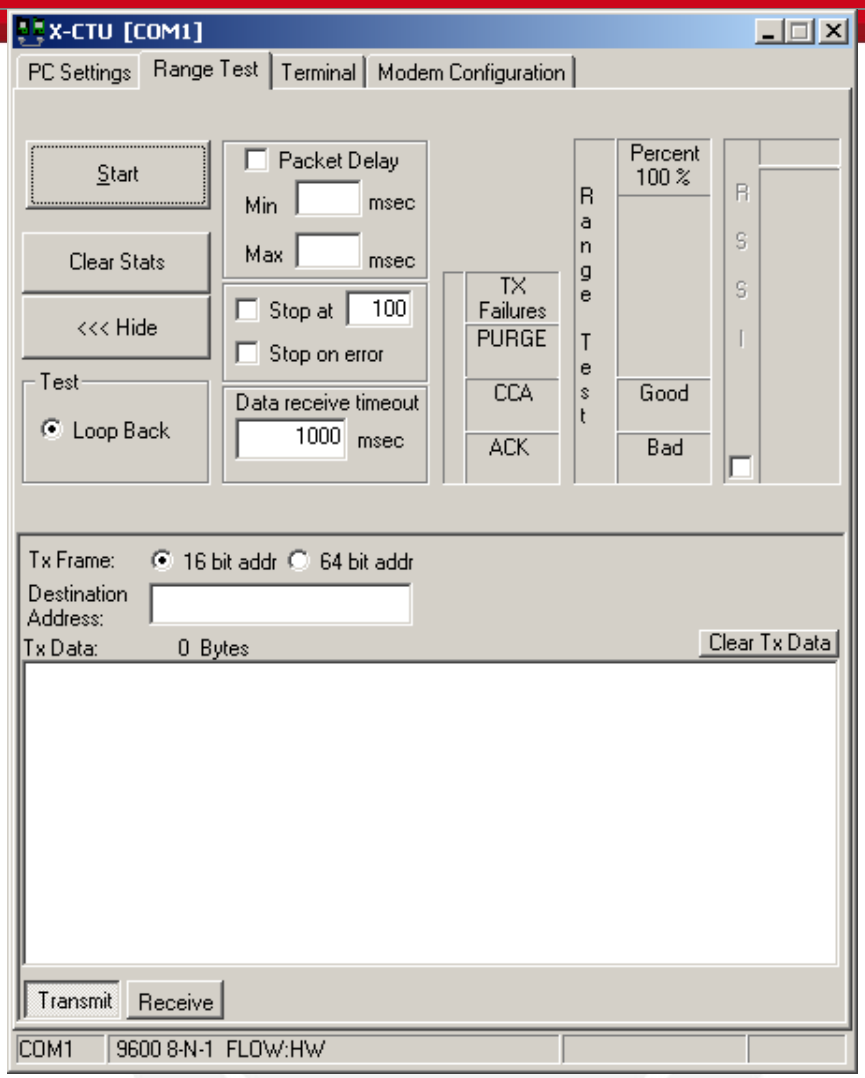


Figure 6

The Terminal Tab

The Terminal tab has three basic functions:

Terminal emulator

Ability to send and receive predefined data pacts (Assemble packet)

Ability to send and receive data in Hex and ASCII formats (Show/Hide hex)

The main terminal window

The main white portion of this tab is where most of the communications information will occur while using X-CTU as a terminal emulator. The text in blue is what has been typed in and directed out to the radio's serial port while the red text is the incoming data from the radio's serial port (see Figure 7).

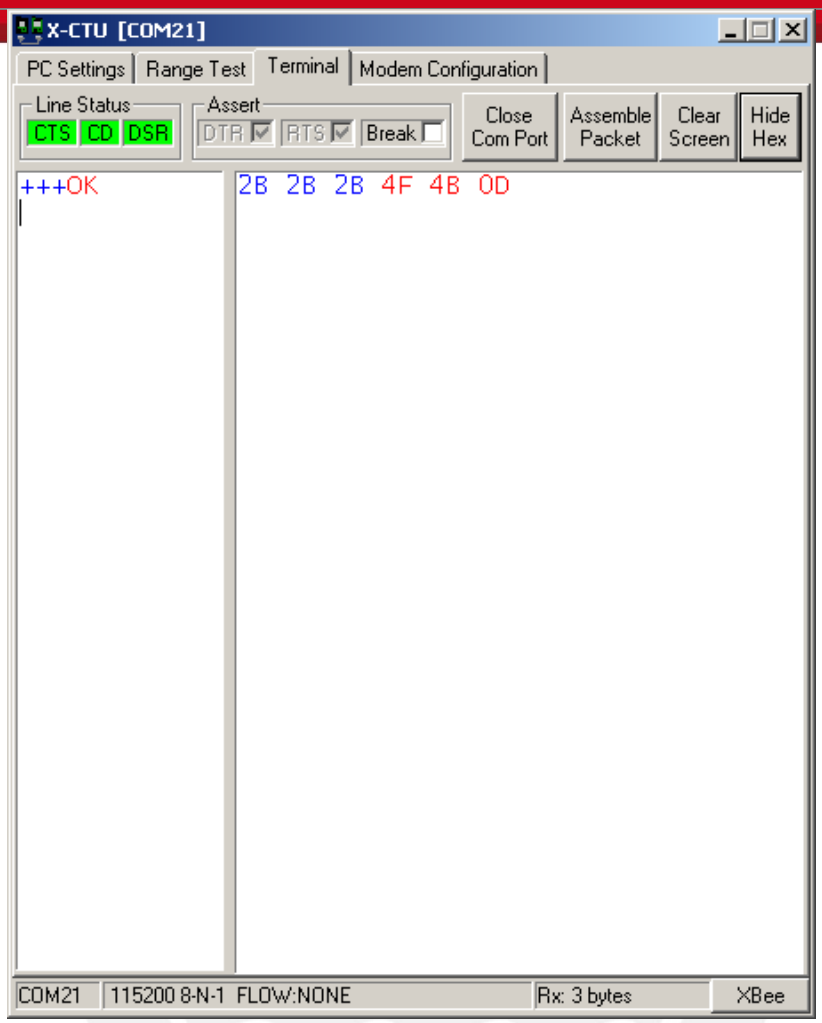


Figure 7

Assemble Packet

The Assemble Packet option on the Terminal tab is designed to allow the user to assemble a data packet in either ASCII or Hex characters. This is accomplished by selecting the Assemble packet window and choosing either ASCII (default) or Hex. Once selected, the data packet is assembled by typing in the desired characters as depicted in Figure 8.

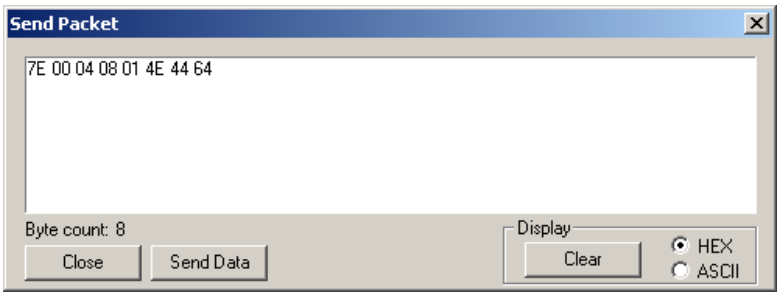


Figure 8

The **Line Status** indicators depicted in Figure 5 shows the status of the RS-232 hardware flow control lines. Green indicates the line is asserted while black indicates de-asserted.

The **Break** option is for engaging the serial line break. This can be accomplished by checking or asserting the Break option. Asserting the Break will place the DI line high and prevent data from being sent to the radio.

Modem Configuration tab

The Modem configuration tab has four basic functions:

1: Provide a Graphical User Interface with a radio's firmware
2: Read and Write firmware to the radio's microcontroller
3: Download updated firmware files from either the web or from a compressed file
4: Saving or loading a modem profile

Reading a radio's firmware

To read a radio's firmware, follow the steps outlined below:

1: Connect the radio module to the interface board and connect this assembly or a packaged radio (PKG) to the PC's corresponding port (IE: USB, RS232, Ethernet etc.).
2: Set the PC Settings tab (see Figure 3) to the radio's default settings.
3: On the Modem Configuration tab, select "Read" from the Modem Parameters and Firmware section (see Figure 9).

Making changes to a radio's firmware

Once the radio's firmware has been read, the configuration settings are displayed in three colors (see Figure 10):

Black - not settable or read-only
Green - Default value
Blue - User-specified

To modify any of the user-settable parameters, click on the associated command and type in the new value for that parameter. For ease of understanding a specific command, once the command is selected, a quick description along with its limits is provided at the bottom of the screen. Once all of the new values have been entered, the new values are ready to be saved to the radio's non-volatile memory.

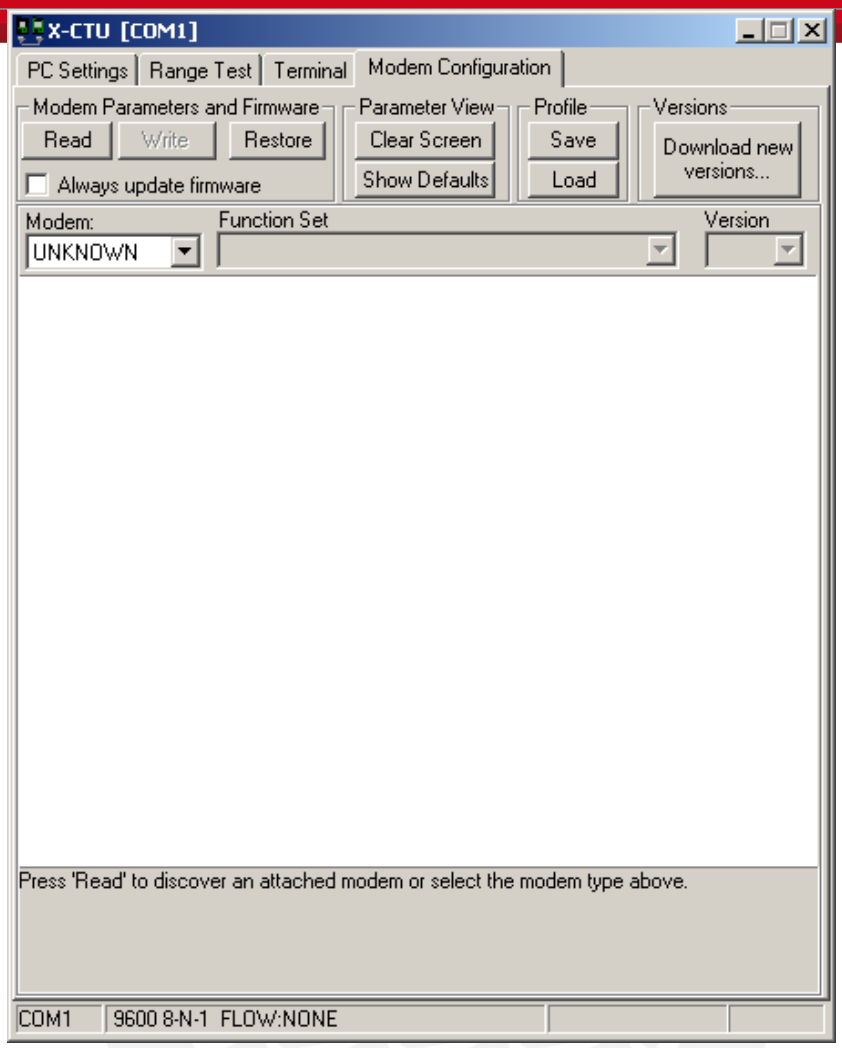


Figure 9

Writing firmware to the Radio

To write the parameter changes to the radio's non-volatile memory, click on the Write button located in the Modem Parameters and Firmware section (see Figure 10)

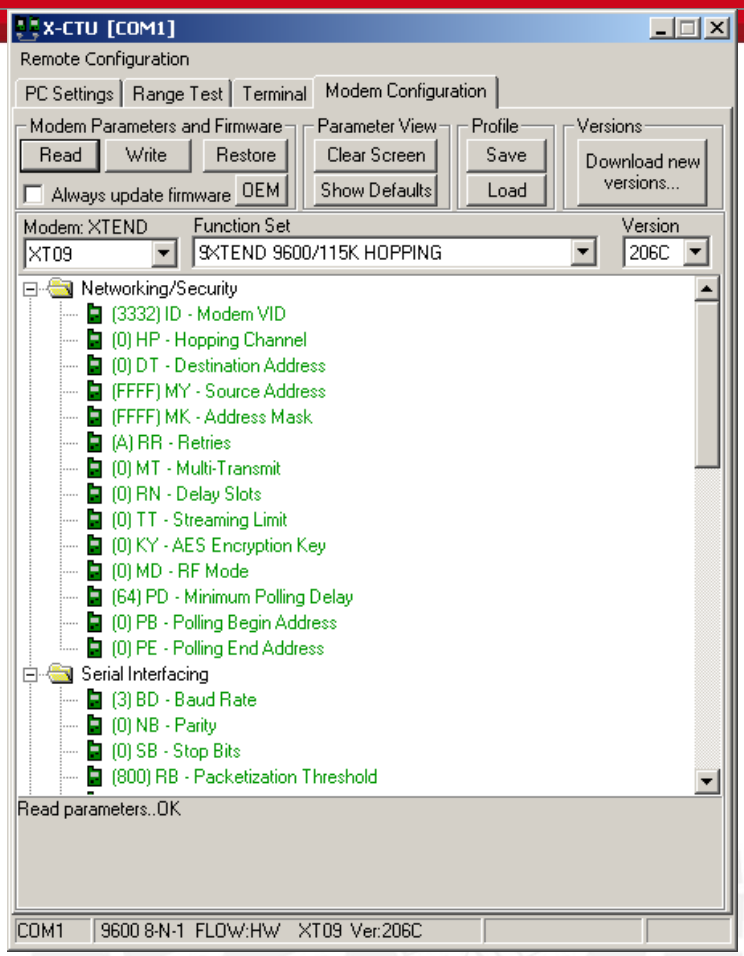


Figure 10

Downloading Updated Firmware Files

Another function of the Modem Configuration tab is allowing the user to download updated firmware files from either the web or install them from a disk or CD. This is accomplished by following the steps below:

1: Click on the Download New Versions... option under the Version section
2a: Click on Web for downloading new firmware files from the web
2b: Click on the File when installing compressed firmware files from a CD or saved file (see Figures 11 and 12)
    2bi: Browse to the location the file is saved at and click on Open (see Figure 13)
3: Click on OK and Done when prompted

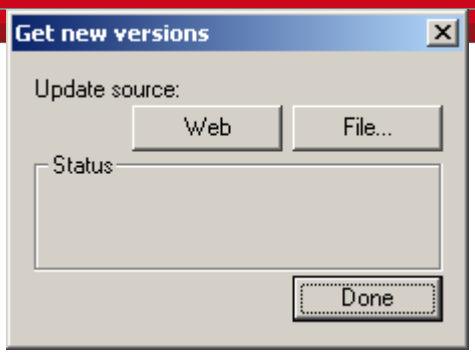


Figure 11

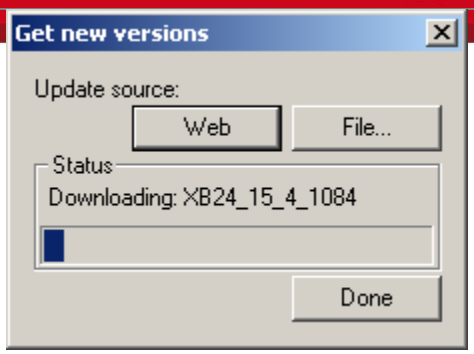


Figure 12

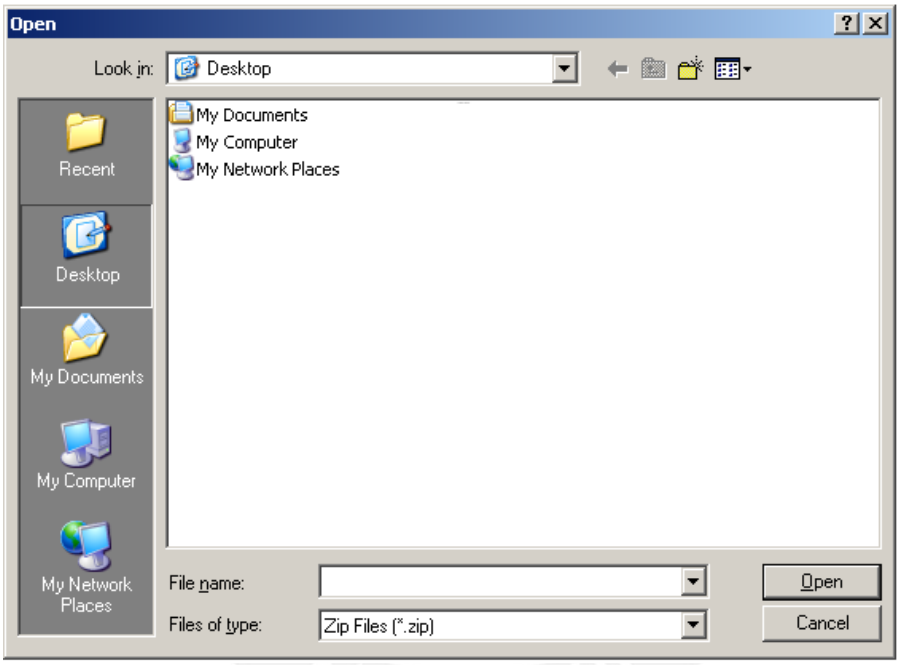


Figure 13

Modem Profiles

The X-CTU has the ability to save and write saved modem profiles or configuration to the radio. This function is useful in a production environment when the same parameters need to be set on multiple radios.

How to save a profile:

1: Set the desired settings within the radio's firmware as described in the Making changes to the radios firmware section
2: Click Save in the Profile section
3: Type in the desired name of this profile in the File Name box (see Figure 14)
4: Browse to the location where you wish to save your profile
5: Click Save

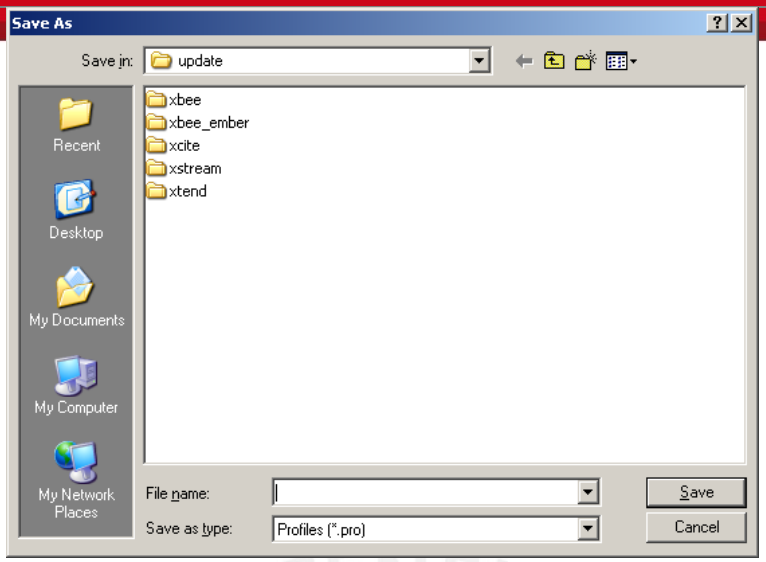


Figure 14

How to load a saved profile:

1: Click on Load from the profile section
2: Browse to the location of the file and click on the desired file (see Figure 15)
3: Click Open

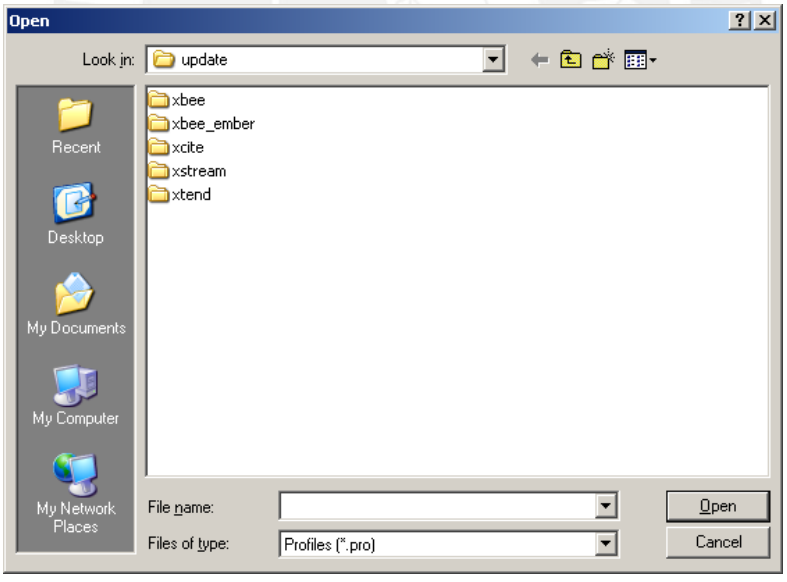


Figure 15

To save the loaded profile to the radio once you have loaded the file, follow the steps outlined in the Writing firmware to the radio section above.

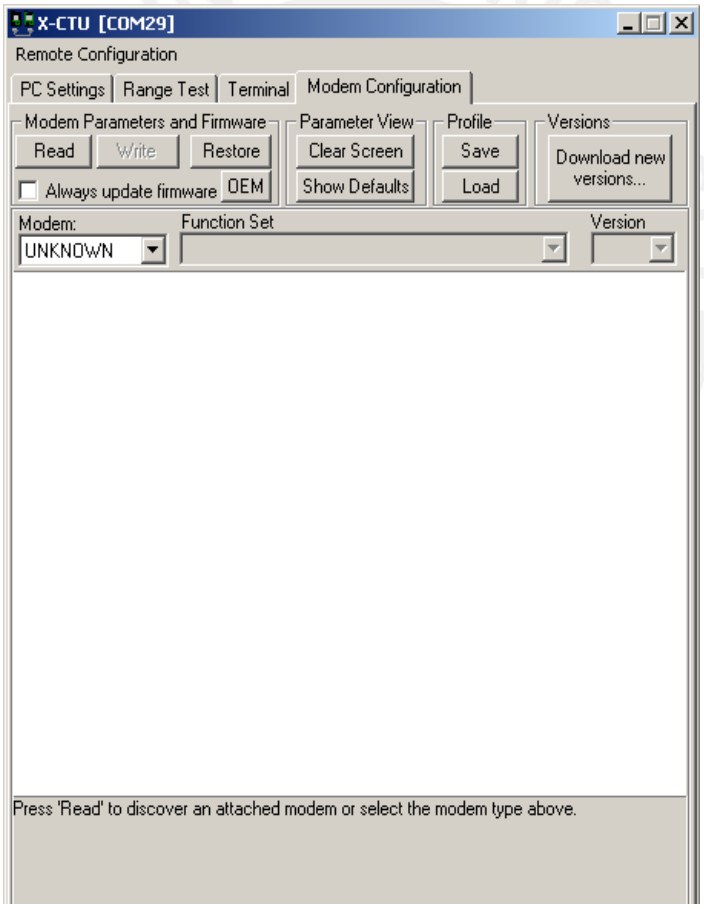
To find out how to load the saved profiles in a production environment from a DOS prompt, please follow the steps outlined in Digi's online Knowledgebase at http://www.maxstream.net/support/knowledgebase/article.php?kb=126

XBee 802.15.4 modules with firmware version 1xCx and above, XBee ZNet 2.5 modules, and XBee ZB modules offer the ability to be configured with over the air commands. With the addition of this new feature, the user is able to configure remote radio parameters with X-CTU or API packets. To use the remote configuration tool, the following is required:

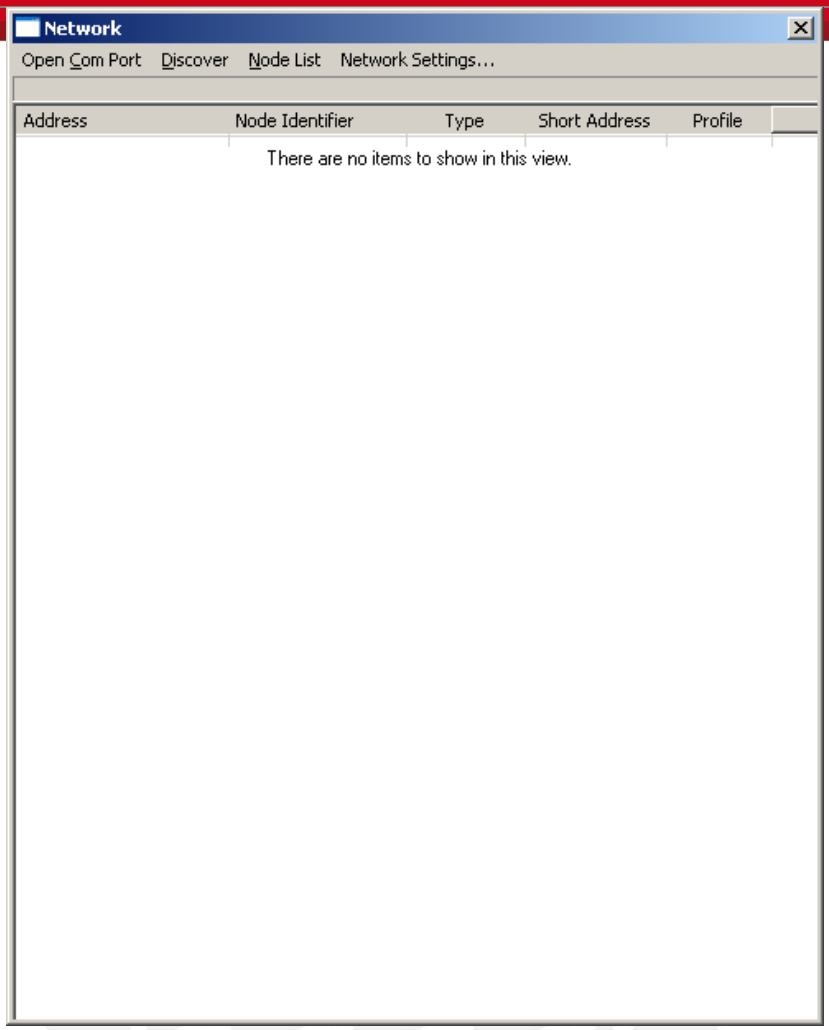
- The radio connected to the PC must be in API mode
- The remote radio must be associated or within range of the base radio

To access remote radios through X-CTU's Modem Configuration tab, perform the steps below:

- Enable API on the PC Settings tab
- Verify the COM port selection and settings
- On the Modem Configuration tab, select the Remote Configuration option on the top left corner of the program



- Select Open Com port
- Select Discover



- Select the desired modem from the discovered node list
- On the Modem configuration tab, select Read

The remote radio's configuration is now displayed on the Modem Configuration tab. At this point, the same options exist with respect to Read and Write parameter changes. Please note that the ability to change firmware versions is still limited to the radio's UART.

To clear the discovered node list, click on Node List and Clear.

The Node List option provides several additional options, including:

- Ability to print the discovered list
- Ability to remove a specific node from a list
- Ability to add additional nodes that have not been discovered
- Save the Node List
- Load a saved Node List
- Select/filter All, Routers, or End nodes

For specific questions related to the X-CTU configuration and test utility software, please contact our Support department, Mon – Fri, 8am – 5pm U.S. Mountain Time:

US and Canada Toll free:
(866)765-9885

Local or International calls:
(801) 765-9885

Online support: http://www.digi.com/support/eservice/login.jsp