PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

ANEXOS

Lima, julio del 2015

INDICE

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 8K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1K Byte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C[(1)]
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Three PWM Channels
  - 8-channel ADC in TQFP and QFN/MLF package
    Eight Channels 10-bit Accuracy
  - 6-channel ADC in PDIP package
    Six Channels 10-bit Accuracy
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V (ATmega8L)
  - 4.5 - 5.5V (ATmega8)
- Speed Grades
  - 0 - 8 MHz (ATmega8L)
  - 0 - 16 MHz (ATmega8)
- Power Consumption at 4 Mhz, 3V, 25°C
  - Active: 3.6 mA
  - Idle Mode: 1.0 mA
  - Power-down Mode: 0.5 µA

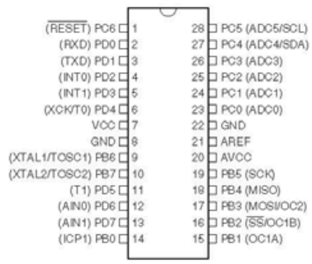
8-bit AVR® with 8K Bytes In-System Programmable Flash

ATmega8 ATmega8L

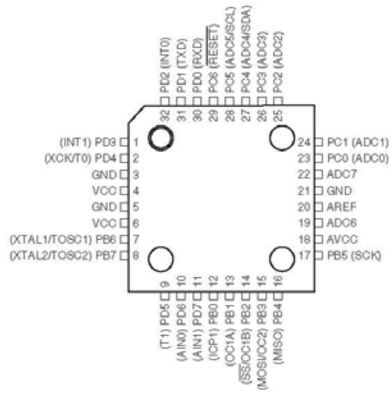
Summary

Pin Configurations

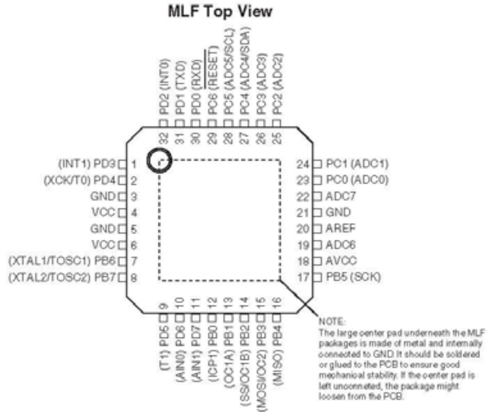
PDIP



TQFP Top View



MLF Top View

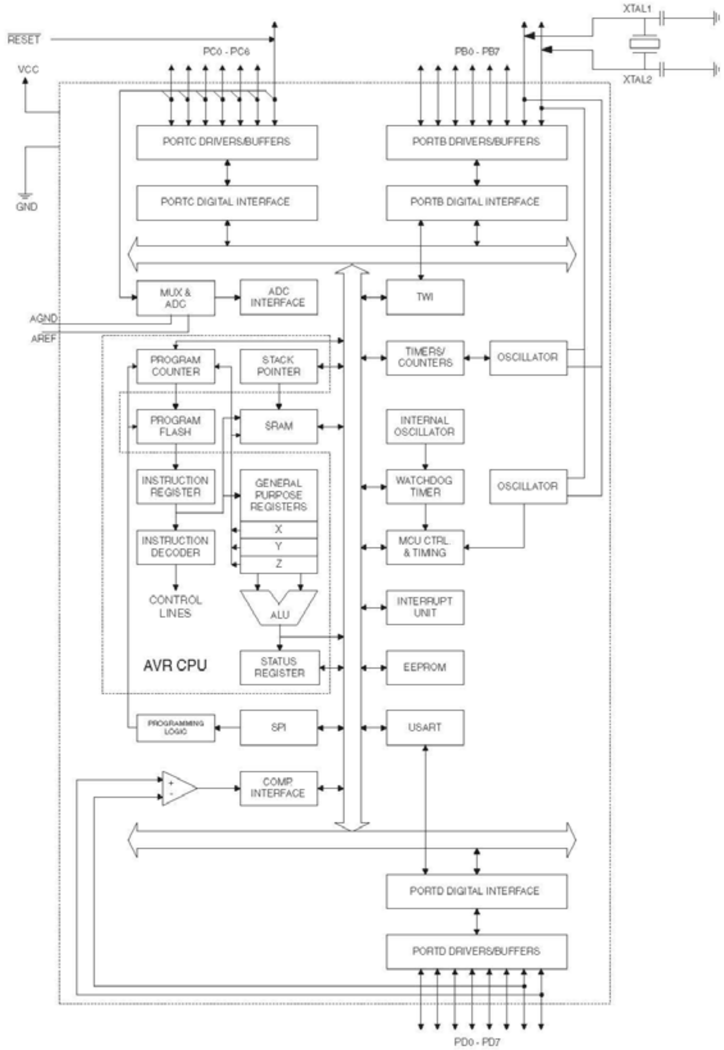


Overview

The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 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 ATmega8 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1K byte of 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 Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next 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 Flash Program memory can 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 ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega8 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.

Disclaimer

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

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port B (PB7..PB0) 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 58 and "System Clock and Clock Options" on page 25.

Port C (PC5..PC0)

Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

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 15 on page 38. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated on page 61.

Port D (PD7..PD0)

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

Port D also serves the functions of various special features of the ATmega8 as listed on page 63.

$\overline{\text{RESET}}$

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

AV_{CC} AV_{CC} is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..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 Port C (5..4) use digital supply voltage, V_{CC}.

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

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.

Resources

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

Data Retention

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

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	11
0x3E (0x5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	13
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	13
0x3C (0x5C)	Reserved									
0x3B (0x5B)	GICR	INT1	INT0	–	–	–	–	IVSEL	IVCE	49, 67
0x3A (0x5A)	GIFR	INTF1	INTF0	–	–	–	–	–	–	68
0x39 (0x59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	72, 102, 122
0x38 (0x58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0	73, 102, 122
0x37 (0x57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	213
0x36 (0x56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	171
0x35 (0x55)	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	33, 66
0x34 (0x54)	MCUCSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	41
0x33 (0x53)	TCCR0	–	–	–	–	–	CS02	CS01	CS00	72
0x32 (0x52)	TCNT0	Timer/Counter0 (8 Bits)								72
0x31 (0x51)	OSCCAL	Oscillator Calibration Register								31
0x30 (0x50)	SFIOR	–	–	–	–	ACME	PUD	PSR2	PSR10	58, 75, 123, 193
0x2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	96
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	100
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High byte								101
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low byte								101
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High byte								101
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low byte								101
0x29 (0x49)	OCR1BH	Timer/Counter1 – Output Compare Register B High byte								101
0x28 (0x48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low byte								101
0x27 (0x47)	ICR1H	Timer/Counter1 – Input Capture Register High byte								102
0x26 (0x46)	ICR1L	Timer/Counter1 – Input Capture Register Low byte								102
0x25 (0x45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	117
0x24 (0x44)	TCNT2	Timer/Counter2 (8 Bits)								119
0x23 (0x43)	OCR2	Timer/Counter2 Output Compare Register								119
0x22 (0x42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	119
0x21 (0x41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	43
0x20[1] (0x40)[1]	UBRRH	URSEL	–	–	–	UBRR[11:8]				158
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	156
0x1F (0x3F)	EEARH	–	–	–	–	–	–	–	EEAR8	20
0x1E (0x3E)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	20
0x1D (0x3D)	EEDR	EEPROM Data Register								20
0x1C (0x3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	20
0x1B (0x3B)	Reserved									
0x1A (0x3A)	Reserved									
0x19 (0x39)	Reserved									
0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	65
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	65
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	65
0x15 (0x35)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
0x14 (0x34)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
0x13 (0x33)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
0x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
0x11 (0x31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
0x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
0x0F (0x2F)	SPDR	SPI Data Register								131
0x0E (0x2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	131
0x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	129
0x0C (0x2C)	UDR	USART I/O Data Register								153
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	154
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	155
0x09 (0x29)	UBRRL	USART Baud Rate Register Low byte								158
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	194
0x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	205
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	207
0x05 (0x25)	ADCH	ADC Data Register High byte								208
0x04 (0x24)	ADCL	ADC Data Register Low byte								208
0x03 (0x23)	TWDR	Two-wire Serial Interface Data Register								173
0x02 (0x22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	174

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x01 (0x21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	173
0x00 (0x20)	TWBR	Two-wire Serial Interface Bit Rate Register								171

Notes:
1. Refer to the USART description for details on how to access UBRRH and UCSRC.
2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
3. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← 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
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
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**

2486TS–AVR–05/08

Instruction Set Summary (Continued)

BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
Mnemonics	**Operands**	**Description**	**Operation**	**Flags**	**#Clocks**

Instruction Set Summary (Continued)

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
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

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package[(1)]	Operation Range
8	2.7 - 5.5	ATmega8L-8AU[(2)] ATmega8L-8PU[(2)] ATmega8L-8MU[(2)]	32A 28P3 32M1-A	Industrial (-40°C to 85°C)
16	4.5 - 5.5	ATmega8-16AU[(2)] ATmega8-16PU[(2)] ATmega8-16MU[(2)]	32A 28P3 32M1-A	Industrial (-40°C to 85°C)

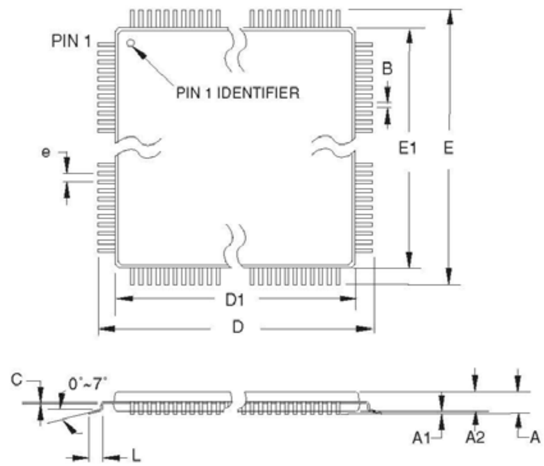
Notes: 1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

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)

Packaging Information

32A



COMMON DIMENSIONS
(Unit of Measure = mm)

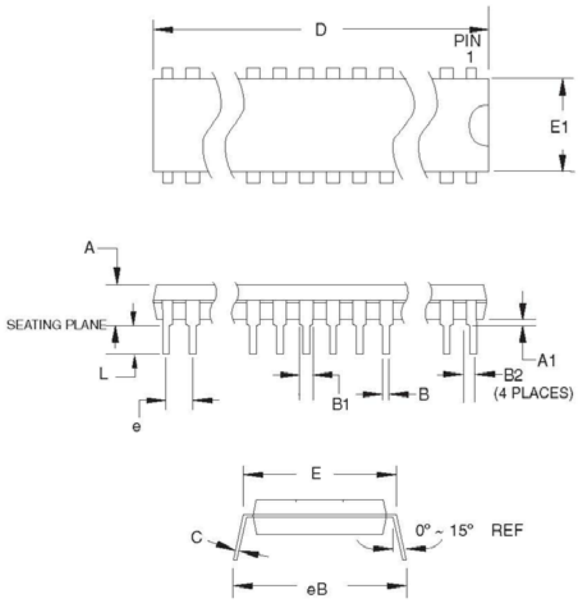
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e		0.80 TYP		

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

28P3



COMMON DIMENSIONS
(Unit of Measure = mm)

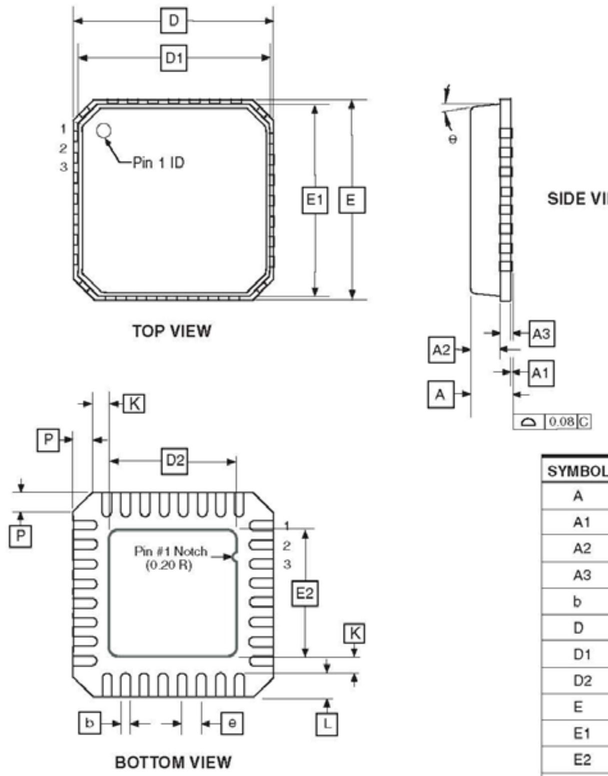
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.5724	
A1	0.508	–	–	
D	34.544	–	34.798	Note 1
E	7.620	–	8.255	
E1	7.112	–	7.493	Note 1
B	0.381	–	0.533	
B1	1.143	–	1.397	
B2	0.762	–	1.143	
L	3.175	–	3.429	
C	0.203	–	0.356	
eB	–	–	10.160	
e	2.540 TYP			

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.
ATMEL 2325 Orchard Parkway San Jose, CA 95131	**28P3**, 28-lead (0.300"/7.62 mm Wide) Plastic Dual Inline Package (PDIP)	28P3	B

32M1-A



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°	
K	0.20	–	–	

Note: JEDEC Standard MO-220, Fig. 2 (Anvil Singulation), VHHD-2.

5/25/06

ATMEL	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

Errata

The revision letter in this section refers to the revision of the ATmega8 device.

ATmega8 Rev. D to I

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **Signature may be Erased in Serial Programming Mode**
- **CKOPT Does not Enable Internal Capacitors on XTALn/TOSCn Pins when 32 KHz Oscillator is Used to Clock the Asynchronous Timer/Counter2**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. **First Analog Comparator conversion may be delayed**

   If the device is powered by a slow rising V_{CC}, the first Analog Comparator conversion will take longer than expected on some devices.

   Problem Fix / Workaround

   When the device has been powered or reset, disable then enable theAnalog Comparator before the first conversion.

2. **Interrupts may be lost when writing the timer registers in the asynchronous timer**

   If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

   Problem Fix / Workaround

   Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

3. **Signature may be Erased in Serial Programming Mode**

   If the signature bytes are read before a chiperase command is completed, the signature may be erased causing the device ID and calibration bytes to disappear. This is critical, especially, if the part is running on internal RC oscillator.

   Problem Fix / Workaround:

   Ensure that the chiperase command has exceeded before applying the next command.

4. **CKOPT Does not Enable Internal Capacitors on XTALn/TOSCn Pins when 32 KHz Oscillator is Used to Clock the Asynchronous Timer/Counter2**

   When the internal RC Oscillator is used as the main clock source, it is possible to run the Timer/Counter2 asynchronously by connecting a 32 KHz Oscillator between XTAL1/TOSC1 and XTAL2/TOSC2. But when the internal RC Oscillator is selected as the main clock source, the CKOPT Fuse does not control the internal capacitors on XTAL1/TOSC1 and XTAL2/TOSC2. As long as there are no capacitors connected to XTAL1/TOSC1 and XTAL2/TOSC2, safe operation of the Oscillator is not guaranteed.

   Problem Fix / Workaround

   Use external capacitors in the range of 20 - 36 pF on XTAL1/TOSC1 and XTAL2/TOSC2. This will be fixed in ATmega8 Rev. G where the CKOPT Fuse will control internal capacitors also when internal RC Oscillator is selected as main clock source. For ATmega8 Rev. G, CKOPT = 0 (programmed) will enable the internal capacitors on XTAL1 and XTAL2. Customers who want compatibility between Rev. G and older revisions, must ensure that CKOPT is unprogrammed (CKOPT = 1).

5. **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.**

   Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

Problem Fix / Workaround

Always use OUT or SBI to set EERE in EECR.

Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

Changes from Rev. 2486S- 08/07 to Rev. 2486T- 05/08

1. Updated Table 98 on page 240.
2. Updated "Ordering Information" on page 292.
   - Commercial Ordering Code removed.
   - No Pb-free packaging option removed.

Changes from Rev. 2486R- 07/07 to Rev. 2486S- 08/07

1. Updated "Features" on page 1.
2. Added "Data Retention" on page 7.
3. Updated "Errata" on page 17.
4. Updated "Slave Mode" on page 129.

Changes from Rev. 2486Q- 10/06 to Rev. 2486R- 07/07

1. Added text to Table 81 on page 218.
2. Fixed typo in "Peripheral Features" on page 1.
3. Updated Table 16 on page 42.
4. Updated Table 75 on page 206.
5. Removed redundancy and updated typo in Notes section of "DC Characteristics" on page 242.

Changes from Rev. 2486P- 02/06 to Rev. 2486Q- 10/06

1. Updated "Timer/Counter Oscillator" on page 32.
2. Updated "Fast PWM Mode" on page 89.
3. Updated code example in "USART Initialization" on page 138.
4. Updated Table 37 on page 97, Table 39 on page 98, Table 42 on page 117, Table 44 on page 118, and Table 98 on page 240.
5. Updated "Errata" on page 17.

Changes from Rev. 2486O-10/04 to Rev. 2486P- 02/06

1. Added "Resources" on page 7.
2. Updated "External Clock" on page 32.
3. Updated "Serial Peripheral Interface – SPI" on page 124.
4. Updated Code Example in "USART Initialization" on page 138.

5. Updated Note in "Bit Rate Generator Unit" on page 170.
6. Updated Table 98 on page 240.
7. Updated Note in Table 103 on page 248.
8. Updated "Errata" on page 17.

Changes from Rev. 2486N-09/04 to Rev. 2486O-10/04

1. Removed to instances of "analog ground". Replaced by "ground".
2. Updated Table 7 on page 29, Table 15 on page 38, and Table 100 on page 244.
3. Updated "Calibrated Internal RC Oscillator" on page 30 with the 1 MHz default value.
4. Table 89 on page 225 and Table 90 on page 225 moved to new section "Page Size" on page 225.
5. Updated descripton for bit 4 in "Store Program Memory Control Register – SPMCR" on page 213.
6. Updated "Ordering Information" on page 13.

Changes from Rev. 2486M-12/03 to Rev. 2486N-09/04

1. Added note to MLF package in "Pin Configurations" on page 2.
2. Updated "Internal Voltage Reference Characteristics" on page 42.
3. Updated "DC Characteristics" on page 242.
4. ADC4 and ADC5 support 10-bit accuracy. Document updated to reflect this.
   Updated features in "Analog-to-Digital Converter" on page 196.
   Updated "ADC Characteristics" on page 248.
5. Removed reference to "External RC Oscillator application note" from "External RC Oscillator" on page 28.

Changes from Rev. 2486L-10/03 to Rev. 2486M-12/03

1. Updated "Calibrated Internal RC Oscillator" on page 30.

Changes from Rev. 2486K-08/03 to Rev. 2486L-10/03

1. Removed "Preliminary" and TBDs from the datasheet.
2. Renamed ICP to ICP1 in the datasheet.
3. Removed instructions CALL and JMP from the datasheet.
4. Updated t_{RST} in Table 15 on page 38, V_{BG} in Table 16 on page 42, Table 100 on page 244 and Table 102 on page 246.
5. Replaced text "XTAL1 and XTAL2 should be left unconnected (NC)" after Table 9 in "Calibrated Internal RC Oscillator" on page 30. Added text regarding XTAL1/XTAL2 and CKOPT Fuse in "Timer/Counter Oscillator" on page 32.

6. Updated Watchdog Timer code examples in "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 45.
7. Removed bit 4, ADHSM, from "Special Function IO Register – SFIOR" on page 58.
8. Added note 2 to Figure 103 on page 215.
9. Updated item 4 in the "Serial Programming Algorithm" on page 238.
10. Added t_{WD_FUSE} to Table 97 on page 239 and updated Read Calibration Byte, Byte 3, in Table 98 on page 240.
11. Updated Absolute Maximum Ratings* and DC Characteristics in "Electrical Characteristics" on page 242.

Changes from Rev. 2486J-02/03 to Rev. 2486K-08/03

1. Updated V_{BOT} values in Table 15 on page 38.
2. Updated "ADC Characteristics" on page 248.
3. Updated "ATmega8 Typical Characteristics" on page 249.
4. Updated "Errata" on page 17.

Changes from Rev. 2486I-12/02 to Rev. 2486J-02/03

1. Improved the description of "Asynchronous Timer Clock – clkASY" on page 26.
2. Removed reference to the "Multipurpose Oscillator" application note and the "32 kHz Crystal Oscillator" application note, which do not exist.
3. Corrected OCn waveforms in Figure 38 on page 90.
4. Various minor Timer 1 corrections.
5. Various minor TWI corrections.
6. Added note under "Filling the Temporary Buffer (Page Loading)" on page 216 about writing to the EEPROM during an SPM Page load.
7. Removed ADHSM completely.
8. Added section "EEPROM Write during Power-down Sleep Mode" on page 23.
9. Removed XTAL1 and XTAL2 description on page 5 because they were already described as part of "Port B (PB7..PB0) XTAL1/XTAL2/TOSC1/TOSC2" on page 5.
10. Improved the table under "SPI Timing Characteristics" on page 246 and removed the table under "SPI Serial Programming Characteristics" on page 241.
11. Corrected PC6 in "Alternate Functions of Port C" on page 61.
12. Corrected PB6 and PB7 in "Alternate Functions of Port B" on page 58.
13. Corrected 230.4 Mbps to 230.4 kbps under "Examples of Baud Rate Setting" on page 159.

14. Added information about PWM symmetry for Timer 2 in "Phase Correct PWM Mode" on page 113.

15. Added thick lines around accessible registers in Figure 76 on page 169.

16. Changed "will be ignored" to "must be written to zero" for unused Z-pointer bits under "Performing a Page Write" on page 216.

17. Added note for RSTDISBL Fuse in Table 87 on page 223.

18. Updated drawings in "Packaging Information" on page 14.

Changes from Rev. 2486H-09/02 to Rev. 2486I-12/02

1. Added errata for Rev D, E, and F on page 17.

Changes from Rev. 2486G-09/02 to Rev. 2486H-09/02

1. Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.

Changes from Rev. 2486F-07/02 to Rev. 2486G-09/02

1. Updated Table 103, "ADC Characteristics," on page 248.

Changes from Rev. 2486E-06/02 to Rev. 2486F-07/02

1. Changes in "Digital Input Enable and Sleep Modes" on page 55.

2. Addition of OCS2 in "MOSI/OC2 – Port B, Bit 3" on page 59.

3. The following tables have been updated:

   Table 51, "CPOL and CPHA Functionality," on page 132, Table 59, "UCPOL Bit Settings," on page 158, Table 72, "Analog Comparator Multiplexed Input(1)," on page 195, Table 73, "ADC Conversion Time," on page 200, Table 75, "Input Channel Selections," on page 206, and Table 84, "Explanation of Different Variables used in Figure 103 and the Mapping to the Z-pointer," on page 221.

4. Changes in "Reading the Calibration Byte" on page 234.

5. Corrected Errors in Cross References.

Changes from Rev. 2486D-03/02 to Rev. 2486E-06/02

1. Updated Some Preliminary Test Limits and Characterization Data

   The following tables have been updated:

   Table 15, "Reset Characteristics," on page 38, Table 16, "Internal Voltage Reference Characteristics," on page 42, DC Characteristics on page 242, Table , "ADC Characteristics," on page 248.

2. Changes in External Clock Frequency

   Added the description at the end of "External Clock" on page 32.

   Added period changing data in Table 99, "External Clock Drive," on page 244.

3. Updated TWI Chapter

More details regarding use of the TWI bit rate prescaler and a Table 65, "TWI Bit Rate Prescaler," on page 173.

Changes from Rev. 2486C-03/02 to Rev. 2486D-03/02

1. **Updated Typical Start-up Times.**

   The following tables has been updated:

   Table 5, "Start-up Times for the Crystal Oscillator Clock Selection," on page 28, Table 6, "Start-up Times for the Low-frequency Crystal Oscillator Clock Selection," on page 28, Table 8, "Start-up Times for the External RC Oscillator Clock Selection," on page 29, and Table 12, "Start-up Times for the External Clock Selection," on page 32.

2. **Added "ATmega8 Typical Characteristics" on page 249.**

Changes from Rev. 2486B-12/01 to Rev. 2486C-03/02

1. **Updated TWI Chapter.**

   More details regarding use of the TWI Power-down operation and using the TWI as Master with low TWBRR values are added into the datasheet.

   Added the note at the end of the "Bit Rate Generator Unit" on page 170.

   Added the description at the end of "Address Match Unit" on page 170.

2. **Updated Description of OSCCAL Calibration Byte.**

   In the datasheet, it was not explained how to take advantage of the calibration bytes for 2, 4, and 8 MHz Oscillator selections. This is now added in the following sections:

   Improved description of "Oscillator Calibration Register – OSCCAL" on page 31 and "Calibration Byte" on page 225.

3. **Added Some Preliminary Test Limits and Characterization Data.**

   Removed some of the TBD's in the following tables and pages:

   Table 3 on page 26, Table 15 on page 38, Table 16 on page 42, Table 17 on page 44, "TA = -40×C to 85×C, VCC = 2.7V to 5.5V (unless otherwise noted)" on page 242, Table 99 on page 244, and Table 102 on page 246.

4. **Updated Programming Figures.**

   Figure 104 on page 226 and Figure 112 on page 237 are updated to also reflect that AV_{CC} must be connected during Programming mode.

5. **Added a Description on how to Enter Parallel Programming Mode if RESET Pin is Disabled or if External Oscillators are Selected.**

   Added a note in section "Enter Programming Mode" on page 228.

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
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

2486TS–AVR–05/08

AOSONG

Temperature and humidity module

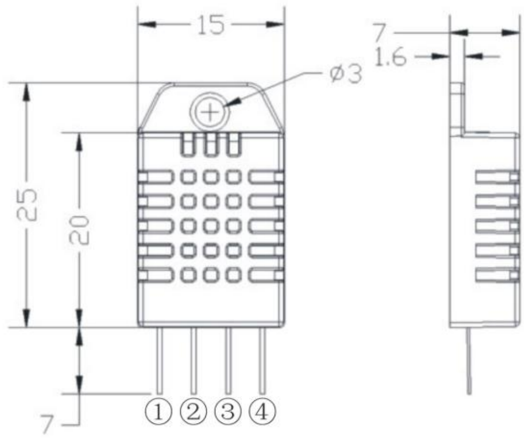
AM2302 Product Manual

www.aosong.com

1、Product Overview

AM2302 capacitive humidity sensing digital temperature and humidity module is one that contains the compound has been calibrated digital signal output of the temperature and humidity sensors. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a capacitive sensor wet components and a high-precision temperature measurement devices, and connected with a high-performance 8-bit microcontroller. The product has excellent quality, fast response, strong anti-jamming capability, and high cost. Each sensor is extremely accurate humidity calibration chamber calibration. The form of procedures, the calibration coefficients stored in the microcontroller, the sensor within the processing of the heartbeat to call these calibration coefficients. Standard single-bus interface, system integration quick and easy. Small size, low power consumption, signal transmission distance up to 20 meters, making it the best choice of all kinds of applications and even the most demanding applications. Products for the 3-lead (single-bus interface) connection convenience. Special packages according to user needs.

Physical map



Dimensions (unit: mm)

2、Applications

HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, home appliances, humidity regulator, medical, weather stations, and other humidity measurement and control and so on.

3、Features

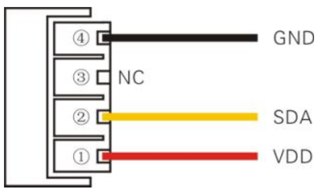
Ultra-low power, the transmission distance, fully automated calibration, the use of capacitive humidity sensor, completely interchangeable, standard digital single-bus output, excellent long-term stability, high accuracy temperature measurement devices.

4、The definition of single−bus interface

4.1 AM2302 Pin assignments

Table 1： AM2302 Pin assignments

Pin	Name	Description
①	VDD	Power (3.3V−5.5V)
②	SDA	Serial data, bidirectional port
③	NC	Empty
④	GND	Ground



PIC1： AM2302 Pin Assignment

4.2 Power supply pins（VDD GND）

AM2302 supply voltage range 3.3V − 5.5V, recommended supply voltage is 5V.

4.3 Serial data（SDA）

SDA pin is tri structure for reading, writing sensor data. Specific communication timing, see the detailed description of the communication protocol.

5、Sensor performance

5.1 Relative humidity

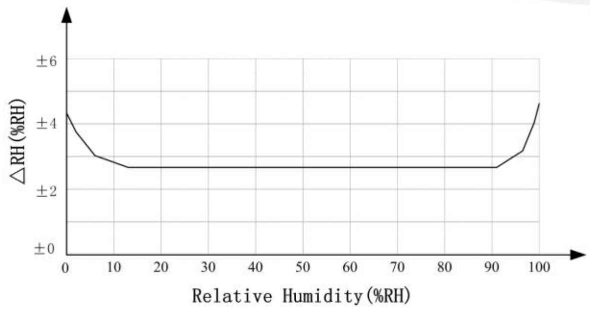
Table 2： AM2302 Relative humidity performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		%RH
Range		0		99.9	%RH
Accuracy [1]	25℃		±2		%RH
Repeatability			±0.3		%RH
Exchange		Completely interchangeable			
Response [2]	1/e(63%)		<5		S
Sluggish			<0.3		%RH
Drift [3]	Typical		<0.5		%RH/yr

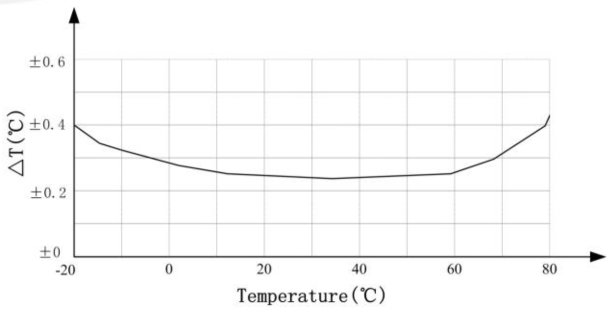
5.2 Temperature

Table 3： AM2302 Relative temperature performance

Parameter	Condition	min	typ	max	Unit
Resolutio			0.1		℃
n			16		bit
Accuracy			±0.5	±1	℃
Range		−40		80	℃
Repeat			±0.2		℃
Exchange		Completely interchangeable			
Response	1/e(63%)		<10		S
Drift			±0.3		℃/yr



Pic2: At25℃ The error of relative humidity



Pic3： The maximum temperature error

6、Electrical Characteristics

Electrical characteristics, such as energy consumption, high, low, input, output voltage, depending on the power supply. Table 4 details the electrical characteristics of the AM2302, if not identified, said supply voltage of 5V. To get the best results with the sensor, please design strictly in accordance with the conditions of design in Table 4.

Table 4: AM2302 DC Characteristics

Parameter	Condition	min	typ	max	Unit
Voltage		3.3	5	5.5	V
Power consumption [4]	Dormancy	10	15		μA
	Measuring		500		μA
	Average		300		μA
Low level output voltage	I_{OL}[5]	0		300	mV
High output voltage	Rp<25 kΩ	90%		100%	VDD
Low input voltage	Decline	0		30%	VDD
Input High Voltage	Rise	70%		100%	VDD
Rpu[6]	VDD = 5V VIN = VSS	30	45	60	kΩ
Output current	turn on		8		mA
	turn off	10	20		μA
Sampling period		2			S

[1] the accuracy of the factory inspection, the sensor 25°C and 5V, the accuracy specification of test conditions, it does not include hysteresis and nonlinearity, and is only suitable for non−condensing environment.

[2] to achieve an order of 63% of the time required under the conditions of 25℃ and 1m / s airflow.

[3] in the volatile organic compounds, the values may be higher. See the manual application to store information.

[4] this value at VDD = 5.0V when the temperature is 25℃, 2S / time, under the conditions of the average.

[5] low output current.

[6] that the pull−up resistor.

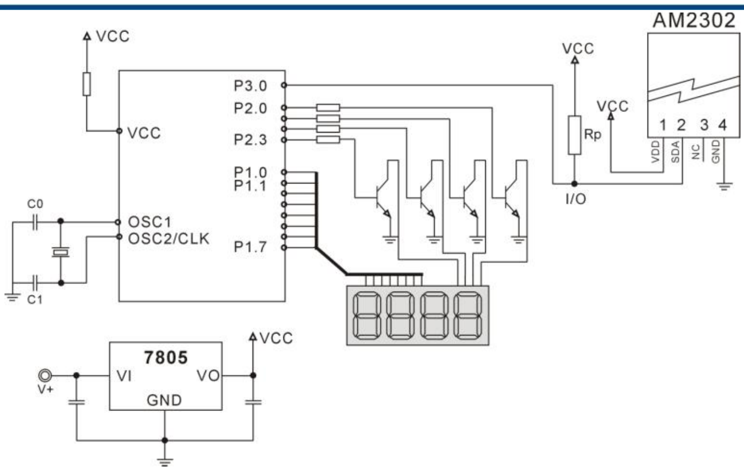
7、Single−bus communication（ONE−WIRE）

7.1 Typical circuits for single bus

Microprocessor and AM2302 connection typical application circuit is shown in Figure 4. Single bus communication mode, pull the SDA microprocessor I / O port is connected.

Special instructions of the single−bus communication：

1. Typical application circuit recommended in the short cable length of 30 meters on the 5.1K pull−up resistor pullup resistor according to the actual situation of lower than 30 m.
2. With 3.3V supply voltage, cable length shall not be greater than 100cm. Otherwise, the line voltage drop will lead to the sensor power supply, resulting in measurement error.
3. Read the sensor minimum time interval for the 2S; read interval is less than 2S, may cause the temperature and humidity are not allowed or communication is unsuccessful, etc..
4. Temperature and humidity values are each read out the results of the last measurement For real−time data that need continuous read twice, we recommend repeatedly to read sensors, and each read sensor interval is greater than 2 seconds to obtain accuratethe data.



Pic4： AM2302 Typical circuits for single bus

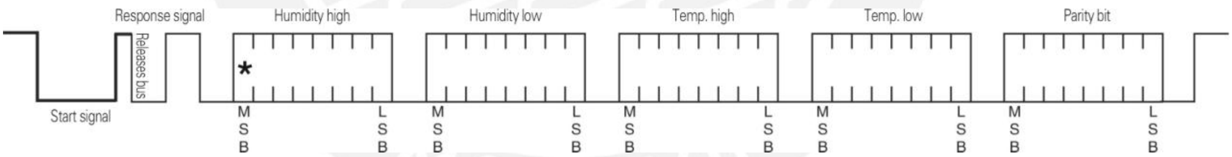
7.2、Single-bus communication protocol

◎Single bus Description

AM2302 device uses a simplified single-bus communication. Single bus that only one data line, data exchange system, controlled by the data line to complete. Equipment (microprocessor) through an open-drain or tri-state port connected to the data line to allow the device does not send data to release the bus, while other devices use the bus; single bus usually require an external about 5.1kΩ pull-up resistor, so when the bus is idle, its status is high. Because they are the master-slave structure, only the host calls the sensor, the sensor will answer, so the hosts to access the sensor must strictly follow the sequence of single bus, if there is a sequence of confusion, the sensor will not respond to the host.

◎Single bus to send data definition

SDA For communication and synchronization between the microprocessor and the AM2302, single-bus data format, a transmission of 40 data, the high first-out. Specific communication timing shown in Figure 5, the communication format is depicted in Table 5.



Pic5： AM2302 Single-bus communication protocol

Table 5：AM2302 Communication format specifier

Name	Single–bus format definition
Start signal	Microprocessor data bus (SDA) to bring down a period of time (at least 800μ s) [1] notify the sensor to prepare the data.
Response signal	Sensor data bus (SDA) is pulled down to 80μ s, followed by high–80μ s response to host the start signal.
Data format	Host the start signal is received, the sensor one–time string from the data bus (SDA) 40 data, the high first–out.
Humidity	Humidity resolution of 16Bit, the previous high; humidity sensor string value is 10 times the actual humidity values.
Temp.	Temperature resolution of 16Bit, the previous high; temperature sensor string value is 10 times the actual temperature value; The temperature is the highest bit (Bit15) is equal to 1 indicates a negative temperature, the temperature is the highest bit (Bit15) is equal to 0 indicates a positive temperature; Temperature in addition to the most significant bit (Bit14 ~ bit 0) temperature values.
Parity bit	Parity bit = humidity high + humidity low + temperature high + temperature low

◎**Single–bus data calculation example**

Example 1：40 Data received：

0000 0010	1001 0010	0000 0001	0000 1101	1010 0010
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Calculate：

0000 0010+1001 0010 +0000 0001+0000 1101= 1010 0010（Parity bit）

Received data is correct：

humidity：0000 0010 1001 0010 = 0292H (Hexadecimal)= 2×256 + 9×16 + 2 = 658

=> Humidity = 65.8%RH

Temp.：0000 0001 0000 1101 = 10DH(Hexadecimal) = 1×256 + 0×16 + 13 = 269

=> Temp.= 26.9℃

◎**Special Instructions**：

When the temperature is below 0 ℃, the highest position of the temperature data.

Example： −10.1 ℃ Expressed as 1 000 0000 0110 0101

Temp.：0000 0000 0110 0101 = 0065H(Hexadecimal)＝ 6×16 +5 = 101

=> Temp. ＝ −10.1℃

Example 2：40 received data：

0000 0010	1001 0010	0000 0001	0000 1101	1011 0010
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Calculate：

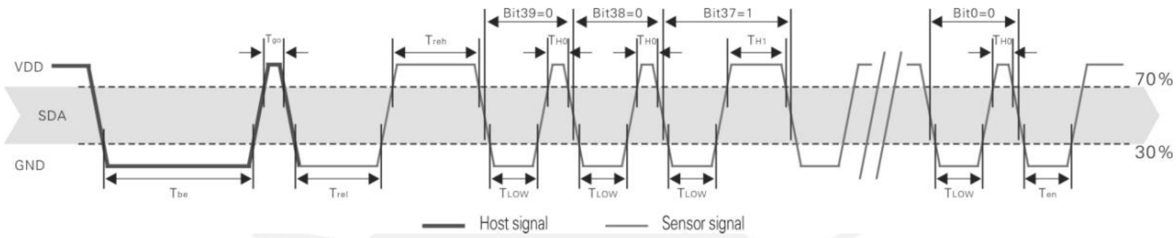
0000 0010+1001 0010 +0000 0001+0000 1101= 1010 0010 ≠ 1011 0010 （Validation error）

The received data is not correct, give up, to re–receive data.

7.3 Single-bus communication timing

User host (MCU) to send a start signal (data bus SDA line low for at least 800μ s) after AM2302 from Sleep mode conversion to high-speed mode.The host began to signal the end of the AM2302 send a response signal sent from the data bus SDA serial 40Bit's data, sends the byte high; data sent is followed by: Humidity high、Humidity low、Temperature high、Temperature low、Parity bit，Send data to the end of trigger information collection, the collection end of the sensor is automatically transferred to the sleep mode, the advent until the next communication.

Detailed timing signal characteristics in Table 6，Single-bus communication timing diagram Pic 6：



Pic 6： AM2302 Single-bus communication timing

Note: the temperature and humidity data read by the host from the AM2302 is always the last measured value, such as the two measurement interval is very long, continuous read twice to the second value of real-time temperature and humidity values, while two readtake minimum time interval be 2S.

Table 6: Single bus signal characteristics

Symbol	Parameter	min	typ	max	Unit
T_{be}	Host the start signal down time	0.8	1	20	mS
T_{go}	Bus master has released time	20	30	200	μS
T_{rel}	Response to low time	75	80	85	μS
T_{reh}	In response to high time	75	80	85	μS
T_{LOW}	Signal "0", "1" low time	48	50	55	μS
T_{H0}	Signal "0" high time	22	26	30	μS
T_{H1}	Signal "1" high time	68	70	75	μS
T_{en}	Sensor to release the bus time	45	50	55	μS

Note： To ensure the accurate communication of the sensor, the read signal, in strict accordance with the design parameters and timing in Table 6 and Figure 6.

7.4 Peripherals read step example

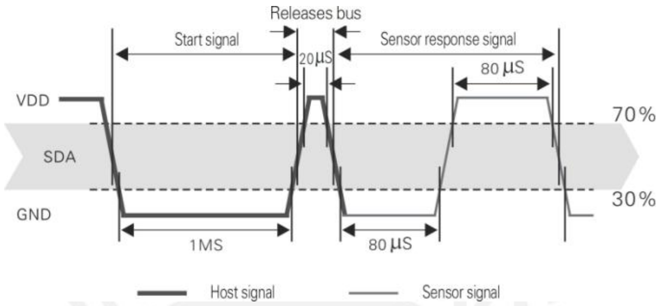
Communication between the host and the sensor can read data through the following three steps to complete.

Step 1

AM2302 have to wait for the power (on AM2302 power 2S crossed the unstable state, the device can not send any instructions to read during this period), the test environment temperature and humidity data, and record data, since the sensor into a sleep state automatically. AM2302 The SDA data line from the previous pull-up resistor pulled up is always high, the AM2302 the SDA pin is in input state, the time detection of external signal.

Step 2

Microprocessor I/O set to output, while output low, and low hold time can not be less than 800us, typical values are down 1MS, then the microprocessor I/O is set to input state, the release of the bus, due to the pull−up resistor, the microprocessor I/O AM2302 the SDA data line also will be high, the bus master has released the AM2302 send a response signal, that is, the output 80 microseconds low as the response signal, tightthen output high of 80 microseconds notice peripheral is ready to receive data signal transmission as shown to Pic7 :



Pic7：Single bus decomposition of the timing diagram

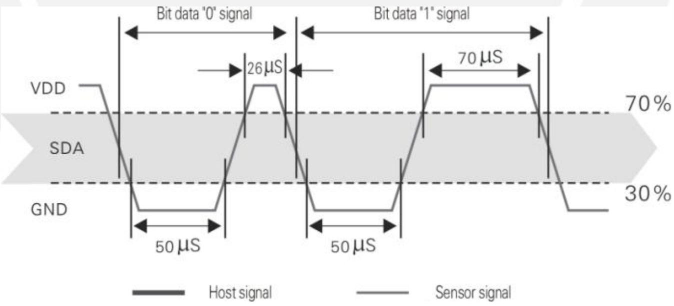
Step 3

AM2302 sending the response, followed by the data bus SDA continuous serial output 40 data, the microprocessor receives 40 data I/O level changes.

Bit data "0" format: 26−28 microseconds 50 microseconds low plus high;

Bit data "1" format: the high level of low plus, 50 microseconds to 70 microseconds;

Bit data "0" bit data "1" format signal shown to pic 8:

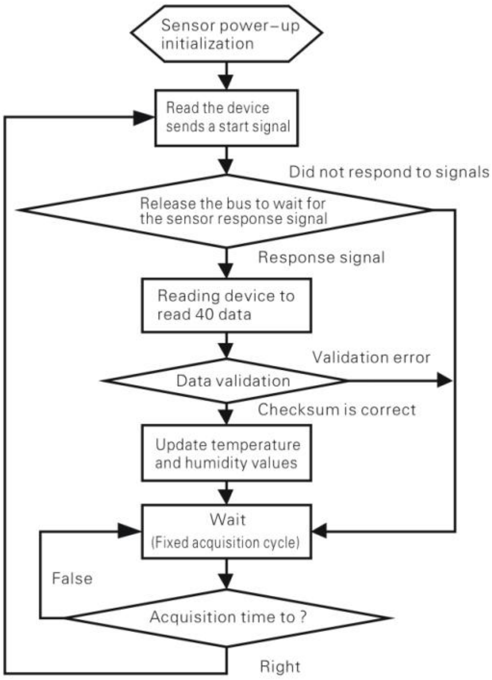


Pic 8：The single bus break down the timing diagram

AM2302 data bus SDA output 40 data continue to output the low 50 microseconds into the input state, followed by pull−up resistor goes high. AM2302 internal re−test environmental temperature and humidity data, and record the data, the end of the test records, the microcontroller automatically into hibernation. Microcontroller only after receipt of the start signal of the host wake−up sensor, into the working state.

7.5 Peripheral to read flow chart

AM2302 sensor read single bus flow chart diagram shown in Figure 9, we also provide the C51 read the code examples, customers need to download, please visit our website (www.aosong.com) related to downloadthis manual does not provide the code description.



Pic9: Single-bus to read the flow chart

8、Application of information

1. Work and storage conditions

Outside the sensor the proposed scope of work may lead to temporary drift of the signal up to 300% RH. Return to normal working conditions, sensor calibration status will slowly toward recovery. To speed up the recovery process may refer to "resume processing". Prolonged use of non-normal operating conditions, will accelerate the aging of the product.

Avoid placing the components on the long-term condensation and dry environment, as well as the following environment.

A, salt spray

B, acidic or oxidizing gases such as sulfur dioxide, hydrochloric acid

Recommended storage environment

Temperature: 10 ~ 40 ℃ Humidity: 60% RH or less

2. The impact of exposure to chemicals

The capacitive humidity sensor has a layer by chemical vapor interference, the proliferation of chemicals in the sensing layer may lead to drift and decreased sensitivity of the measured values. In a pure environment, contaminants will slowly be released. Resume processing as described below will accelerate this process. The high concentration of chemical pollution (such as ethanol) will lead to the complete damage of the sensitive layer of the sensor.

3. The temperature influence

Relative humidity of the gas to a large extent dependent on temperature. Therefore, in the measurement of humidity, should be to ensure that the work of the humidity sensor at the same temperature. With the release of heat of electronic components share a printed circuit board, the installation should be as far as possible the sensor away from the electronic components and mounted below the heat source, while maintaining good ventilation of the enclosure. To reduce the thermal conductivity sensor and printed circuit board copper plating should be the smallest possible, and leaving a gap between the two.

4. Light impact

Prolonged exposure to sunlight or strong ultraviolet radiation, and degrade performance.

5. Resume processing

Placed under extreme working conditions or chemical vapor sensor, which allows it to return to the status of calibration by the following handler. Maintain two hours in the humidity conditions of 45℃ and <10% RH (dry); followed by 20−30℃ and> 70% RH humidity conditions to maintain more than five hours.

6. Wiring precautions

The quality of the signal wire will affect the quality of the voltage output, it is recommended to use high quality shielded cable.

7. Welding information

Manual welding, in the maximum temperature of 300℃ under the conditions of contact time shall be less than 3 seconds.

8. Product upgrades

Details, please the consultation Aosong electronics department.

9、The license agreement

Without the prior written permission of the copyright holder, shall not in any form or by any means, electronic or mechanical (including photocopying), copy any part of this manual, nor shall its contents be communicated to a third party. The contents are subject to change without notice.

The Company and third parties have ownership of the software, the user may use only signed a contract or software license.

10、Warnings and personal injury

This product is not applied to the safety or emergency stop devices, as well as the failure of the product may result in injury to any other application, unless a particular purpose or use authorized. Installation, handling, use or maintenance of the product refer to product data sheets and application notes. Failure to comply with this recommendation may result in death and serious personal injury. The Company will bear all damages resulting personal injury or death, and waive any claims that the resulting subsidiary company managers and employees and agents, distributors, etc. that may arise, including: a variety of costs, compensation costs, attorneys' fees, and so on.

11、Quality Assurance

The company and its direct purchaser of the product quality guarantee period of three months (from the date of delivery). Publishes the technical specifications of the product data sheet shall prevail. Within the warranty period, the product was confirmed that the quality is really defective, the company will provide free repair or replacement. The user must satisfy the following conditions:

① The product is found defective within 14 days written notice to the Company;

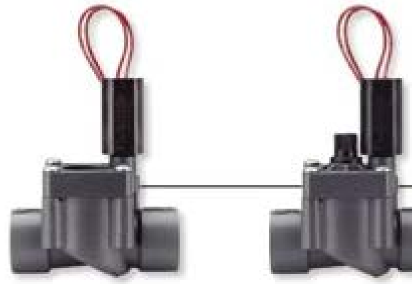
② The product shall be paid by mail back to the company;

③ The product should be within the warranty period.

The Company is only responsible for those used in the occasion of the technical condition of the product defective product. Without any guarantee, warranty or written statement of its products used in special applications. Company for its products applied to the reliability of the product or circuit does not make any commitment.

Hunter®

Electroválvula SRV



SRV

La válvula económica que asegura fiabilidad a las instalaciones residenciales

¿Quién dijo que rendimiento fiable y precio asequible eran términos incompatibles en lo que se refiere a electroválvulas residenciales? Con Hunter puede obtener ambas características, además de la típica construcción resistente de un precio más elevado. Nuestra SRV ofrece un cuerpo en línea de PVC de gran calidad y duración, así como un diafragma reforzado diseñado para hacer frente al gran uso diario. Además, la SRV cuenta con un soporte de diafragma para evitar deformaciones, una purga manual interna que mantiene seca la arqueta de electroválvulas y el resistente solenoide de Hunter. Asimismo, para satisfacer las necesidades individuales de cada sistema que instale, la SRV está disponible en los modelos con regulador de caudal y sin regulador de caudal. La sencilla y fiable SRV de Hunter se adapta a cualquier presupuesto de instalación de riego.

CARACTERISTICAS Y VENTAJAS

Solenoide Hunter para uso intensivo
Duración y fiabilidad

Construcción de gran calidad
Fabricado en PVC duradero y de acero inoxidable para su larga duración

Arranque manual sin pérdida de agua
La arqueta permanece seca

Modelo con la opción de regulador de caudal
Facilita la regulación de caudal

Modelo de conexión para encolar
Permite la conexión directa del cuerpo a tuberías de PVC

Construcción resistente de PVC
Material de gran duración

Modelo

SRV-100G-B – Electroválvula de 1" BSP en línea

SRV-101G-B – Electroválvula de 1" BSP con regulación de caudal

Dimensiones

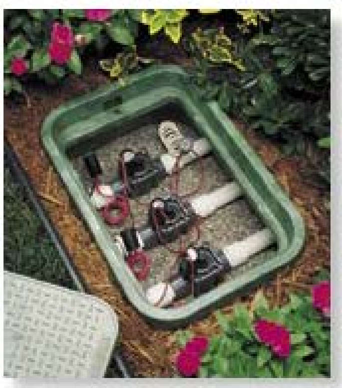
- Altura: 13 cm
- Longitud: 11 cm
- Ancho: 6 cm

Especificaciones

- Caudal: 0,23 a 6,81 m3/h; 4 a 114 l/min
- Presión: 1,38 a 10,34 bares; 137,9 a 1034,21 kPa
- Solenoide de gran resistencia: 24VCA, corriente de arranque 370mA, corriente de mantenimiento 190mA, 60 ciclos, corriente de arranque de agua 475mA, corriente de mantenimiento 230mA, 50 ciclos

Opciones

- Maneta identificativa de agua reciclada o no potable sólo para modelos de regulador de caudal (# 269205)
- Solenoide latch de CC (# 458200)
- Tapa de conducto de solenoide (# 464322)



Prioridad en la Eficacia

Desde que se introdujo en el mercado la electroválvula SRV, su fiabilidad y sus altos rendimientos la hicieron inigualable. Sólo le faltaba una cosa: el ajuste del caudal. Ya lo tiene. ¡Ya existe la electroválvula con esta opción! Con este modelo puede afinar el ajuste del caudal y la presión del riego de cada zona de su instalación. ¡El riego se optimiza y sus céspedes y jardines son más bonitos!

SRV – Pérdida de carga en bares

m³/hr	1" en línea
0,23	0,08
1,14	0,18
2,27	0,18
3,41	0,11
4,54	0,23
5,68	0,34
6,81	0,42

SRV – Pérdida de carga en kPa

l/min	1" en línea
4,0	7,58
19,0	18,10
38,0	18,10
57,0	11,03
76,0	22,75
95,0	34,47
114,0	42,06

ESPECIFICACIONES TÉCNICAS

EJEMPLO: SRV - 100G - B - DC

MODELO	CARACTERÍSTICAS	OPCIONES – INSTALADAS DE FÁBRICA	OPCIONES – INSTALADAS POR EL PROPIO USUARIO
SRV	100G = 1" en línea, sin regulación de caudal 101G = 1" en línea, con regulación de caudal	S : Estándar americano B : Rosca BSP	R : Maneta identificativa de agua reciclada DC : Solenoide de impulsos CC : Tapa de conducto de solenoide

Hunter Industries Incorporated • Los Innovadores del Riego
U.S.A.: 1940 Diamond Street • San Marcos, California 92069 • TEL: (1) 760-744-5240 • FAX: (1) 760-744-7461 • www.HunterIndustries.com
Europe: Bât. A2 - Europarc de Pichaury • 1330, rue Guillibert de la Lauzière • 13856 Aix-en-Provence Cedex 3, France • TEL: (33) 4-42-37-16-90 • FAX: (33) 4-42-39-89-71
Australia: 8 The Parade West • Kent Town, South Australia 5067 • TEL: (61) 8-8363-3599 • FAX: (61) 8-8363-3687

P/N 700241 INT-231 2/04

FICHA TÉCNICA

PRODUCTO: Motorreductor RW45-3 Monofásico

REFERENCIA: 013M

CARACTERÍSTICAS TÉCNICAS

	50 Hz	60 Hz
Par mecánico (Nm)	90	70
Potencia eléctrica (Kw)	0.09	0.09
Nº fases eléctricas	2 (Monof.)	2 (Monof.)
Nº de revoluciones (rpm)	3	3.6
Voltaje (V)	220-240/380-420	220-266/380-460
Tensión (A)	0.76 / 0.44	0.67 / 0.39
L (mm)	250	250
H (mm)	27.5	27.5
Peso (Kg)	17	17

1*Max. Long. de ventana (m)	50	40
2*Max. Sup. de pantalla (m2)	1.000	800

1* Longitud aproximada, considerando v. cenitales ½ arco, en túneles de 8m.

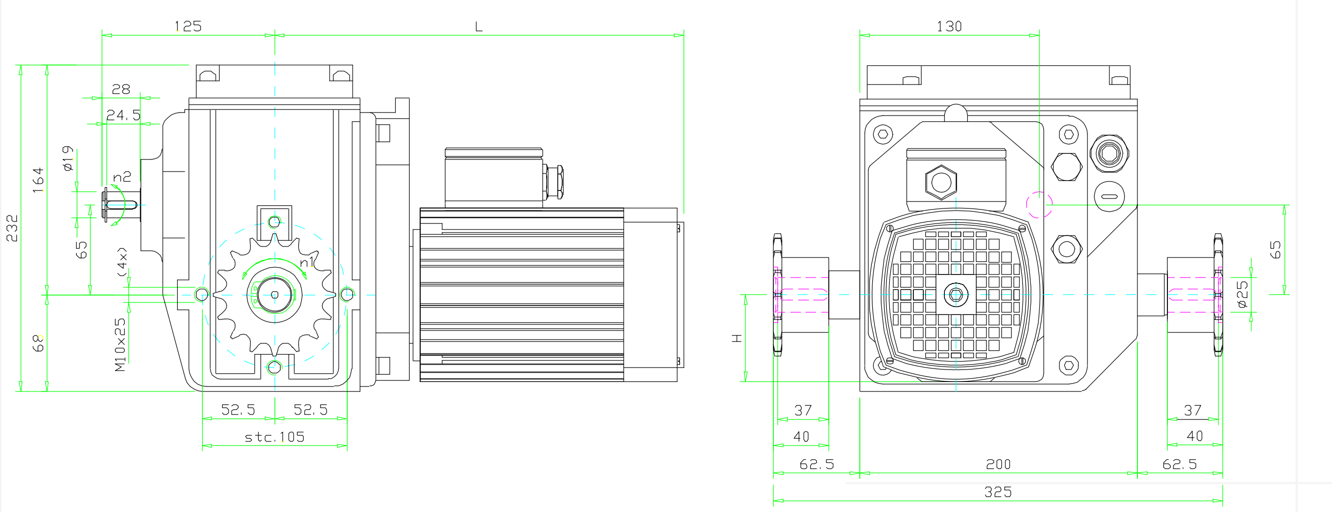
2* Considerar superficie, ancho y largo de invernadero. Consultar dptº técnico.

ACOM DESARROLLO S.L. Pol Ind Los Palomares 30591 Balsicas – Murcia (ESPAÑA) Tel.: +34 968 33 99 00 Fax: 968 585 770

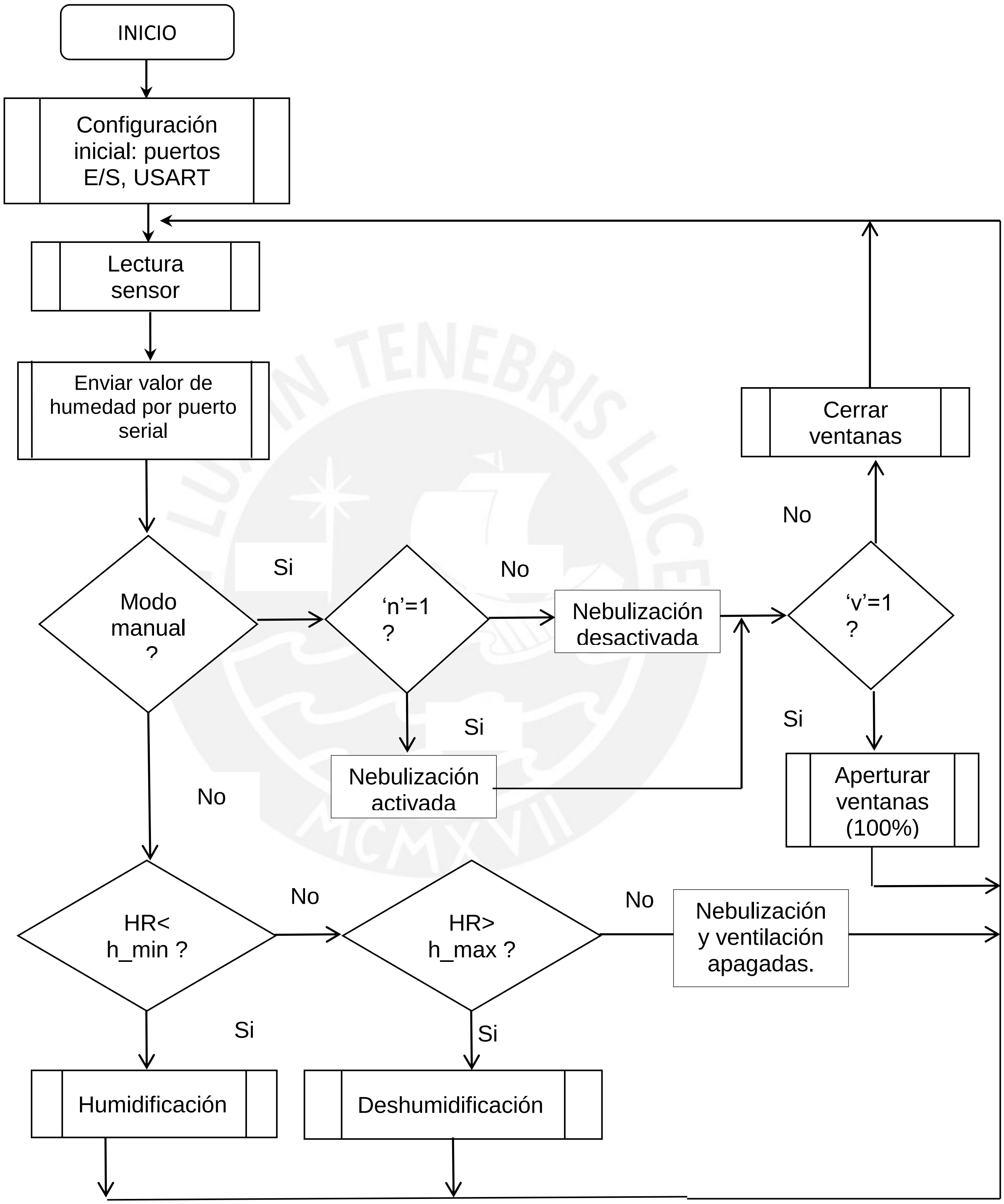
CARACTERÍSTICAS CONSTRUCTIVAS

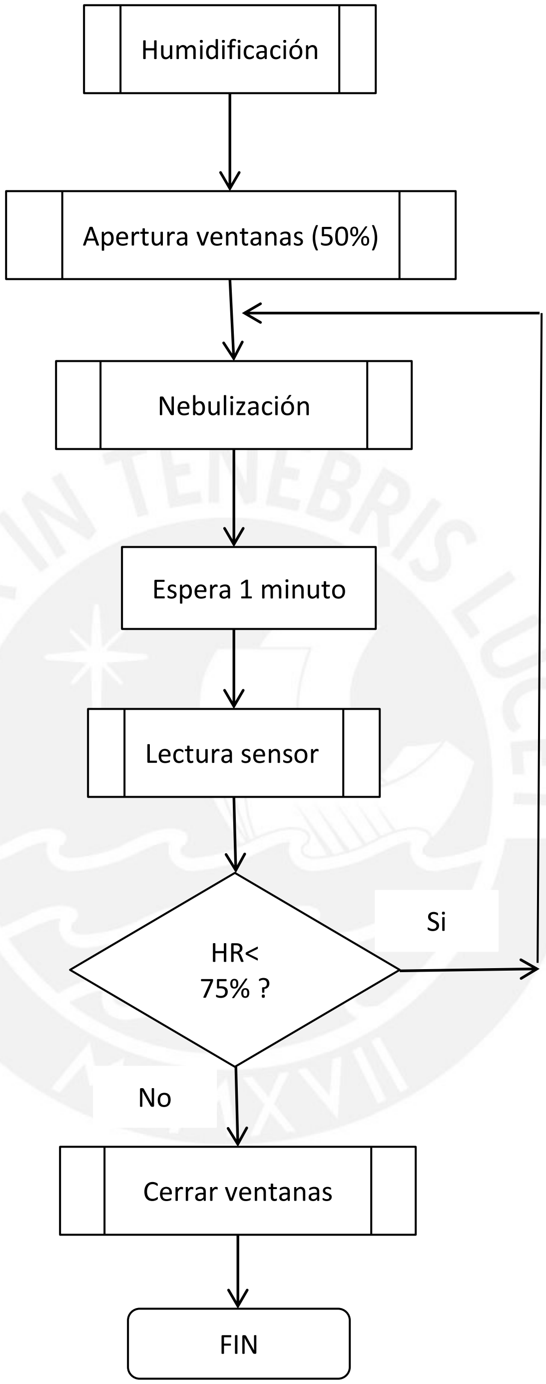
- El motorreductor especialmente diseñado para uso en invernaderos, se puede integrar también en cualquier tipo de sistema de transmisión, bien en pantallas térmicas, en ventanas o en cualquier otro sistema que requiera el uso de un motorreductor.
- La potencia eléctrica consumida es muy baja en relación al par mecánico de salida y los finales de carrera eléctricos van incorporados al motor.
- Bajo pedido dispone de cadenas de acoplamiento, placa de montaje y salida para accionamiento manual (D).

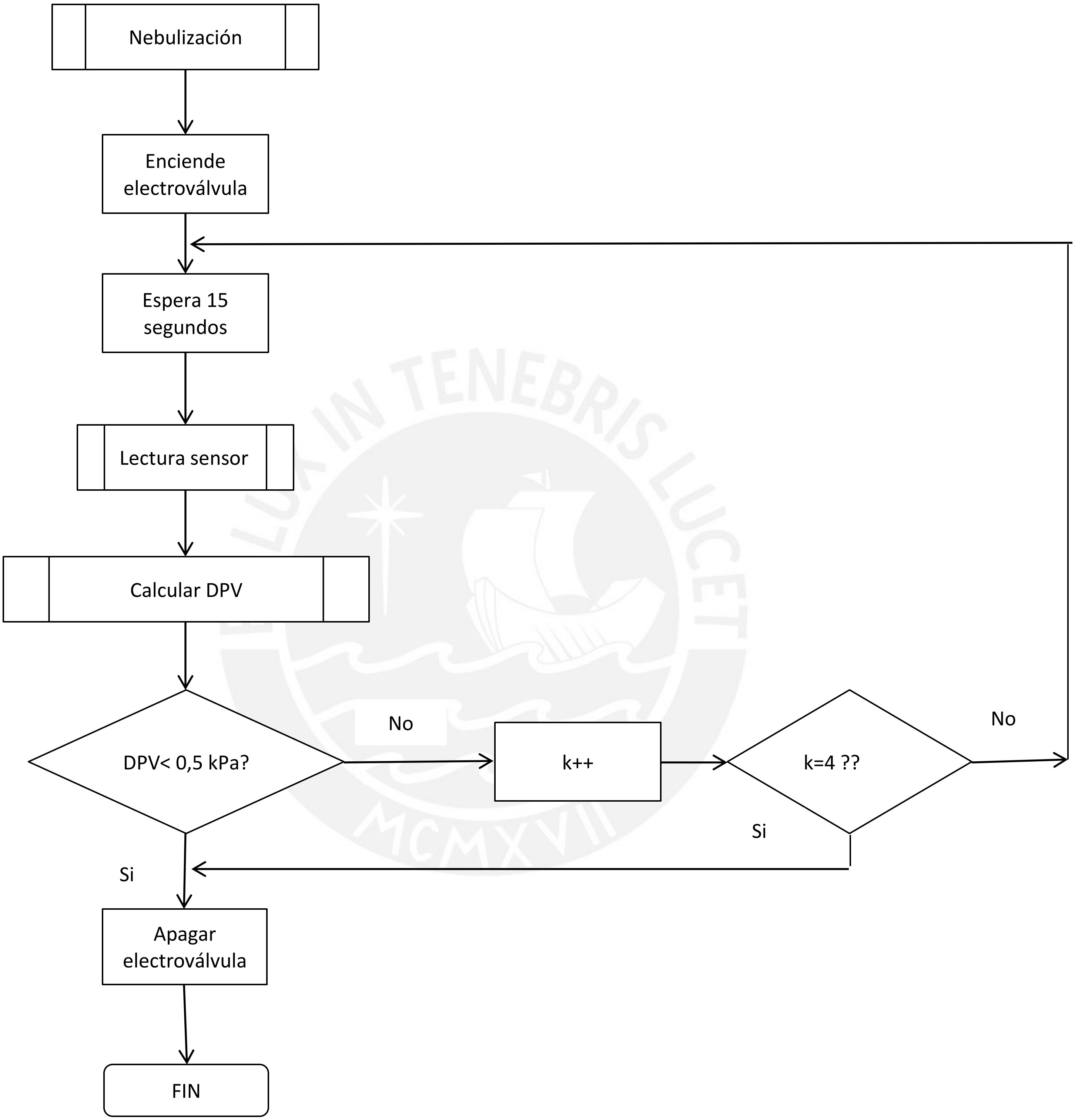
DIMENSIONES

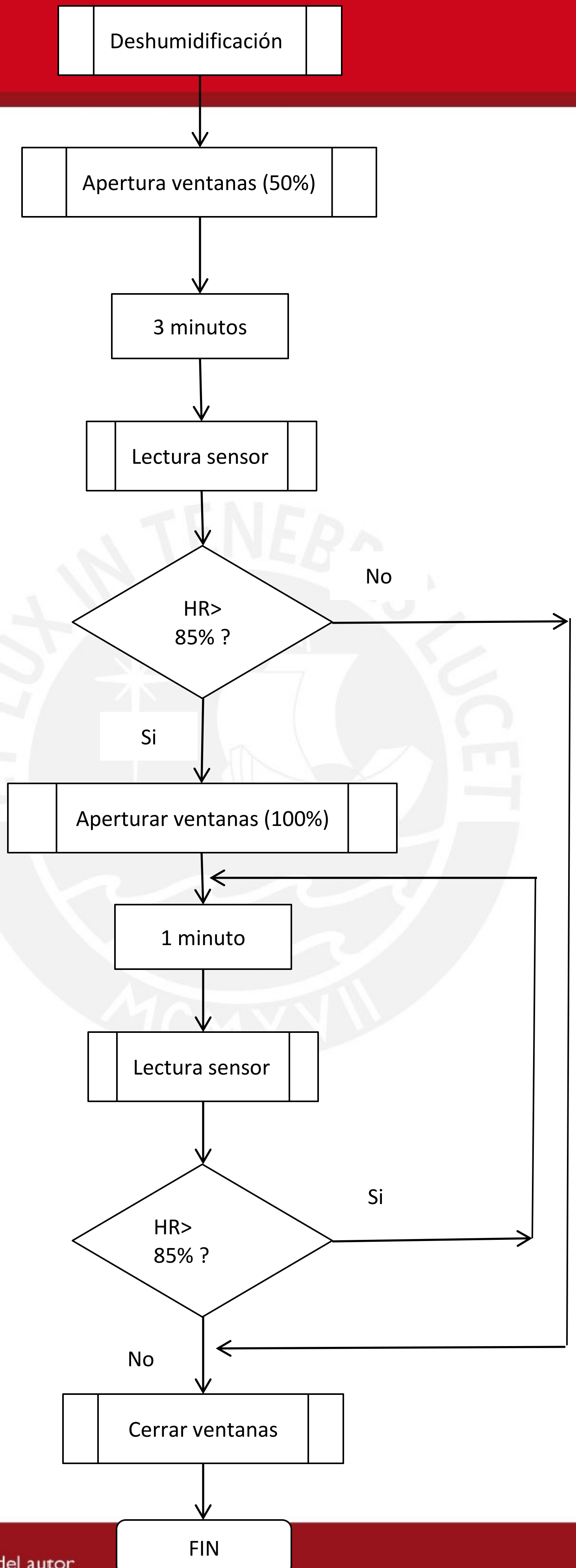


Anexo 5 - Diagrama de flujo del programa




Humidificación
Apertura ventanas (50%)
Nebulización
Espera 1 minuto
Lectura sensor
HR< 75% ?
Si
No
Cerrar ventanas
FIN


Nebulización
Enciende electroválvula
Espera 15 segundos
Lectura sensor
Calcular DPV
DPV< 0,5 kPa?
No
k++
k=4 ??
No
Si
Si
Apagar electroválvula
FIN


Deshumidificación
Apertura ventanas (50%)
3 minutos
Lectura sensor
HR> 85% ?
No
Si
Apertura ventanas (100%)
1 minuto
Lectura sensor
HR> 85% ?
Si
No
Cerrar ventanas
FIN

Anexo 6 - Código del programa

```
#include "avr/io.h"
#include <stdlib.h>
#include <avr/interrupt.h>
#include <util/delay.h>

#define BAUDRATE 9600
#define VALOR_UBRR (1000000 /BAUDRATE / 16 - 1)
#define DHT_DDR DDRD
#define DHT_PIN PIND
#define DHT_PORT PORTD
#define DHT_PINDATA PD4            //salida de datos del sensor

uint8_t datos[40];        //almacena  los 40 valores enviados por el sensor.
uint16_t datos_sensor[2];      //almacena  los valores en bytes de humedad y temperatura
uint8_t paridad;
float humedad=0;
float temperatura=0;
char str_humedad[30];          //almacena ASCII del valor de humedad para enviar por el
puerto serial
uint8_t error=0;

float DPV=0;
float h_max=85.0;      //valores referenciales iniciales definidos por el tipo de cultivo
elegidos en este proyecto
float h_min=75.0;


volatile char v_referencias[20];        //almacena los valores de referencia de humedad
enviados por la PC

volatile uint8_t modo_manual=0;       //en caso de que se trabaje en  modo manual

volatile uint8_t n=0;              //encender nebulización
volatile uint8_t v=0;              //encender ventilación

//////////////////////////////////////////////////
//Funciones para comunicación serial////////////
//////////////////////////////////////////////////

void envio_caracter(uint8_t c){
          while(!(UCSRA&(1<<UDRE))){}
```

```
        UDR=c;
        return;
}

void envio_cadena(char *s){
        while(*s) envio_caracter(*s++);
        return;
}


char recibo_caracter(){
        while(!(UCSRA&(1<<RXC))){}
        return UDR;
}


void inicio_USART(){
        UCSRB|=(1<<RXEN)|(1<<TXEN);    //inicializa USART
        UCSRC|=(1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1);        //asíncrona, 8, N,1
        UBRRH=(VALOR_UBRR>>8);
        UBRRL=VALOR_UBRR;
        return;
}

////////////////////////////////////////
//Funciones para el sensor/////////////
//////////////////////////////////////


void confg_salida (){           //configura el pin  PD4 como salida
        DHT_DDR|=1<<DHT_PINDATA;
}


void confg_entrada (){          //configura el pin PD4 como entrada
        DHT_DDR &=~(1<<DHT_PINDATA);
}


void pin_baja(){        //envía señal de inicio al sensor en baja
        DHT_PORT&=~(1<<DHT_PINDATA);
}
```

```
void pin_alta(){          // envía señal de inicio al sensor en alta
          DHT_PORT|=(1<<DHT_PINDATA);
}


void calcular_humedad(){      //
          humedad=(float)datos_sensor[0]/10;
          dtostrf(humedad, 3, 2, str_humedad);
          envio_cadena(str_humedad);          //envía valor humedad por el puerto serial a la
PC
}


void calcular_temperatura(){ //
          temperatura=(float)datos_sensor[1]/10;
}


uint8_t verificar_error(){        //verifica si se recibió los datos correctos del sensor
          uint8_t suma=0;
          suma=((datos_sensor[0]>>8)+(datos_sensor[1]>>8)+(datos_sensor[0])+(datos_sen
sor[1]));
          if(suma==paridad)error=0;
          else error=1;
          return error;
}


void lectura_sensor(){
          uint8_t k=0;
          uint8_t l=0;
          uint8_t h=0;

 if(DHT_PIN&(1<<DHT_PINDATA)){
  datos_sensor[0]=0;            //inicializa variables
  datos_sensor[1]=0;
  paridad=0;
  confg_salida();
  pin_baja();
  _delay_ms(1);         //señal de inicio
  pin_alta();
  confg_entrada();
  while((DHT_PIN&(1<<DHT_PINDATA)));            //espera respuesta del sensor
  while(!(DHT_PIN&(1<<DHT_PINDATA))){k++;}
```

```
if((k>17)&&(k<21)){
         while((DHT_PIN&(1<<DHT_PINDATA))){l++;}
         if((l>17)&&(l<21)){
           for(uint8_t i=0;i<40;i++){
                      while(!(DHT_PIN&(1<<DHT_PINDATA)))h=0;
       while((DHT_PIN&(1<<DHT_PINDATA))) h++;
                      datos[i]=h;
           }
  //verifica si se envió un bit "1" o "0"
           for(uint8_t i=0; i<40;i++){
                      if((datos[i]>3)&&(datos[i]<9)) datos[i]=0;
                      else if((datos[i]>15)&&(datos[i]<21))datos[i]=1;
           }
  //agrupa los 40 bits en bytes
    uint8_t j=15;
  //2 bytes de humedad
           for(uint8_t i=0;i<16;i++){
              if(datos[i]==1) datos_sensor[0]|=(1<<j);
                      j--;
           }
  //2 bytes de temperatura
    j=15;
    for(uint8_t i=16;i<32;i++){
              if(datos[i]==1) datos_sensor[1]|=(1<<j);
                      j--;
           }
         //byte de paridad
           j=7;
           for(uint8_t i=32;i<40;i++){
                      if(datos[i]==1) paridad|=(1<<j);
              j--;
           }

   if(!verificar_error()){
calcular_humedad();
calcular_temperatura();
}


         }
         else {error=1;return;}
 }
 else {error=1;return;}
```

```
}

}

///////////////////////////////////////////
//Funciones para activar los actuadores/////
///////////////////////////////////////////
void minuto(){
	uint8_t i=0;
	do{
	_delay_ms(1000);
	i++;
   }while(i<60);
}

void aperturaparcial_ventanas(){
	//abrir ventana 1
	if(PINC&(1<<PC2)) {   //verifica si ventana1 está cerrada
	while(!(PINC&(1<<PC1))){
	PORTB|=(1<<PB1);
	PORTB&=~(1<<PB0);}
	//se prende motor 1 hasta que ventana semiabierto
	PORTB&=~((1<<PB0)|(1<<PB1));//se vuelve a apagar motor1
	}

	//abrir ventana 2
	if(PINC&(1<<PC5)) {   //verifica si ventana2 está cerrada
	while(!(PINC&(1<<PC4))){
	PORTB|=(1<<PB3);
	PORTB&=~(1<<PB2);}
	//se prende motor 2 hasta que este semiabierto
	PORTB&=~((1<<PB2)|(1<<PB3));//se vuelve a apagar motor2
	}

	//abrir ventana 3
	if(PIND&(1<<PD3)) {   //verifica si ventana 3 está cerrada
	while(!(PIND&(1<<PD2))){
	PORTB|=(1<<PB5);
	PORTB&=~(1<<PB4);}
	//se prende motor 3 hasta que este semiabierto
	PORTB&=~((1<<PB4)|(1<<PB5));//se vuelve a apagar motor3
	}

}
```

```
void aperturacompleta_ventanas(){
	//abrir ventana 1
	if((PINC&(1<<PC2)) ||(PINC&(1<<PC1))) {   //verifica si ventana1 está cerrada o
semiabierta
	while(!(PINC&(1<<PC0))){
	PORTB|=(1<<PB1);
	PORTB&=~(1<<PB0);}
	//se prende motor 1 hasta que ventana completamente abierto
	PORTB&=~((1<<PB0)|(1<<PB1));//se vuelve a apagar motor1
	}
	//abrir ventana 2
	if((PINC&(1<<PC5)) || (PINC&(1<<PC4))) {  //verifica si ventana 2 está cerrada o
semiabierta
	while(!(PINC&(1<<PC3))){
	PORTB|=(1<<PB3);
	PORTB&=~(1<<PB2);}
	//se prende motor  2 hasta que ventana completamente abierto
	PORTB&=~((1<<PB2)|(1<<PB3));//se vuelve a apagar motor 2
	}
	//abrir ventana 3
	if(PIND&(1<<PD3)) {   //verifica si ventana 3 está cerrada
	while(!(PIND&(1<<PD2))){
	PORTB|=(1<<PB5);
	PORTB&=~(1<<PB4);}
	//se prende motor 3 hasta que ventana completamente abierto
	PORTB&=~((1<<PB4)|(1<<PB5));//se vuelve a apagar motor 3
	}
}

void cerrar_ventanas(){
	//cerrar ventana 1
	if((PINC&(1<<PC0)) ||(PINC&(1<<PC1))) {   //verifica si ventana1 está abierta o
semiabierta
	while(!(PINC&(1<<PC2)))PORTB|=(1<<PB0)|(1<<PB1);
	//se prende motor 1 hasta que ventana cerrada
	PORTB&=~((1<<PB0)|(1<<PB1));//se vuelve a apagar motor1
	}
	//cerrar ventana 2
	if((PINC&(1<<PC3)) || (PINC&(1<<PC4))) {  //verifica si ventana 2 está abierta o
semiabierta
	while(!(PINC&(1<<PC5))) PORTB|=(1<<PB2)|(1<<PB3);
	//se prende motor  2 hasta que ventana cerrada
```

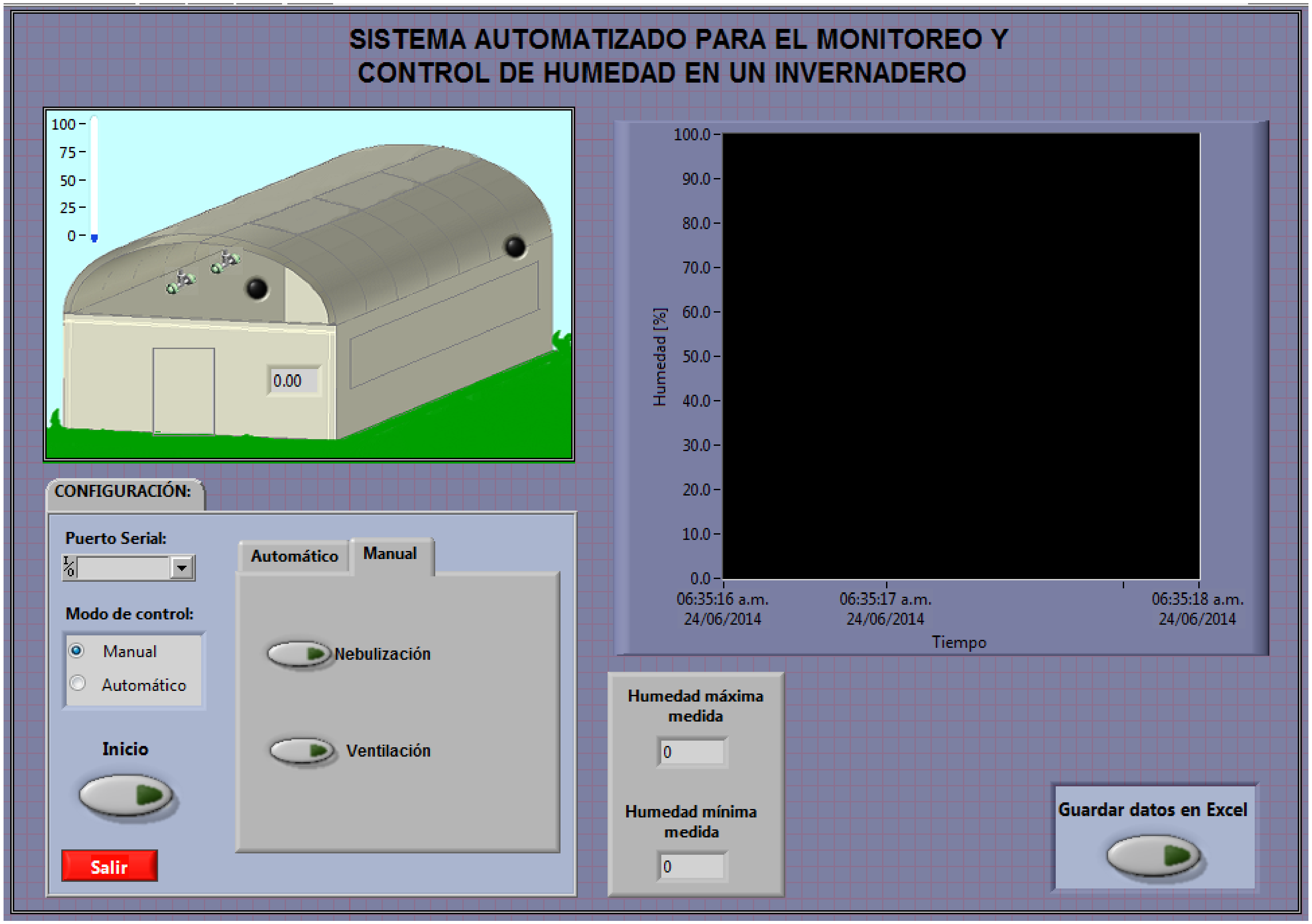
```
        PORTB&=~((1<<PB2)|(1<<PB3));//se vuelve a apagar motor 2
        }
        //cerrar ventana 3
        if(PIND&(1<<PD2)) {   //verifica si ventana 3 está abierta
        while(!(PIND&(1<<PD3))) PORTB|=(1<<PB4)|(1<<PB5);
        //se prende motor 3 hasta que ventana cerrada
        PORTB&=~((1<<PB4)|(1<<PB5));//se vuelve a apagar motor 3
        }
}


void encender_nebulizacion(){
        uint8_t k=0;
        PORTD|=(1<<PD7);// activa electroválvula
        do{
        for(uint8_t i=0;i<15;i++)_delay_ms(1000);  // esperar 15 segundos
        lectura_sensor();
        calcular_DPV();
        if(DPV<0.5) {
        PORTD&=~(1<<PD7);// apaga electroválvula
        k=4;
        }
        else k++;
        }while(k<4);
        PORTD&=~(1<<PD7);// apagar la electroválvula
}

void calcular_DPV(){

        DPV=(6.1078*exp(17.2693882*temperatura/(temperatura+237.3)))-
((6.1078*exp(17.2693882*temperatura/(temperatura+237.3)))*humedad/100);

}

void activar_humidificacion(){
        aperturaparcial_ventanas();
        do{
        encender_nebulizacion();
        for(uint8_t i=0;i<1;i++)minuto();  //  esperar 1minuto
        lectura_sensor();
        }while(humedad<h_min);
        cerrar_ventanas();
}
```

```
void activar_deshumidificacion(){
	aperturaparcial_ventanas();
	for(uint8_t i=0;i<3;i++)minuto(); //  esperar 3 minutos
	lectura_sensor();
	if(humedad>h_max){ //apertura ventanas al 100%
	aperturacompleta_ventanas();
	do{
	for(uint8_t i=0;i<1;i++)minuto(); //  esperar 1 minuto
	lectura_sensor();
	}while(humedad>h_max);
	cerrar_ventanas();
				}
	cerrar_ventanas();
}

///////////////////////////////////////////////////////////////////////////
//Funcion para la recepción de datos enviados por el usuario desde la PC
//Obtiene los valores mínimos y máximo definido por el usuario /////////
//trama recibida(mxxx.xMxxx.xT)ejm:m80.0M95.0T ,HRmín=80 y HRmáx=95 ////
///////////////////////////////////////////////////////////////////////////

void valores_referencia(){
	char v_max[10];
	char v_min[10];
	uint8_t i=0;
	uint8_t j=0;
	uint8_t a=0;
	uint8_t min=0;
	uint8_t max=0;

	do{
	  if((v_referencias[i]!='M')&&((v_referencias[i]=='m')||(min==1))){
		min=1;
		max=0;
		if(v_referencias[i]!='m'){v_min[j]=v_referencias[i];j++;}
	  }
	  else if((v_referencias[i]!='m')&&((v_referencias[i]=='M')||(max==1))){
		max=1;
		min=0;
		if((v_referencias[i]!='M')){v_max[a]=v_referencias[i];a++;}
	  }

	i++;
```

```
        }while(v_referencias[i]!='T');

        v_max[a]=0;
        v_min[j]=0;

        h_max=(atoi(v_min));
        h_max=(atoi(v_max));
        }
////////////////////////////////////////////////
//Interrupcion USART//////////////////////////////
////////////////////////////////////////////////

ISR(USART_RXC_vect){
        char valorUDR=0;
        uint8_t i=0;

        do{
        valorUDR=recibo_caracter();

                if(valorUDR=='n'){n=1;modo_manual=1;}
                else if(valorUDR=='o'){n=0;modo_manual=1;}
                else if(valorUDR=='v'){v=1;modo_manual=1;}
                else if(valorUDR=='p'){v=0;modo_manual=1;}

                else {
           modo_manual=0;
                v_referencias[i]=valorUDR;
                i++;
                }

        }while(valorUDR!='T');
}




//////////////////////////////////////////////////////////////////////////
/*        PROGRAMA PRINCIPAL        */

///////////////////////////////////////////////////////////////////////
void main(){
//salidas para los motores: PB0 y PB1 ->Motor 1  PB2 y PB3 ->Motor 2  PB4 y PB5 -
>Motor 3
        DDRB|=(1<<PB0)|(1<<PB1)|(1<<PB2)|(1<<PB3)|(1<<PB4)|(1<<PB5);
```

```
        DDRD|=(1<<PD7); //salidas para la electroválvula: PD7->Electrovalvula
        DDRC&=~((1<<PC0)|(1<<PC1 )|(1<<PC2)|(1<<PC3 )|(1<<PC4)|(1<<PC5));
//entradas de 6 interruptores de final de carrera
        DDRD&=~((1<<PD2)|(1<<PD3 ));  //entradas de 2 interruptores de final de carrera

        inicio_USART();
        UCSRB|=(1<<RXCIE); //activa la interrupcion USART
        sei();

        do{
   lectura_sensor();//se controla actuadores según valor de humedad y modo de
operación

        //Sistema en modo automático//

        if      (modo_manual==0){
                        if(humedad<h_min){ //enciende humidificacion

                        activar_humidificacion();
                        }
                        else if (humedad>h_max){//enciende deshumidificacion
                        activar_deshumidificacion();
                        }
                        else {//apagar ambos sistemas
                        PORTD&=~(1<<PD7);// apagar la electroválvula
                        cerrar_ventanas();
                        }
                }
        else {

        //Sistema en modo manual//

                    if(n==1) PORTD|=(1<<PD7);// se prende la electroválvula
                    else if(n==0) PORTD&=~(1<<PD7);// apagar la electroválvula
                    if(v==1)aperturacompleta_ventanas();
                    else if(v==0) cerrar_ventanas();

        }

        }while(1);
}
```

Anexo 7 - Panel de control y diagrama de Bloques de la interfaz de usuario en LabVIEW.



Panel de control de la interfaz en LabView

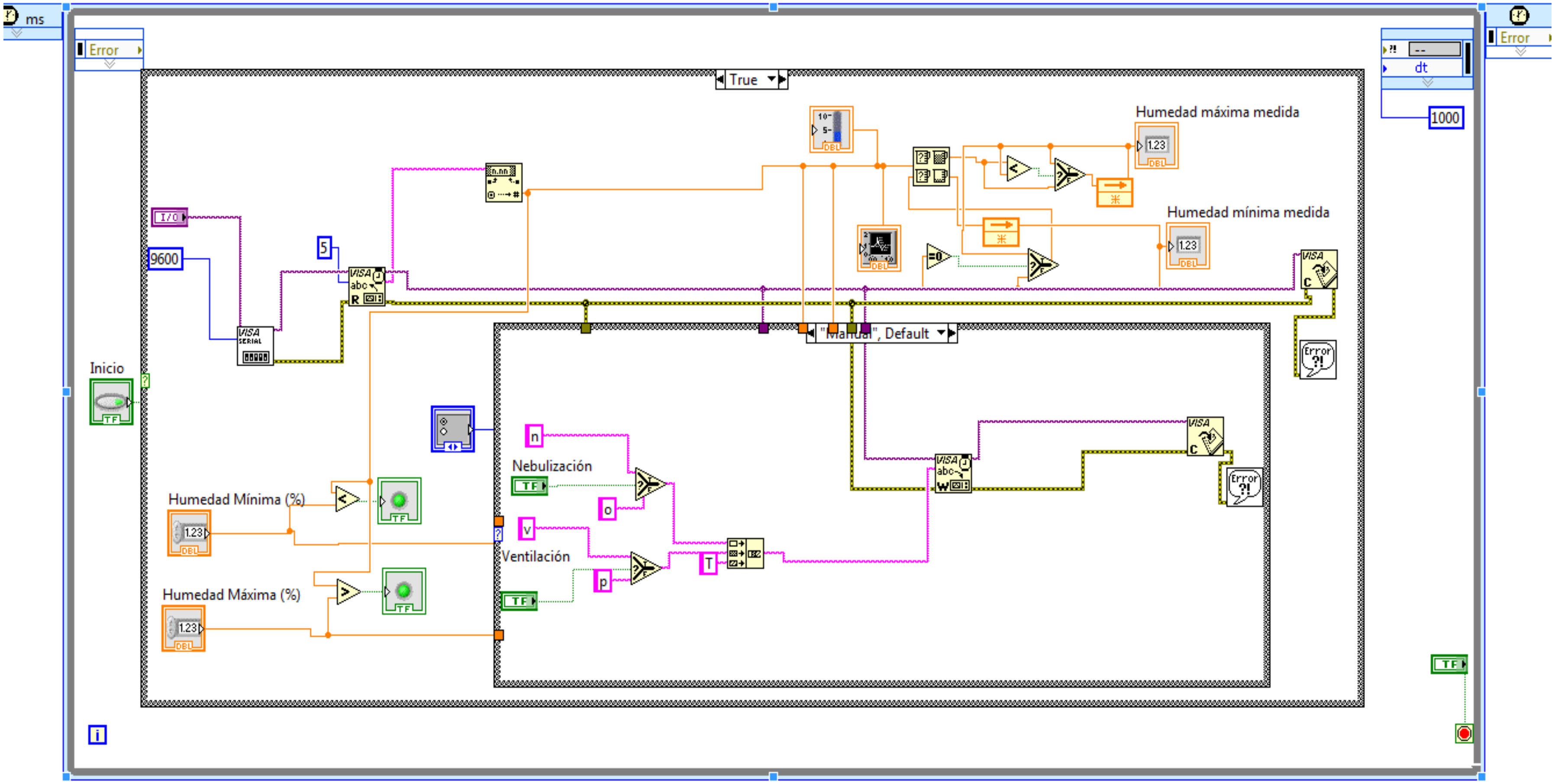


Diagrama de bloques - Modo manual

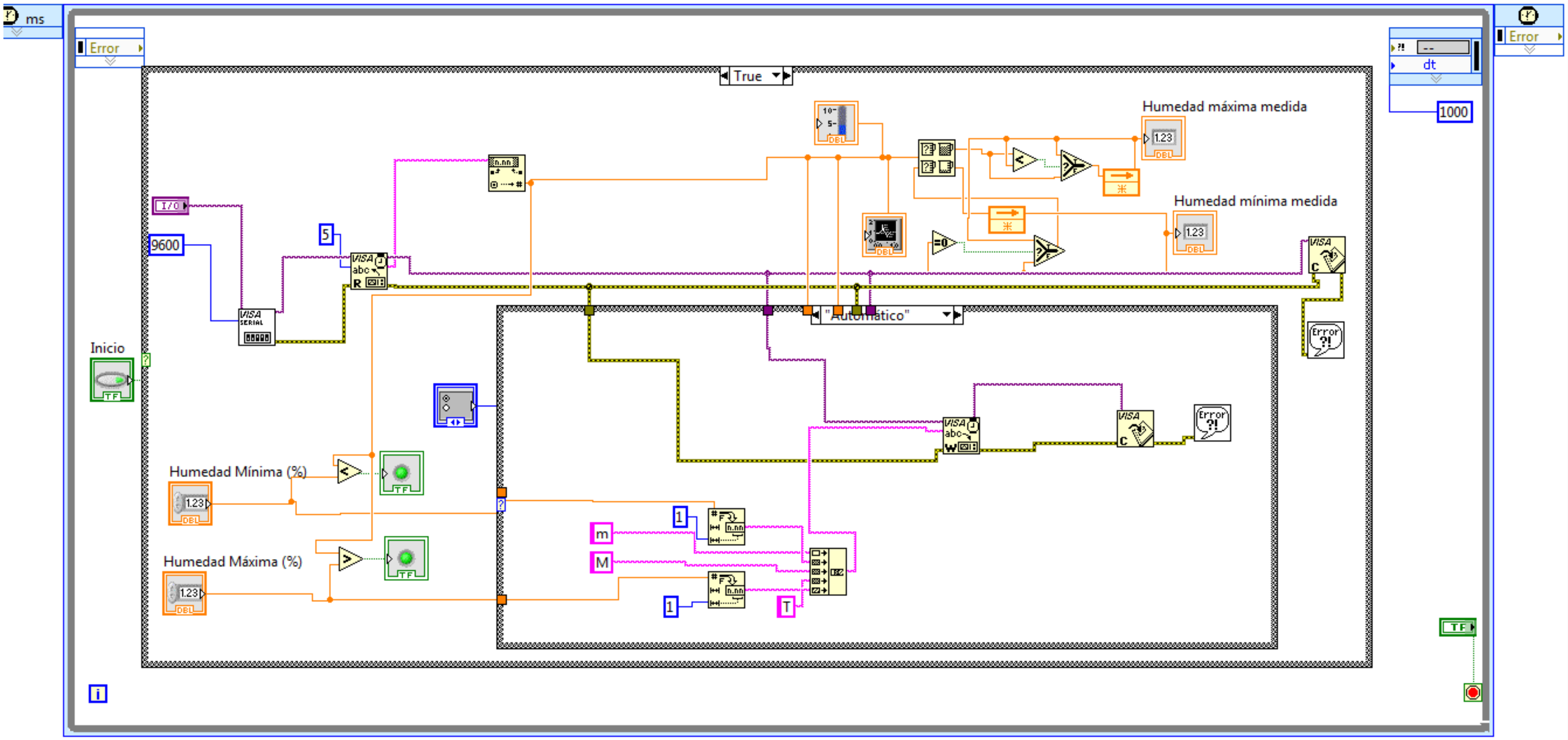
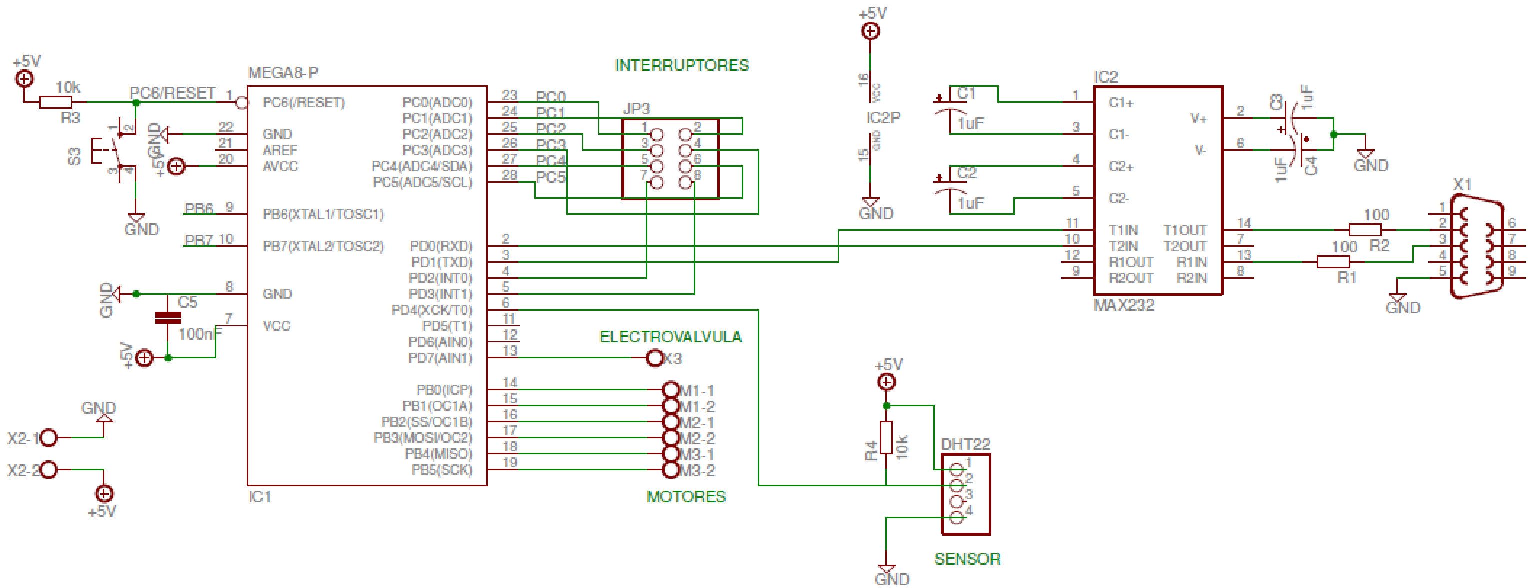


Diagrama de bloques - Modo automático

Anexo 8 - Diagramas esquemático y de pistas de los circuitos.



Esquemático del circuito del microcontrolador con el MAX232

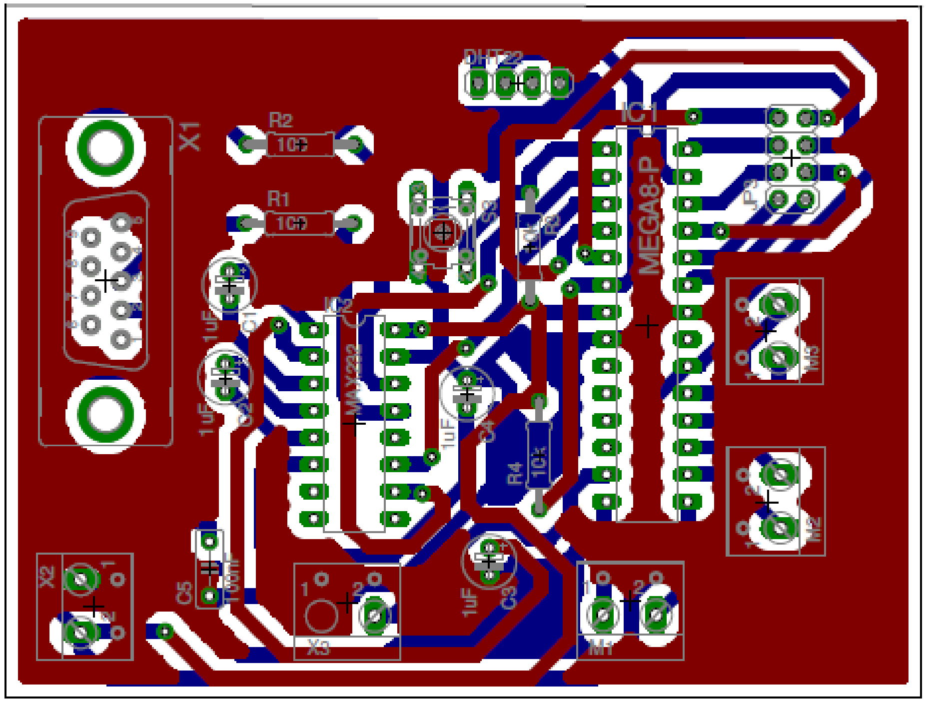
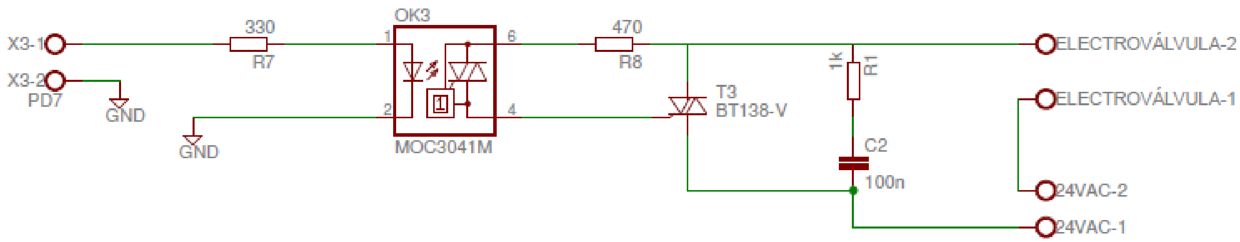


Diagrama de pistas del circuito del microcontrolador con el MAX232



Esquemático del circuito de potencia de la electroválvula

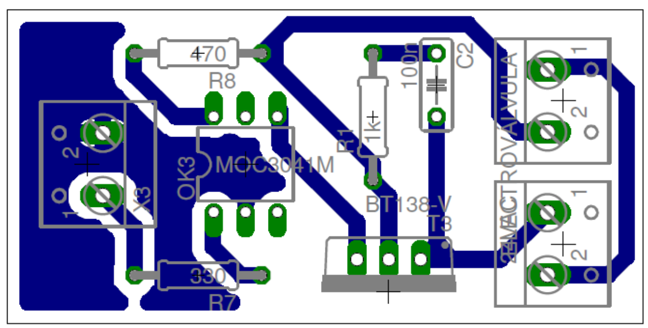
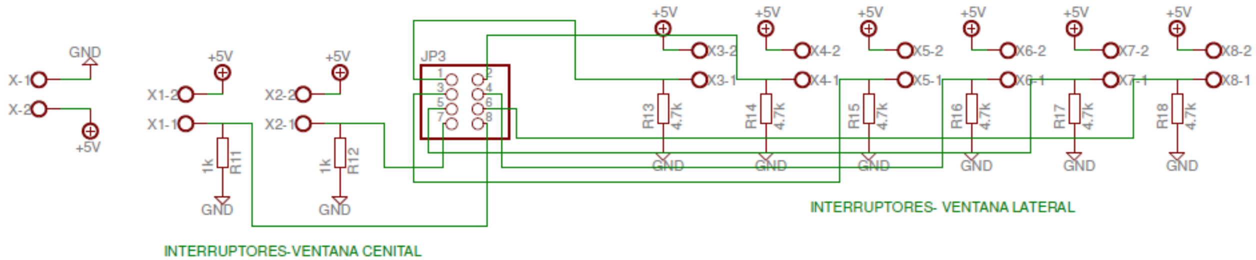


Diagrama de pistas del circuito de potencia dela electroválvula



Esquemático del circuito de los interruptores finales de carrera

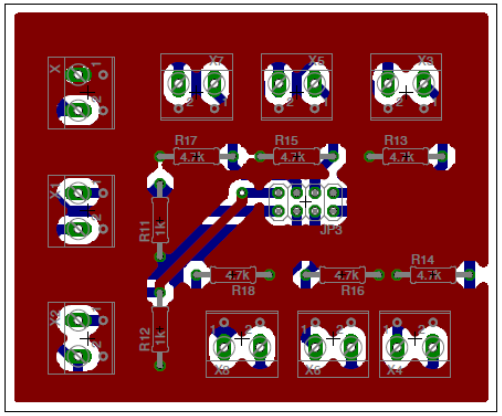
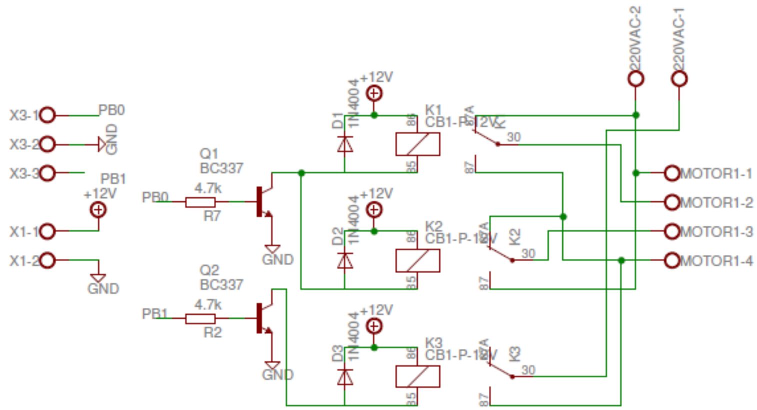


Diagrama de pistas del circuito de los interruptores finales de carrera



Esquemático del circuito de potencia del motor 1

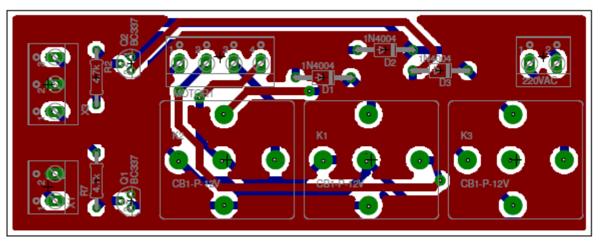
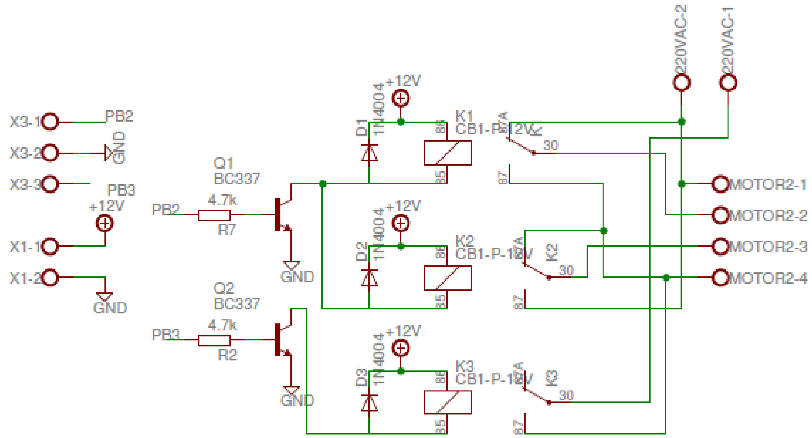


Diagrama de pistas del circuito de potencia del motor 1



Esquemático del circuito de potencia del motor 2

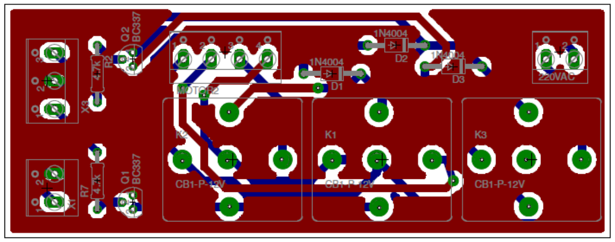
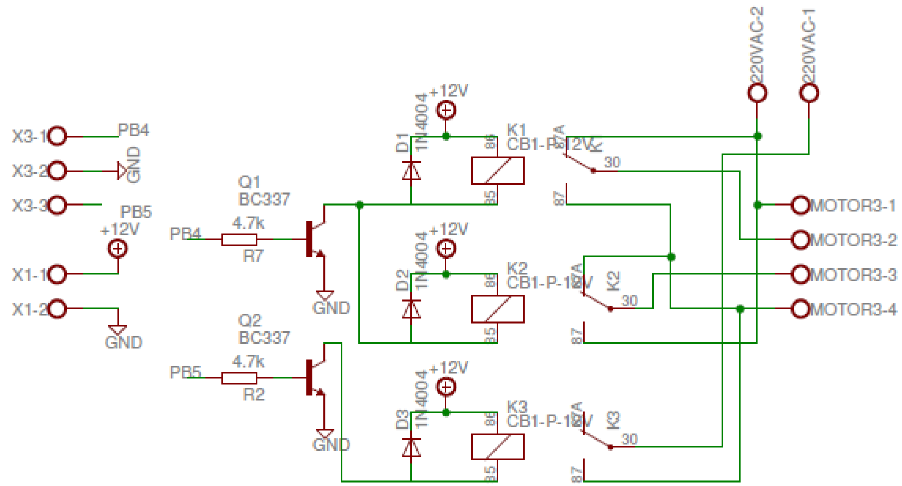


Diagrama de pistas del circuito de potencia del motor 2



Esquemático del circuito de potencia del motor 3

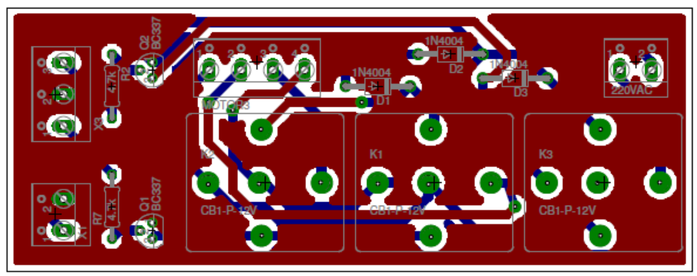
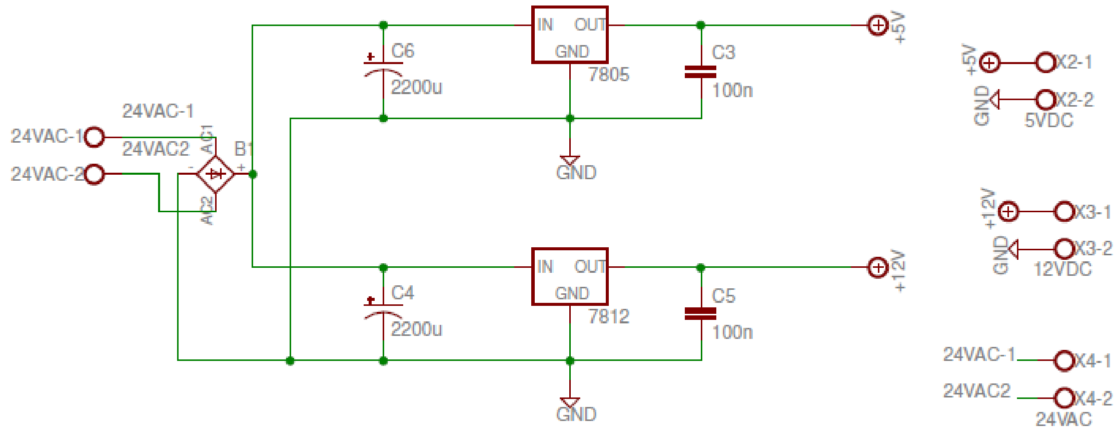


Diagrama de pistas del circuito de potencia del motor 3



Esquemático de la fuente de alimentación de 5V y 12 V

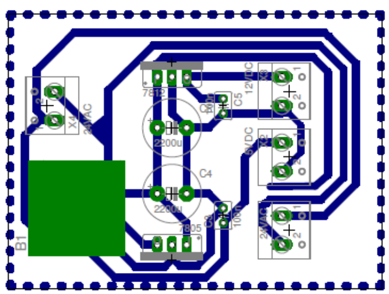


Diagrama de pistas de la fuente de alimentación de 5V y 12 V