LAMPIRAN - LAMPIRAN

tures

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

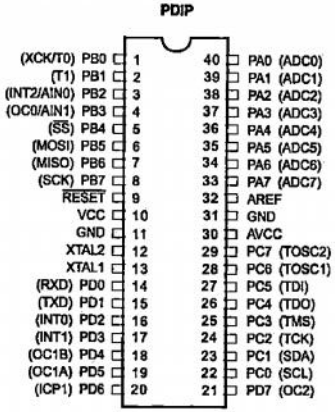
8-bit AVR® Microcontroller with 16K Bytes In-System Programmable Flash

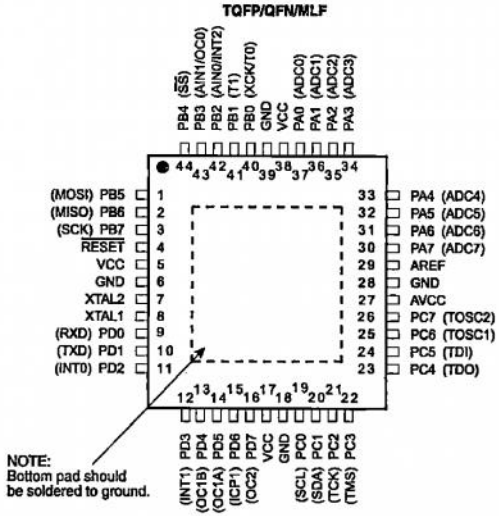
ATmega16
ATmega16L

Summary

Rev. 2466TS–AVR–07/10

Figure 1. Pinout ATmega16





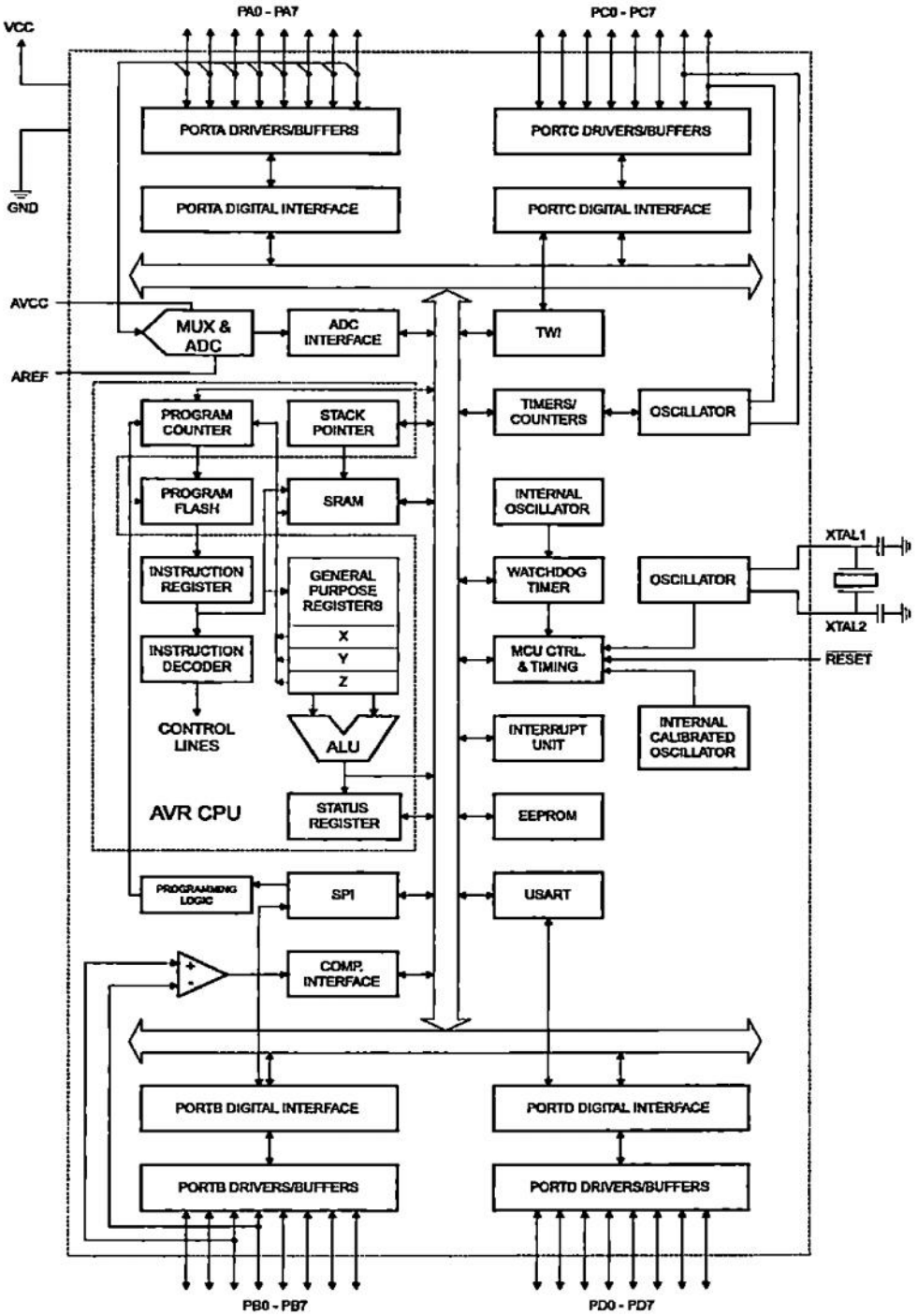
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

rview

k Diagram

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

Figure 2. Block Diagram



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

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

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

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

escriptions

Digital supply voltage.

Ground.

(PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

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

(PB7..PB0)

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

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

(PC7..PC0)

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

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

(PD7..PD0)

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

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

$\overline{T}$

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.

1

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

2

Output from the inverting Oscillator amplifier.

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

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

ources

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

a 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.

ster Summary

ss	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
F)	SREG	I	T	H	S	V	N	Z	C	9
E)	SPH	–	–	–	–	–	SP10	SP9	SP8	12
D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
C)	OCR0	Timer/Counter0 Output Compare Register								85
B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	48, 69
A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	70
9)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	85, 115, 133
8)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	86, 115, 133
7)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	250
6)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	180
5)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	32, 68
4)	MCUCSR	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	41, 69, 231
3)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	83
2)	TCNT0	Timer/Counter0 (8 Bits)								85
1)[1]	OSCCAL	Oscillator Calibration Register								30
	OCDR	On-Chip Debug Register								227
0)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	57,88,134,201,221
F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	110
E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	113
D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								114
C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								114
B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								114
A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								114
9)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								114
8)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								114
7)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								114
6)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								114
5)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	128
4)	TCNT2	Timer/Counter2 (8 Bits)								130
3)	OCR2	Timer/Counter2 Output Compare Register								130
2)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	131
1)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	43
0)[2]	UBRRH	URSEL	–	–	–	UBRR[11:8]				167
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	166
F)	EEARH	–	–	–	–	–	–	–	EEAR8	19
E)	EEARL	EEPROM Address Register Low Byte								19
D)	EEDR	EEPROM Data Register								19
C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	19
B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	68
A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	68
9)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	68
8)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	68
7)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	68
6)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	68
5)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	67
4)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	67
3)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	67
2)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	67
1)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	67
0)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	67
F)	SPDR	SPI Data Register								142
E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	142
D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	140
C)	UDR	USART I/O Data Register								163
B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	164
A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	165
9)	UBRRL	USART Baud Rate Register Low Byte								167
8)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	202
7)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	217
6)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	219
5)	ADCH	ADC Data Register High Byte								220
4)	ADCL	ADC Data Register Low Byte								220
3)	TWDR	Two-wire Serial Interface Data Register								182
2)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	182

ress	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
($21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	181
($20)	TWBR	Two-wire Serial Interface Bit Rate Register								180

1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debugger specific documentation for details on how to use the OCDR Register.
2. Refer to the USART description for details on how to access UBRRH and UCSRC.
3. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
4. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

ruction Set Summary

onics	Operands	Description	Operation	Flags	#Clocks
ETIC AND LOGIC INSTRUCTIONS					
	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
	Rd	One's Complement	Rd ← $FF – Rd	Z,C,N,V	1
	Rd	Two's Complement	Rd ← $00 – Rd	Z,C,N,V,H	1
	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
	Rd	Set Register	Rd ← $FF	None	1
	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
U	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
H INSTRUCTIONS					
	k	Relative Jump	PC ← PC + k + 1	None	2
		Indirect Jump to (Z)	PC ← Z	None	2
	k	Direct Jump	PC ← k	None	3
	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
		Indirect Call to (Z)	PC ← Z	None	3
	k	Direct Subroutine Call	PC ← k	None	4
		Subroutine Return	PC ← STACK	None	4
		Interrupt Return	PC ← STACK	I	4
	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1/2/3
	Rd,Rr	Compare	Rd – Rr	Z, N,V,C,H	1
	Rd,Rr	Compare with Carry	Rd – Rr – C	Z, N,V,C,H	1
	Rd,K	Compare Register with Immediate	Rd – K	Z, N,V,C,H	1
	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1/2/3
	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1/2/3
	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1/2/3
	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1/2/3
	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1/2
	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1/2
	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1/2
	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1/2
	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1/2
	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1/2
	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1/2
	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1/2
	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1/2
	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1/2
	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1/2
	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1/2
	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1/2
	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1/2
	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1/2
	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1/2
	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1/2

onics	Operands	Description	Operation	Flags	#Clocks
	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
RANSFER INSTRUCTIONS					
	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
	Rd, K	Load Immediate	Rd ← K	None	1
	Rd, X	Load Indirect	Rd ← (X)	None	2
	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
	Rd, Y	Load Indirect	Rd ← (Y)	None	2
	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
	Rd, Z	Load Indirect	Rd ← (Z)	None	2
	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
	X, Rr	Store Indirect	(X) ← Rr	None	2
	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
	Y, Rr	Store Indirect	(Y) ← Rr	None	2
	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
	Z, Rr	Store Indirect	(Z) ← Rr	None	2
	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
		Load Program Memory	R0 ← (Z)	None	3
	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
		Store Program Memory	(Z) ← R1:R0	None	-
	Rd, P	In Port	Rd ← P	None	1
	P, Rr	Out Port	P ← Rr	None	1
	Rr	Push Register on Stack	STACK ← Rr	None	2
	Rd	Pop Register from Stack	Rd ← STACK	None	2
D BIT-TEST INSTRUCTIONS					
	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
	s	Flag Set	SREG(s) ← 1	SREG(s)	1
	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
		Set Carry	C ← 1	C	1
		Clear Carry	C ← 0	C	1
		Set Negative Flag	N ← 1	N	1
		Clear Negative Flag	N ← 0	N	1
		Set Zero Flag	Z ← 1	Z	1
		Clear Zero Flag	Z ← 0	Z	1
		Global Interrupt Enable	I ← 1	I	1
		Global Interrupt Disable	I ← 0	I	1
		Set Signed Test Flag	S ← 1	S	1
		Clear Signed Test Flag	S ← 0	S	1
		Set Twos Complement Overflow.	V ← 1	V	1
		Clear Twos Complement Overflow	V ← 0	V	1
		Set T in SREG	T ← 1	T	1
		Clear T in SREG	T ← 0	T	1
		Set Half Carry Flag in SREG	H ← 1	H	1

nonics	Operands	Description	Operation	Flags	#Clocks
		Clear Half Carry Flag in SREG	H ← 0	H	1
CONTROL INSTRUCTIONS					
		No Operation		None	1
P		Sleep	(see specific descr. for Sleep function)	None	1
		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
K		Break	For On-Chip Debug Only	None	N/A

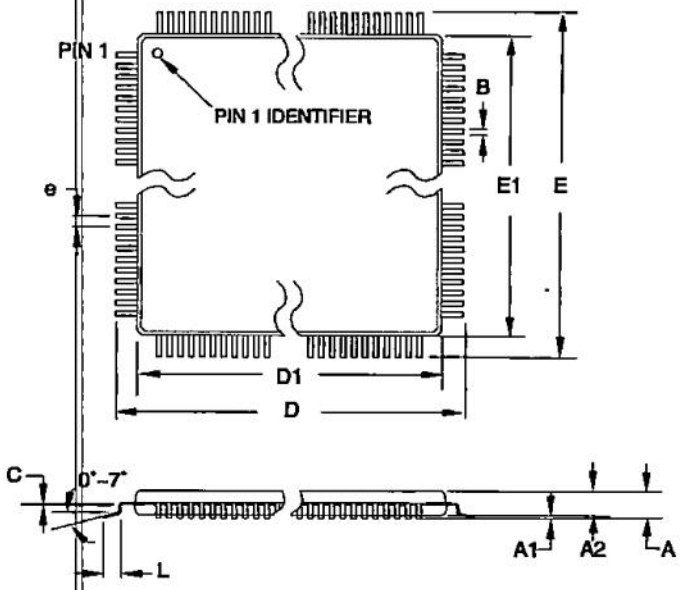
ering Information

peed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7V - 5.5V	ATmega16L-8AU[1] ATmega16L-8PU[1] ATmega16L-8MU[1]	44A 40P6 44M1	Industrial (-40°C to 85°C)
16	4.5V - 5.5V	ATmega16-16AU[1] ATmega16-16PU[1] ATmega16-16MU[1]	44A 40P6 44M1	Industrial (-40°C to 85°C)

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

	Package Type
	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
	44-pad, 7 × 7 × 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

aging Information



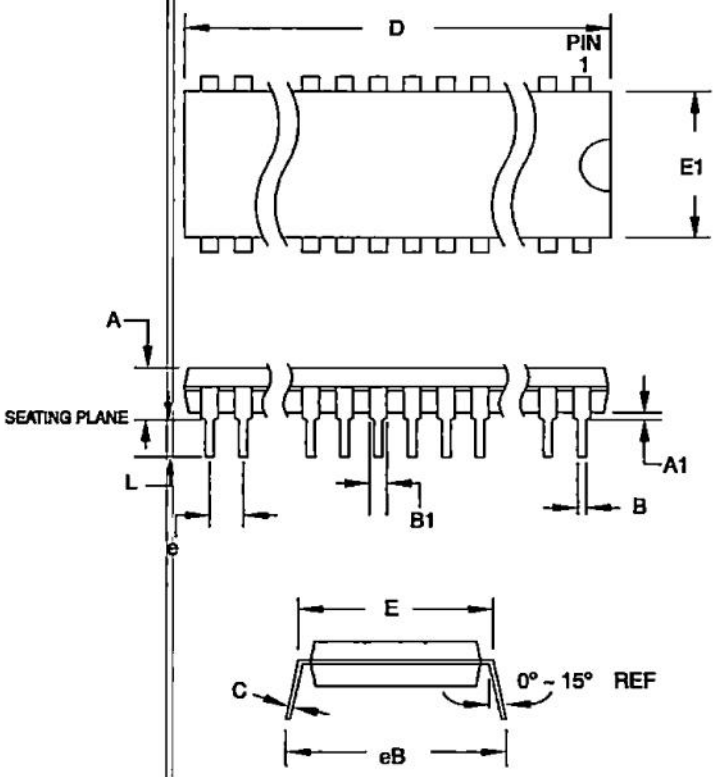
COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	44A	B



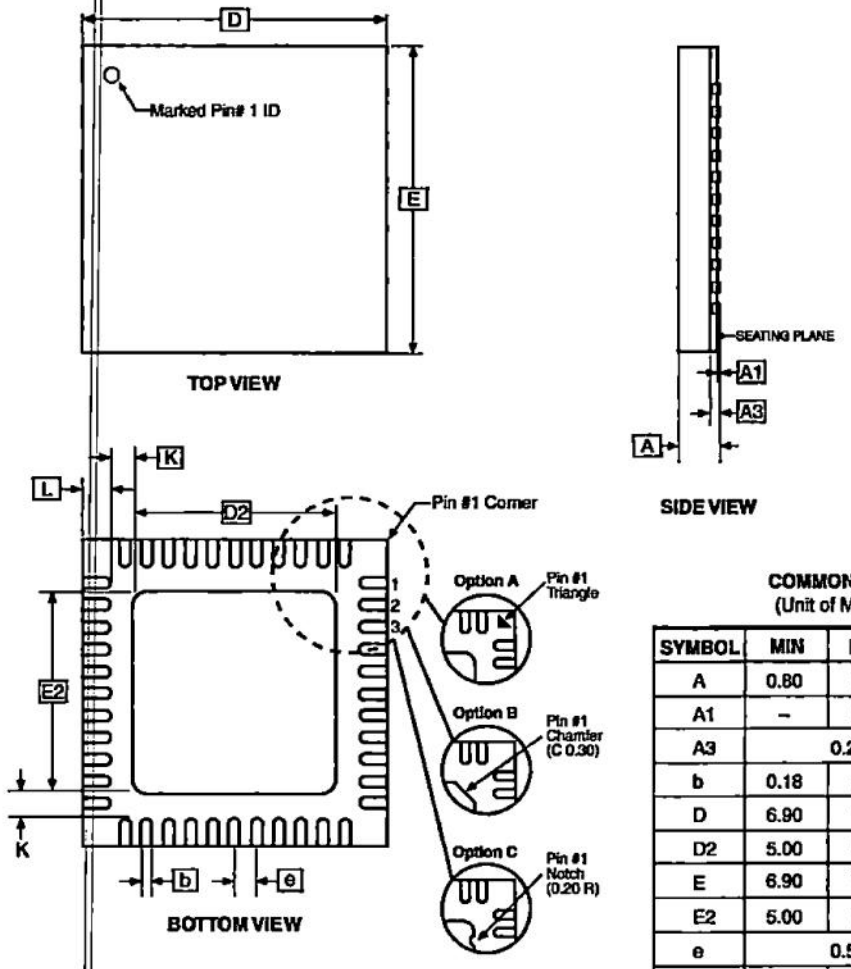
COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	40P6, 40-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	40P6	B



Note: JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-3.

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A3		0.20 REF		
b	0.18	0.23	0.30	
D	6.90	7.00	7.10	
D2	5.00	5.20	5.40	
E	6.90	7.00	7.10	
E2	5.00	5.20	5.40	
e		0.50 BSC		
L	0.59	0.64	0.69	
K	0.20	0.26	0.41	

9/26/08

Package Drawing Contact: packagedrawings@atmel.com	TITLE	GPC	DRAWING NO.	REV.
	44M1, 44-pad, 7 x 7 x 1.0 mm Body, Lead Pitch 0.50 mm, 5.20 mm Exposed Pad, Thermally Enhanced Plastic Very Thin Quad Flat No Lead Package (VQFN)	ZWS	44M1	H

ıta

ega16(L) Rev.

The revision letter in this section refers to the revision of the ATmega16 device.

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

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

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

   Problem Fix/Workaround

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

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

   The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

   Problem Fix / Workaround

   Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

   - If ATmega16 is the only device in the scan chain, the problem is not visible.
   - Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
   - If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ega16(L) Rev.

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

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

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

Problem Fix/Workaround

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

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

The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

Problem Fix / Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

Problem Fix / Workaround

Always use OUT or SBI to set EERE in EECR.

ega16(L) Rev.

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

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

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

Problem Fix/Workaround

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

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

The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

Problem Fix / Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

   - If ATmega16 is the only device in the scan chain, the problem is not visible.
   - Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
   - If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ega16(L) Rev.

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

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

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

   Problem Fix/Workaround

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

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

   The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

   Problem Fix / Workaround

   Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

Problem Fix / Workaround

Always use OUT or SBI to set EERE in EECR.

ega16(L) Rev.

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

1. First Analog Comparator conversion may be delayed

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

Problem Fix/Workaround

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

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

The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

Problem Fix / Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. IDCODE masks data from TDI Input

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ega16(L) Rev.

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**

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

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

   Problem Fix/Workaround

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

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

   The interrupt will be lost if a timer register that is synchronized to the asynchronous timer clock is written when the asynchronous Timer/Counter register(TCNTx) is 0x00.

   Problem Fix / Workaround

   Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register(TCCRx), asynchronous Timer Counter Register(TCNTx), or asynchronous Output Compare Register(OCRx).

3. **IDCODE masks data from TDI input**

   The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

   Problem Fix / Workaround

   - If ATmega16 is the only device in the scan chain, the problem is not visible.
   - Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
   - If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

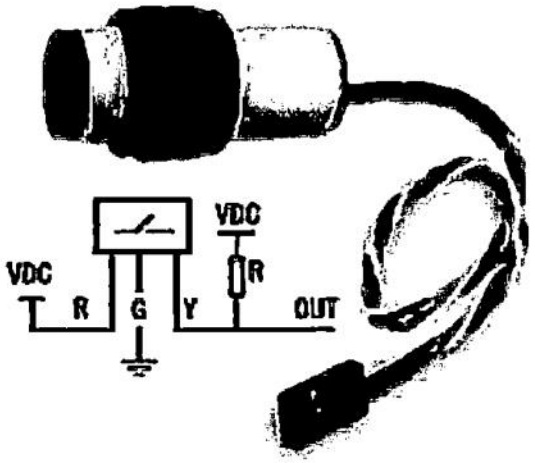
   Always use OUT or SBI to set EERE in EECR.

E18-D80NK-N
Adjustable Infrared Sensor Switch Manual

61mcu.Com
北京亿学通电子

Introduction

This is an infrared distance switch. It has an adjustable detection range, 3cm - 80cm. It is small, easy to use/assemble, inexpensive. Useful for robot, interactive media, industrial assembly line, etc.



Specification

Model NO: E18-D80NK-N

Sensing range: 3-80cm adjustable

Sensing object: Translucency, opaque

Supply voltage: DC5V

Load current: 100mA

Output operation: Normally open(O)

Output: DC three-wire system(NPN)

Diameter: 18mm, Length: 45mm

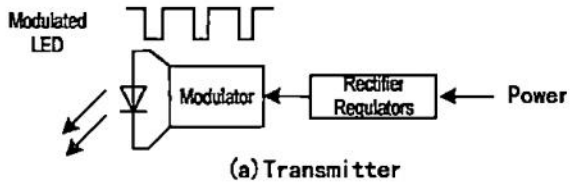
Appearance: Threaded cylindrical

Material: Plastic

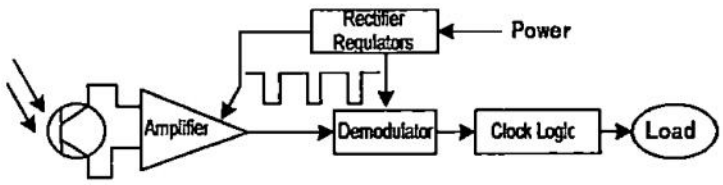
Guard mode: Reverse polarity protection

Ambient temperature: -25-70℃

Red: +5V; Yellow:Signal;Green:GND



(a) Transmitter



(a) Receiver