Features

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

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

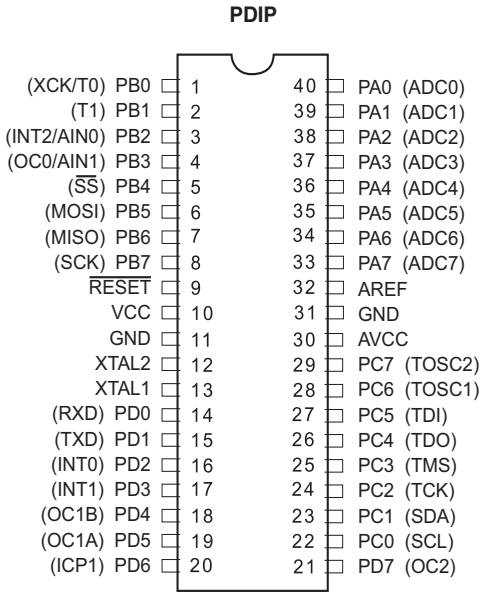
ATmega16
ATmega16L

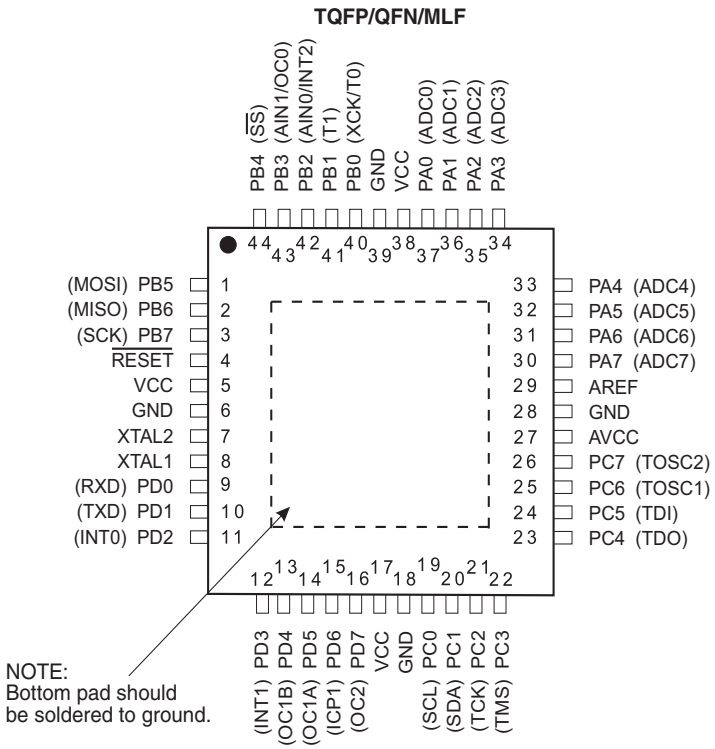
Summary

Rev. 2466TS–AVR–07/10

Pin Configurations

Figure 1. Pinout ATmega16





Disclaimer

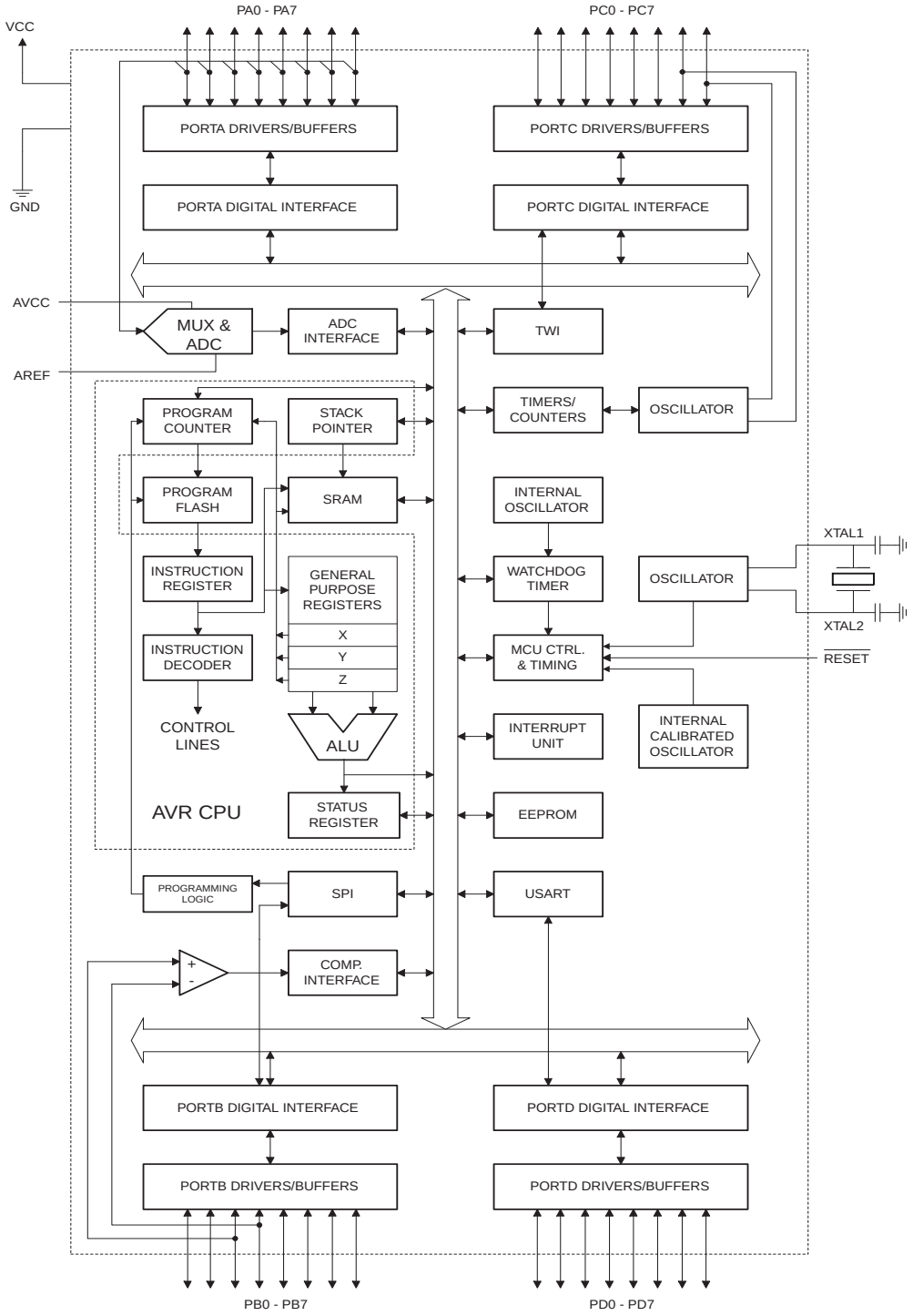
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

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

Block Diagram

Figure 2. Block Diagram



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

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

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

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

Pin Descriptions

VCC Digital supply voltage.

GND Ground.

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

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

Port B (PB7..PB0)

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

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

Port C (PC7..PC0)

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

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

Port D (PD7..PD0)

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

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

$\overline{\text{RESET}}$

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

XTAL1

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

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

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

AREF

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

Resources

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

Data Retention

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

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	9
$3E ($5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	12
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
$3C ($5C)	OCR0	Timer/Counter0 Output Compare Register								85
$3B ($5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	48, 69
$3A ($5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	70
$39 ($59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	85, 115, 133
$38 ($58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	86, 115, 133
$37 ($57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	250
$36 ($56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	180
$35 ($55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	32, 68
$34 ($54)	MCUCSR	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	41, 69, 231
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	83
$32 ($52)	TCNT0	Timer/Counter0 (8 Bits)								85
$31[1] ($51)[1]	OSCCAL	Oscillator Calibration Register								30
	OCDR	On-Chip Debug Register								227
$30 ($50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	57,88,134,201,221
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	110
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	113
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								114
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								114
$2B ($4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								114
$2A ($4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								114
$29 ($49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								114
$28 ($48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								114
$27 ($47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								114
$26 ($46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								114
$25 ($45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	128
$24 ($44)	TCNT2	Timer/Counter2 (8 Bits)								130
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								130
$22 ($42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	131
$21 ($41)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	43
$20[2] ($40)[2]	UBRRH	URSEL	–	–	–	UBRR[11:8]				167
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	166
$1F ($3F)	EEARH	–	–	–	–	–	–	–	EEAR8	19
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								19
$1D ($3D)	EEDR	EEPROM Data Register								19
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	19
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	66
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	66
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	66
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	66
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	66
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	66
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	67
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	67
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	67
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	67
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	67
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	67
$0F ($2F)	SPDR	SPI Data Register								142
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	142
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	140
$0C ($2C)	UDR	USART I/O Data Register								163
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	164
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	165
$09 ($29)	UBRRL	USART Baud Rate Register Low Byte								167
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	202
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	217
$06 ($26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	219
$05 ($25)	ADCH	ADC Data Register High Byte								220
$04 ($24)	ADCL	ADC Data Register Low Byte								220
$03 ($23)	TWDR	Two-wire Serial Interface Data Register								182
$02 ($22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	182

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$01 ($21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	181
$00 ($20)	TWBR	Two-wire Serial Interface Bit Rate Register								180

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

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF – Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 – Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd – Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2

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

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

Ordering Information

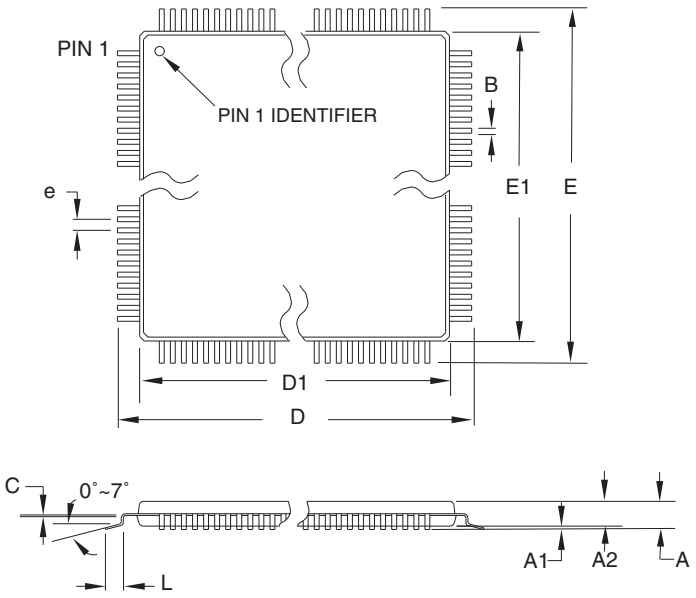
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7V - 5.5V	ATmega16L-8AU[1] ATmega16L-8PU[1] ATmega16L-8MU[1]	44A 40P6 44M1	Industrial (-40°C to 85°C)
16	4.5V - 5.5V	ATmega16-16AU[1] ATmega16-16PU[1] ATmega16-16MU[1]	44A 40P6 44M1	Industrial (-40°C to 85°C)

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

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-pin, 0.600” Wide, Plastic Dual Inline Package (PDIP)
44M1	44-pad, 7 × 7 × 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

Packaging Information

44A



COMMON DIMENSIONS
(Unit of Measure = mm)

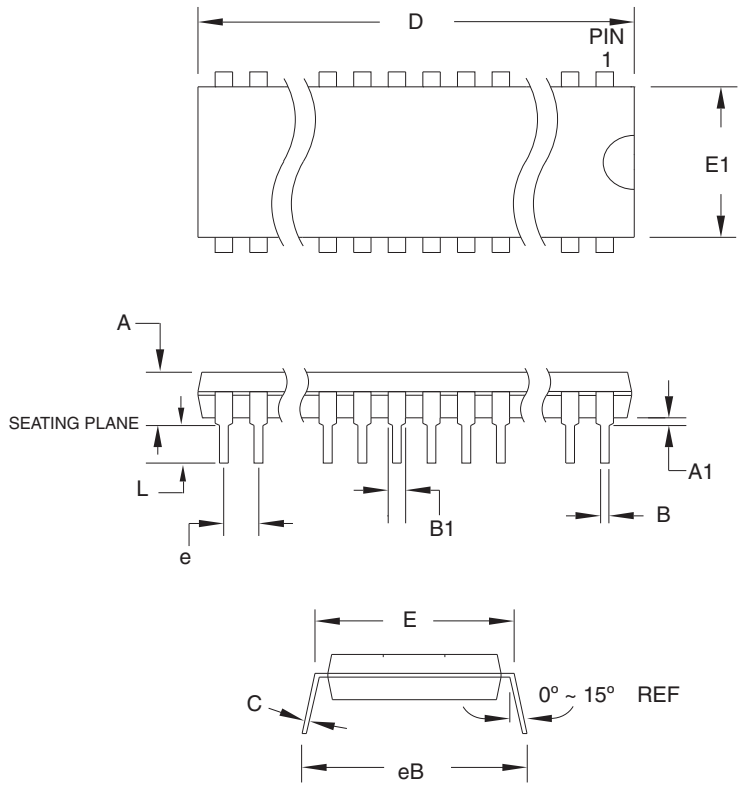
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

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

10/5/2001

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

40P6



COMMON DIMENSIONS
(Unit of Measure = mm)

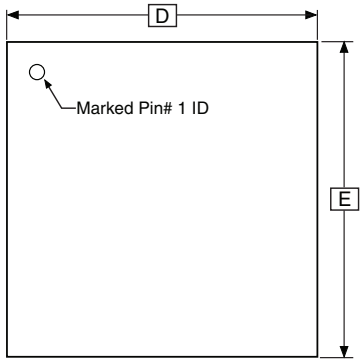
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

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").

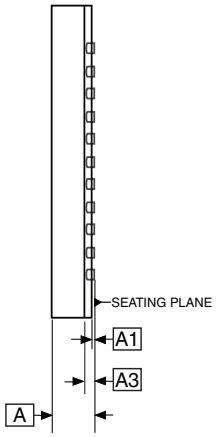
09/28/01

	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

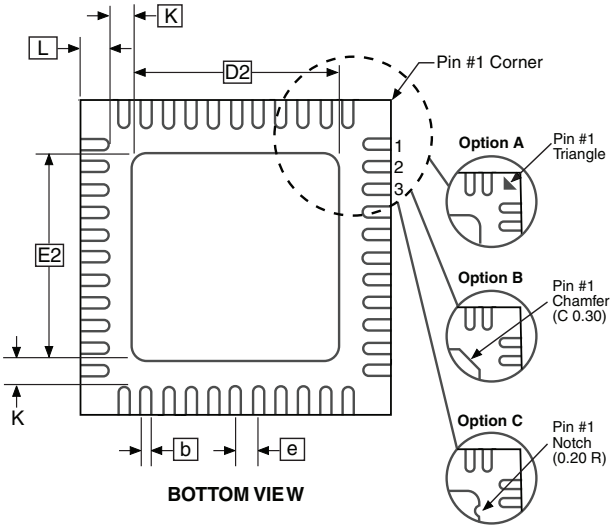
44M1



TOP VIEW



SIDE VIEW



BOTTOM VIEW

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

COMMON DIMENSIONS
(Unit of Measure = mm)

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

9/26/08

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

Errata

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

ATmega16(L) Rev. M

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

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

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

   Problem Fix/Workaround

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

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

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

   Problem Fix / Workaround

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

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

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

   Problem Fix / Workaround

   – If ATmega16 is the only device in the scan chain, the problem is not visible.

   – Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.

   – If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. L

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

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

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

Problem Fix/Workaround

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

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

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

   Problem Fix / Workaround

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

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

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

   Problem Fix / Workaround

   – If ATmega16 is the only device in the scan chain, the problem is not visible.

   – Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.

   – If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. K

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

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

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

   Problem Fix/Workaround

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

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

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

Problem Fix / Workaround

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

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

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

   Problem Fix / Workaround

   – If ATmega16 is the only device in the scan chain, the problem is not visible.

   – Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.

   – If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. J

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

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

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

   Problem Fix/Workaround

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

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

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

   Problem Fix / Workaround

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

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

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

Problem Fix / Workaround

– If ATmega16 is the only device in the scan chain, the problem is not visible.

– Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.

– If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

Problem Fix / Workaround

Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. I

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

1. First Analog Comparator conversion may be delayed

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

Problem Fix/Workaround

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

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

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

Problem Fix / Workaround

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

3. IDCODE masks data from TDI input

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

Problem Fix / Workaround

– If ATmega16 is the only device in the scan chain, the problem is not visible.

– Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.

– If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

ATmega16(L) Rev. H

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

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

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

   Problem Fix/Workaround

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

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

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

   Problem Fix / Workaround

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

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

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

   Problem Fix / Workaround

   – If ATmega16 is the only device in the scan chain, the problem is not visible.

   – Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.

   – If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the fist device in the chain.

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

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

   Problem Fix / Workaround

   Always use OUT or SBI to set EERE in EECR.

Datasheet Revision History

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

Rev. 2466T-07/10

1. Corrected use of comma in formula Rp in Table 120, “Two-wire Serial Bus Requirements,” on page 294.
2. Updated document according to Atmel’s Technical Terminology
3. Note 6 and Note 7 under Table 120, “Two-wire Serial Bus Requirements,” on page 294 have been removed.

Rev. 2466S-05/09

1. Updated “Errata” on page 340.
2. Updated the last page with Atmel’s new adresses.

Rev. 2466R-06/08

1. Added “Not recommended for new designs” note in Figure on page 1.

Rev. 2466Q-05/08

1. Updated “Fast PWM Mode” on page 77 in “8-bit Timer/Counter0 with PWM” on page 71:
   – Removed the last section describing how to achieve a frequency with 50% duty cycle waveform output in fast PWM mode.
2. Removed note from Feature list in “Analog to Digital Converter” on page 204.
3. Removed note from Table 84 on page 218.
4. Updated “Ordering Information” on page 336:
   - Commercial ordering codes removed.
   - Non Pb-free package option removed.

Rev. 2466P-08/07

1. Updated “Features” on page 1.
2. Added “Data Retention” on page 6.
3. Updated “Errata” on page 340.
4. Updated “Slave Mode” on page 140.

Rev. 2466O-03/07

1. Updated “Calibrated Internal RC Oscillator” on page 29.
2. Updated C code example in “USART Initialization” on page 149.
3. Updated “ATmega16 Boundary-scan Order” on page 241.
4. Removed “premilinary” from “ADC Characteristics” on page 297.
5. Updated from V to mV in “I/O Pin Input Hysteresis vs. V_{CC}” on page 317.
6. Updated from V to mV in “Reset Input Pin Hysteresis vs. V_{CC}” on page 318.

Rev. 2466N-10/06

1. Updated "Timer/Counter Oscillator" on page 31.
2. Updated "Fast PWM Mode" on page 102.
3. Updated Table 38 on page 83, Table 40 on page 84, Table 45 on page 111, Table 47 on page 112, Table 50 on page 128 and Table 52 on page 129.
4. Updated C code example in "USART Initialization" on page 149.
5. Updated "Errata" on page 340.

Rev. 2466M-04/06

1. Updated typos.
2. Updated "Serial Peripheral Interface – SPI" on page 135.
3. Updated Table 86 on page 221, Table 116 on page 276 ,Table 121 on page 295 and Table 122 on page 297.

Rev. 2466L-06/05

1. Updated note in "Bit Rate Generator Unit" on page 178.
2. Updated values for V_{INT} in "ADC Characteristics" on page 297.
3. Updated "Serial Programming Instruction set" on page 276.
4. Updated USART init C-code example in "USART" on page 144.

Rev. 2466K-04/05

1. Updated "Ordering Information" on page 336.
2. MLF-package alternative changed to "Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF".
3. Updated "Electrical Characteristics" on page 291.

Rev. 2466J-10/04

1. Updated "Ordering Information" on page 336.

Rev. 2466I-10/04

1. Removed references to analog ground.
2. Updated Table 7 on page 28, Table 15 on page 38, Table 16 on page 42, Table 81 on page 209, Table 116 on page 276, and Table 119 on page 293.
3. Updated "Pinout ATmega16" on page 2.
4. Updated features in "Analog to Digital Converter" on page 204.
5. Updated "Version" on page 229.
6. Updated "Calibration Byte" on page 261.
7. Added "Page Size" on page 262.

Rev. 2466H-12/03

1. Updated "Calibrated Internal RC Oscillator" on page 29.

Rev. 2466G-10/03

1. Removed “Preliminary” from the datasheet.
2. Changed ICP to ICP1 in the datasheet.
3. Updated “JTAG Interface and On-chip Debug System” on page 36.
4. Updated assembly and C code examples in “Watchdog Timer Control Register – WDTCR” on page 43.
5. Updated Figure 46 on page 103.
6. Updated Table 15 on page 38, Table 82 on page 217 and Table 115 on page 276.
7. Updated “Test Access Port – TAP” on page 222 regarding JTAGEN.
8. Updated description for the JTD bit on page 231.
9. Added note 2 to Figure 126 on page 252.
10. Added a note regarding JTAGEN fuse to Table 105 on page 260.
11. Updated Absolute Maximum Ratings* and DC Characteristics in “Electrical Characteristics” on page 291.
12. Updated “ATmega16 Typical Characteristics” on page 299.
13. Fixed typo for 16 MHz QFN/MLF package in “Ordering Information” on page 336.
14. Added a proposal for solving problems regarding the JTAG instruction IDCODE in “Errata” on page 340.

Rev. 2466F-02/03

1. Added note about masking out unused bits when reading the Program Counter in “Stack Pointer” on page 12.
2. Added Chip Erase as a first step in “Programming the Flash” on page 288 and “Programming the EEPROM” on page 289.
3. Added the section “Unconnected pins” on page 55.
4. Added tips on how to disable the OCD system in “On-chip Debug System” on page 34.
5. Removed reference to the “Multi-purpose Oscillator” application note and “32 kHz Crystal Oscillator” application note, which do not exist.
6. Added information about PWM symmetry for Timer0 and Timer2.
7. Added note in “Filling the Temporary Buffer (Page Loading)” on page 253 about writing to the EEPROM during an SPM Page Load.
8. Removed ADHSM completely.

9. Added Table 73, “TWI Bit Rate Prescaler,” on page 182 to describe the TWPS bits in the “TWI Status Register – TWSR” on page 181.

10. Added section “Default Clock Source” on page 25.

11. Added note about frequency variation when using an external clock. Note added in “External Clock” on page 31. An extra row and a note added in Table 118 on page 293.

12. Various minor TWI corrections.

13. Added “Power Consumption” data in “Features” on page 1.

14. Added section “EEPROM Write During Power-down Sleep Mode” on page 22.

15. Added note about Differential Mode with Auto Triggering in “Prescaling and Conversion Timing” on page 207.

16. Added updated “Packaging Information” on page 337.

Rev. 2466E-10/02

1. Updated “DC Characteristics” on page 291.

Rev. 2466D-09/02

1. Changed all Flash write/erase cycles from 1,000 to 10,000.

2. Updated the following tables: Table 4 on page 26, Table 15 on page 38, Table 42 on page 85, Table 45 on page 111, Table 46 on page 111, Table 59 on page 143, Table 67 on page 167, Table 90 on page 235, Table 102 on page 258, “DC Characteristics” on page 291, Table 119 on page 293, Table 121 on page 295, and Table 122 on page 297.

3. Updated “Errata” on page 340.

Rev. 2466C-03/02

1. Updated typical EEPROM programming time, Table 1 on page 20.

2. Updated typical start-up time in the following tables:

   Table 3 on page 25, Table 5 on page 27, Table 6 on page 28, Table 8 on page 29, Table 9 on page 29, and Table 10 on page 29.

3. Updated Table 17 on page 43 with typical WDT Time-out.

4. Added Some Preliminary Test Limits and Characterization Data.

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

   Table 15 on page 38, Table 16 on page 42, Table 116 on page 272 (table removed in document review #D), “Electrical Characteristics” on page 291, Table 119 on page 293, Table 121 on page 295, and Table 122 on page 297.

5. Updated TWI Chapter.

   Added the note at the end of the “Bit Rate Generator Unit” on page 178.

6. Corrected description of ADSC bit in “ADC Control and Status Register A – ADCSRA” on page 219.

7. Improved description on how to do a polarity check of the ADC doff results in “ADC Conversion Result” on page 216.

8. **Added JTAG version number for rev. H in Table 87 on page 229.**

9. **Added not regarding OCDEN Fuse below Table 105 on page 260.**

10. **Updated Programming Figures:**

    Figure 127 on page 262 and Figure 136 on page 273 are updated to also reflect that AVCC must be connected during Programming mode. Figure 131 on page 269 added to illustrate how to program the fuses.

11. **Added a note regarding usage of the "PROG_PAGELOAD ($6)" on page 280 and "PROG_PAGEREAD ($7)" on page 280.**

12. **Removed alternative algortihm for leaving JTAG Programming mode.**

    See "Leaving Programming Mode" on page 288.

13. **Added Calibrated RC Oscillator characterization curves in section "ATmega16 Typical Characteristics" on page 299.**

14. **Corrected ordering code for QFN/MLF package (16MHz) in "Ordering Information" on page 336.**

15. **Corrected Table 90, "Scan Signals for the Oscillators(1)(2)(3)," on page 235.**

Headquarters

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

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

Bluetooth to serial HC-06 wireless module

Product Description:

1，Mainstream CSR Bluetooth chip, Bluetooth V2.0 protocol standards

2，serial module operating voltage 3.3V.

3，the baud rate for 1200, 2400, 4800, 9600, 19200, 38400 and users can be set

4，core module size: 28mm x 15 mm x 2.35mm.

5，the working current: 40mA

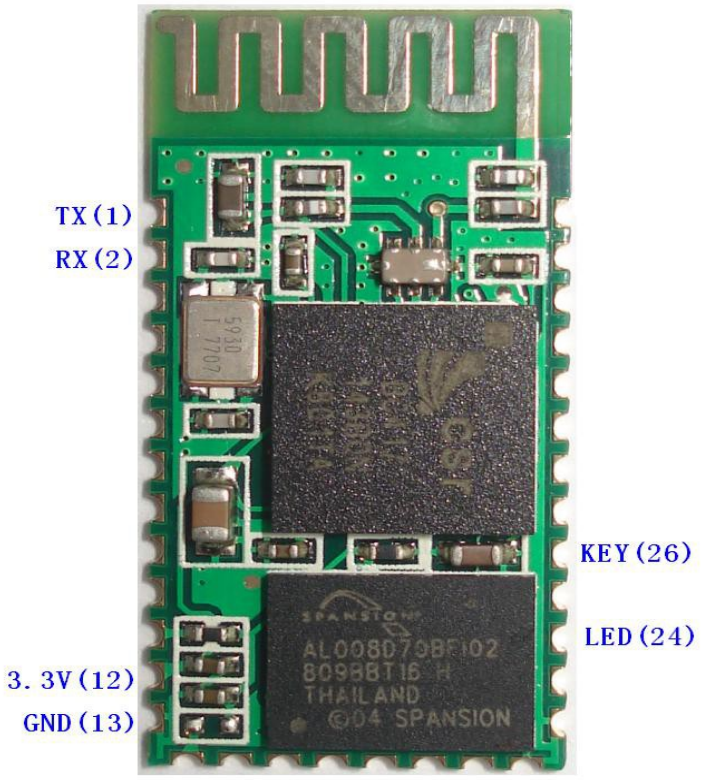
6，Sleep current: <1mA

7，for the GPS navigation system, utility meter reading system, the industrial field, collecting and controlling system.

8，with a Bluetooth laptop computer to the Bluetooth adapter, PDA and other devices to seamlessly connect

The module's host and slave, the host and slave pairing communication from the machine and from the machine or between the host and the host can not communicate, communication function and computers, mobile phones and other Bluetooth pairing purchase default slave , requires that the host needs to be indicated]

Main distinction: 1, if the chip is not specified on, the lights flash slow main fast from; September 2,2009, all manufactured host will be playing in the IC a hook or paste There are the "main" characters, there is no hook or not affixed to the word "master" is the slave. The date of manufacture can be obtained from the Bluetooth address]



The factory default parameters:

Slave, baud rate: 9600, n, 8,1. Passkey: 1234; need host mode, indicate when the orders.

Second, AT command set as follows:

1, test communications

Send: AT (return OK, one second left and right)

Back: OK

2, change the Bluetooth serial communication baud rate

Send: AT + BAUD1

Back to: OK1200

Send: AT + BAUD2

Back to: OK2400

.....

1 --------- 1200

2 --------- 2400

3 --------- 4800

4 --------- 9600

5 --------- 19200

6 --------- 38400

7 --------- 57600

8 --------- 115200

9 --------- 230400

A --------- 460800

B --------- 921600

C --------- 1382400

Not recommended to use more than 115200 baud rate, signal interference causes the system to instability.

The settings over 115,200 with a computer is not available, use microcontroller programming in higher than 115200 to use this baud rate and re-issue the AT command set low baud rate

AT command set baud rate, the next power do not need to set up and can be powered down to save the baud rate.

3, change the Bluetooth name

Send: AT + NAMEname

Back to: OKname

Parameter name: To set the current name, the name of the Bluetooth search. 20 characters or less.

Example: Sending AT + NAMEbill_gates

Back OKname

The Bluetooth name changed to bill_gates

The parameters can be powered down to save, simply modify the time. PDA the end refresh can see the Bluetooth name changed.

4, change the Bluetooth pairing password

Send: AT + PINxxxx

Returns: OKsetpin

Parameter xxxx: To set a passcode, 4 bytes, this command can be used from the machine or host. The slave adapter or mobile phone pops up to enter when pairing the password window, manually enter this parameter can be connected from the machine. Host in the main Bluetooth module connected digital camera, digital camera from the machine, find the camera pairing password, and then set up the White Bluetooth module, the main Bluetooth module can automatically connect the camera.

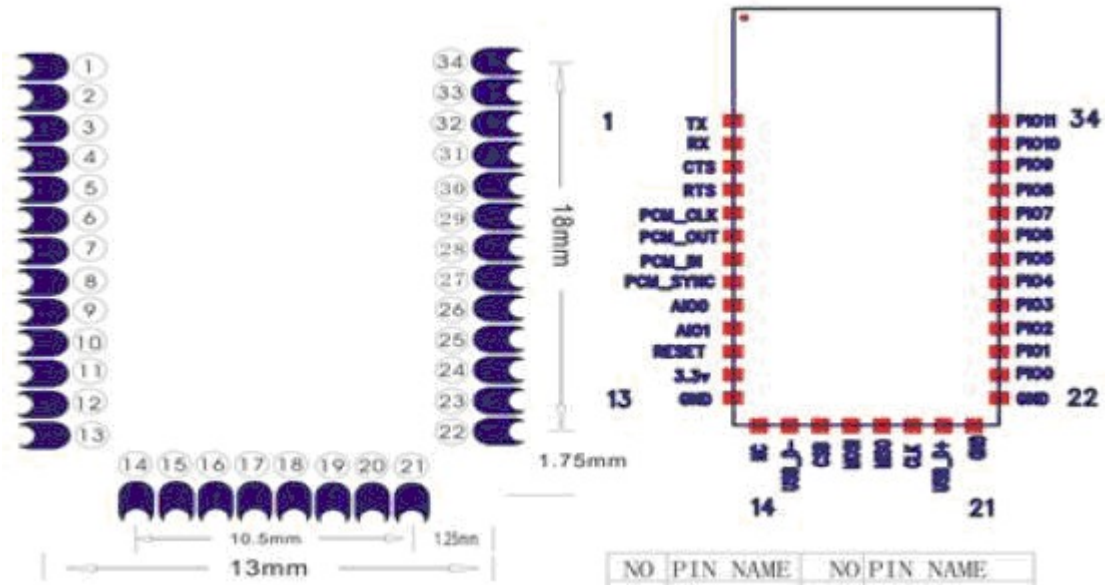
Example: Sending AT + PIN8888

Back OKsetpin

The Bluetooth pairing password to 8888, module paired at the factory default password is 1234.

The parameters can be powered down to save, simply modify the time.

Package pin diagram:

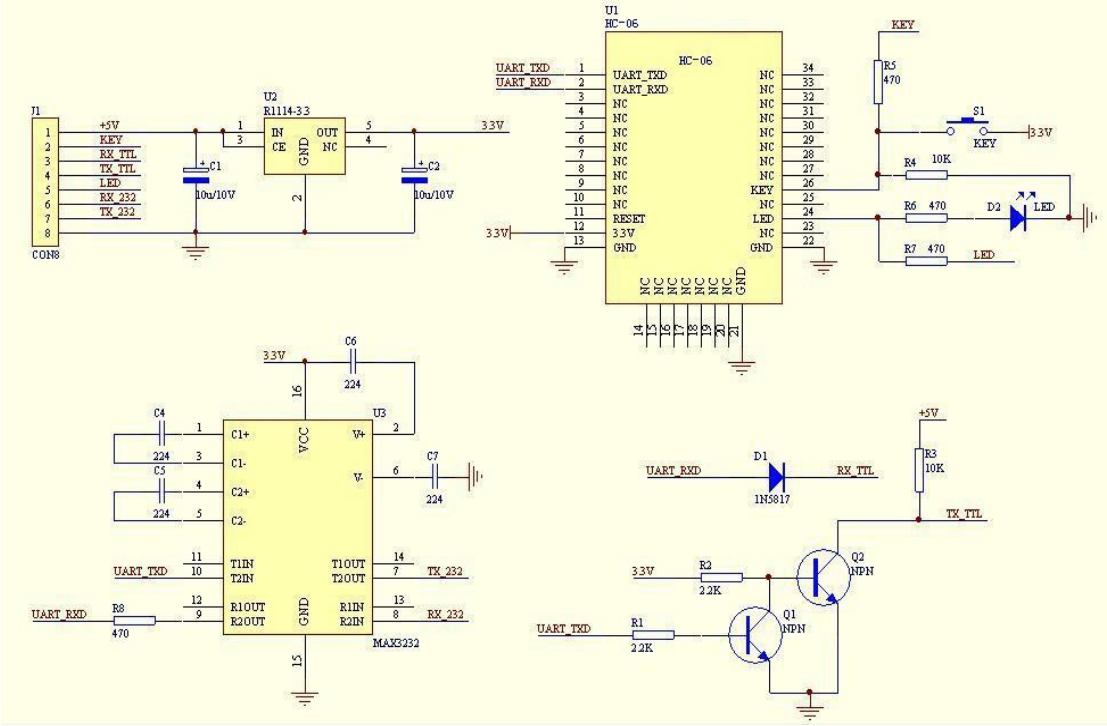


NO	PIN NAME	NO	PIN NAME
1	TX	20	USB D+
2	RX	21	GND
3	CTS	22	GND
4	RTS	23	P100
5	PCM CLK	24	P101
6	PCM OUT	25	P102
7	PCM IN	26	P103
8	PCM SYNC	27	P104
9	AIO0	28	P105
10	AIO1	29	P106
11	RESET	30	P107
12	3.3V	31	P108
13	GND	32	P109
14	NC	33	P1010
15	USB D-	34	P1011
16	CSB		
17	MOSI		
18	MISO		
19	CLK		

The old customer please refer to (schematics, wiring diagrams, etc.)

This module is for upgrade BCM_LV module version

The figure below shows the core module peripheral circuit schematic diagram of the typical



application

If you use the factory default baud rate, do not want to modify the contents of the baud rate below do not need to care about:

The AT command for the master and slave. When used in pairs, the master and slave can be different baud rate can also transfer data, but the host and the device connected to the host baud rate to be the same, but also slave and slave devices connected to the same baud rate .

Setting mode:

Initial communication parameters for 9600, N, 8,1, before pairing (pairing indicator flashes) Send to modify the baud rate command

Before sending AT commands to ensure that the hardware is connected as follows:

The eight-pin interface with base plate, the first pin external power supply (3.3 to 5V), the sixth leg connected to the computer COM1 port (DB9 male) pin, pin 7 computer COM1 port pin, the eighth pin is connected to the computer COM1 port 5 feet. Using HyperTerminal or serial debugging assistant to open the COM1 port of the computer, enter the text "AT" to manually send.

Special note: master and slave pair can communicate between master and slave baud rate is required to be consistent between master and slave is to take the Bluetooth protocol, rather than serial port protocol.

1, test communications

Send: AT (return OK, one second left and right)

Back: OK

2, change the Bluetooth serial communication baud rate

Send: AT + BAUD1

Back to: OK1200

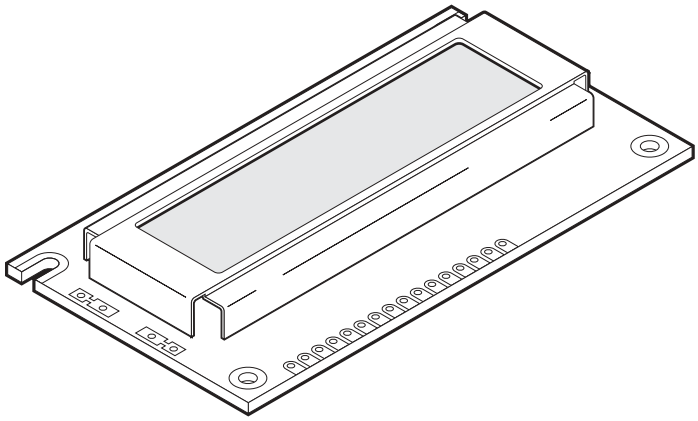
Send: AT + BAUD2
Back to: OK2400
.....
1 --------- 1200
2 --------- 2400
3 --------- 4800
4 --------- 9600
5 --------- 19200
6 --------- 38400
7 --------- 57600
8 --------- 115200
9 --------- 230400
A --------- 460800
B --------- 921600
C --------- 1382400
Not recommended to use more than 115200 baud rate, signal interference causes the system to instability.
The settings over 115,200 with a computer is not available, use microcontroller programming in higher than 115200 to use this baud rate and re-issue the AT command set low baud rate
AT command set baud rate, the next power do not need to set up and can be powered down to save the baud rate.

3, change the Bluetooth name (February 2008 after 24 new features)
Send: AT + NAMEname
Back to: OKname
Parameter name: To set the current name, the name of the Bluetooth search. 20 characters or less.
Example: Sending AT + NAMEbill_gates
Back OKname
The Bluetooth name changed to bill_gates
The parameters can be powered down to save, simply modify the time. PDA the end refresh can see the Bluetooth name changed.
4, change the Bluetooth pairing password
Send: AT + PINxxxx
Returns: OKsetpin
Parameter xxxx: To set a passcode, 4 bytes, this command can be used from the machine or host. The slave adapter or mobile phone pops up to enter the pairing password window, then manually enter this parameter
The number can be connected to the slave. Host in the main Bluetooth module connected digital camera, digital camera from the machine, find the pairing password of the camera, and then set up the White Bluetooth module, the master Bluetooth module can be。

ALPHANUMERIC LCD DISPLAY (16 x 2)

Order Code

LED008 16 x 2 Alphanumeric Display
FRM010 Serial LCD Firmware (optional)



Contents

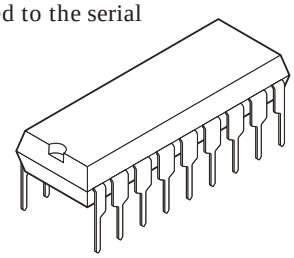
1 x 16x2 Alphanumeric Display
1 x data booklet

Introduction

Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. A full list of the characters and symbols is printed on pages 7/8 (note these symbols can vary between brand of LCD used). This booklet provides all the technical specifications for connecting the unit, which requires a single power supply (+5V).

Further Information

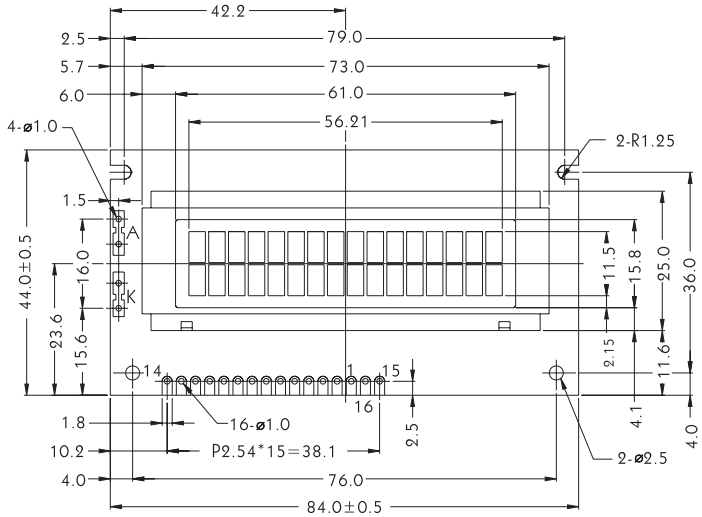
Available as an optional extra is the Serial LCD Firmware, which allows serial control of the display. This option provides much easier connection and use of the LCD module. The firmware enables microcontrollers (and microcontroller based systems such as the PICAXE) to visually output user instructions or readings onto an LCD module. All LCD commands are transmitted serially via a single microcontroller pin. The firmware can also be connected to the serial port of a computer.

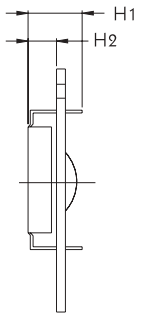


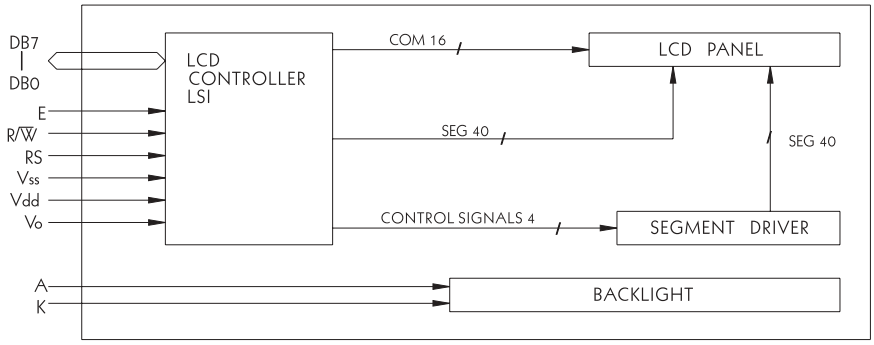
An example PICAXE instruction to print the text 'Hello' using the **serout** command is as follows:

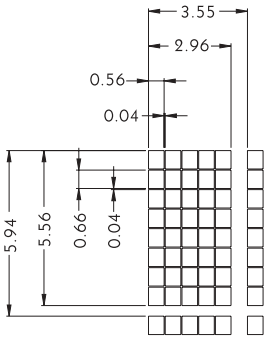
```
serout 7,T2400,("Hello")
```

Outline Dimension and Block Diagram









The tolerance unless classified ±0.3mm

MECHANICAL SPECIFICATION			
Overall Size	84.0 * 44.0	Module	H2 / H1
View Area	61.0 * 15.8	W/O B/L	5.1 / 9.7
Dot Size	0.56 * 0.66	EL B/L	5.1 / 9.7
Dot Pitch	0.60 * 0.70	LED B/L	9.4 / 14.0

PIN ASSIGNMENT		
Pin no.	Symbol	Function
1	Vss	Power supply (GND)
2	Vdd	Power supply (+5V)
3	Vo	Contrast Adjust
4	RS	Register select signal
5	R/W	Data read /write
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for LED B/L (+)
16	K	Power supply for LED B/L (−)

ABSOLUTE MAXIMUM RATING					
Item	Symbol	Conditions	Min.	Max.	Unit
Power Supply Voltage	Vdd−Vss	−	0	7	V
LCD Driving Supply Voltage	Vdd−Vee	−	0	13	V
Input Voltage	Vin	−	−0.3	Vdd+0.3	V
Operating Temperature	Topr	Nor.	0	50	°C
Storage Temperature	Tstg	Nor.	−20	+70	°C

ELECTRICAL CHARACTERISTICS (Vdd =+5V, Ta =25°C)						
Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Logic Supply Voltage	Vdd	−	4.5	5	5.5	V
"H" Input Voltage	V_{IH}	−	2.2	−	−	V
"L" Input Voltage	V_{IL}	−	−	−	0.6	V
"H" Output Voltage	V_{OH}	−	2.4	−	−	V
"L" Output Voltage	V_{OL}	−	−	−	0.4	V
Supply Current	Idd	−	2	−	−	mA
LCD Driving Voltage	V_{LCD}	Vdd−Vo	4.3	−	4.8	V

Electrical Characteristics

Vdd = 5V±5%
Vss = 0V

Item	Symbol	Condition	Standard value			Unit	Applicable terminal
			Min.	Typ.	Max.		
Power voltage	Vdd		4.5	5.00	5.5	V	Vdd
Input H- level voltage	VIH		2.2	—	Vdd	V	RS, R/$\overline{W}$, E DB0~DB7
Input L - level voltage	VIL		−0.3	—	0.6	V	
Output H - level voltage	VOH	−IOH = 0.205mA	2.4	—	—	V	DB0~DB7
Output L - level voltage	VOL	IOL = 1.2mA	—	—	0.4	V	
I/O leakage current	IIL	Vin = 0~Vdd	−1	—	1.0	μA	RS, R/$\overline{W}$, E DB0~DB7
Supply current	Idd	Vdd = 5V	2	—	—	mA	Vdd
LCD operating voltage	VLCD	Vdd—V0	3.0	—	11.0	V	V0

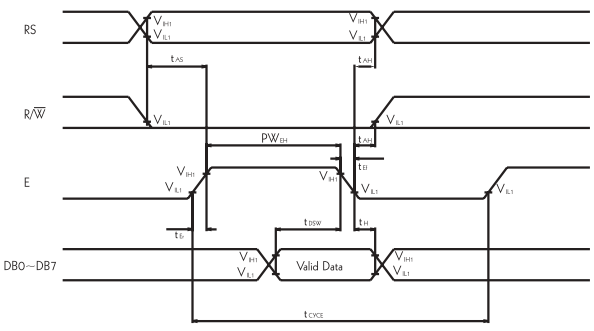
Timing Characteristics

Vdd = 5V±5%
Vss = 0V

Item		Symbol	Min.	Max.	Unit
Enable cycle time		TCYCE	500	—	ns
Enable pulse width	"High" level	PWEH	220	—	ns
Enable rise / fall time		TER, TEF	—	25	ns
Set-up time	RS, R/$\overline{W}$, E	TAS	40	—	ns
Address hold time		TAH	10	—	ns
Data set-up time		TDSH	60	—	ns
Data delay time		TDDR	60	120	ns
Data hold time (writing)		TH	10	—	ns
Data hold time (reading)		TDHR	20	—	ns
Clock oscillating frequency		TOSC	270(TYP.)		KHz

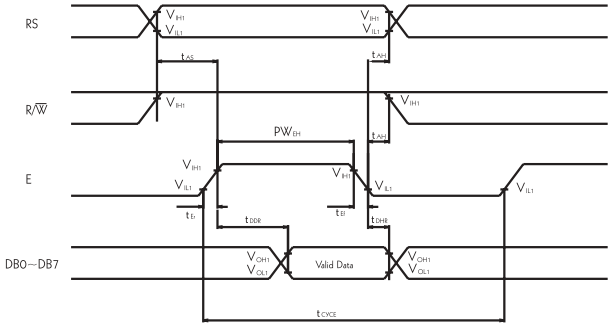
Timing Chart

◆ FIG.1 WRITE OPERATION



(Write Data from MPU to MODULE)

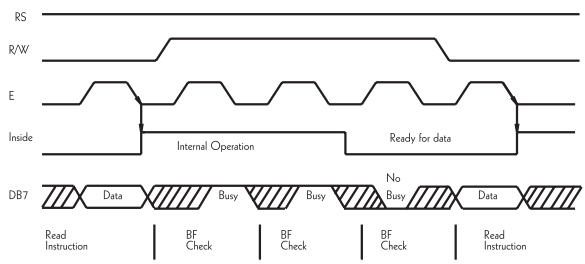
◆ FIG.2 READ OPERATION

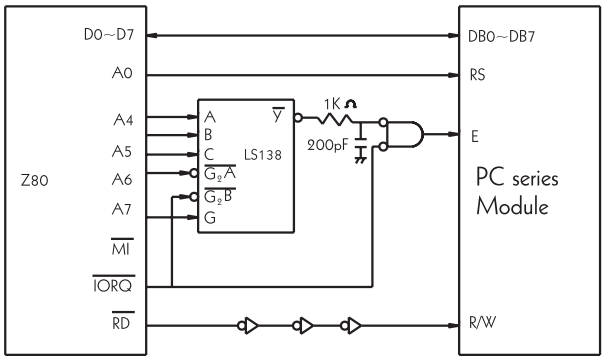


(Read Data from MODULE to MPU)

Interface with MPU

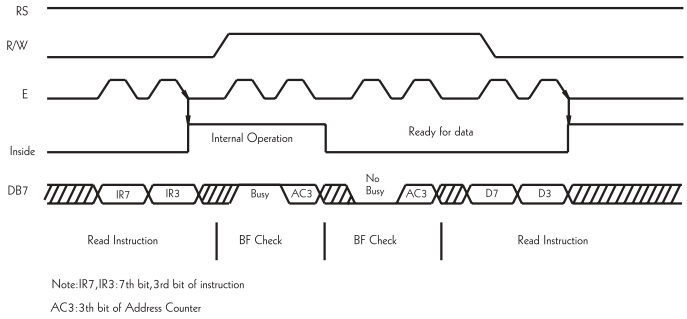
Example of Interface with 8-bit MPU (Z80)

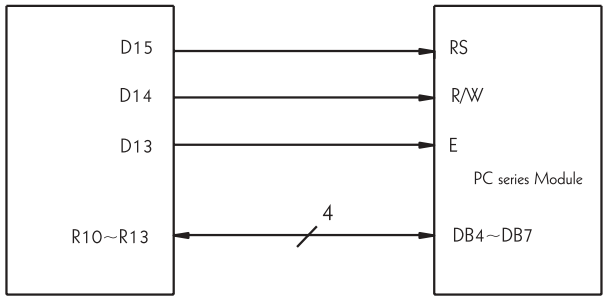




Example of interface with 4-bit MPU

Interface with 4-bit MPU can be made through I / O port of 4-bit MPU. If there are enough I / O ports, data can be transfered by 8-bit, however, if there are not data transfer can be done by 4-bit in twice (select interface is 4-bit long), and timing sequence will be complicated in this case. Please take into account that 2 cycles of BF check is necessary, while 2 cycles of data transfer are also necessary.





Features

(1) Interface with 8-bit or 4-bit MPU is available.
(2) 192 kind of alphabets, numerals, symbols and special characters can be displayed by built-in character generator (ROM).
(3) Other preferred characters can be displayed by character generator (RAM).
(4) Various functions of instruction are available by programming.
- Clear display • Cursor at home • On / off cursor
- Blink character • Shift display • Shift cursor
- Read / write display data.....etc.

(5) Compact and light weight design which can be easily assembled in devices.
(6) Single power supply +5V drive (except for extended temp. type).
(7) Low power consumption.

*Interface between data bus line and 4-bit or 8-bit MPU is available. Data transfer are made in twice in case of 4-bit MPU, and once in case of 8-bit MPU.

If interface data is 4-bit long

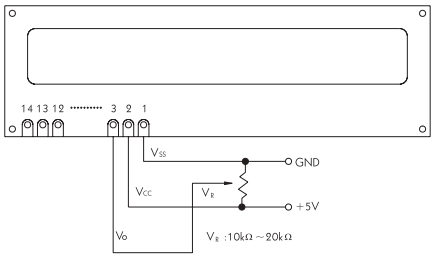
Data transfer are made through 4 bus lines from DB4 to DB7. (while the rest of 4 bus lines fromDB0 to DB3 are not used.) Data transfer with MPU are completed when 4-bit data are transfered in twice.
(first upper 4-bit data. then lower 4-bit data.)

If interface data is 8-bit long

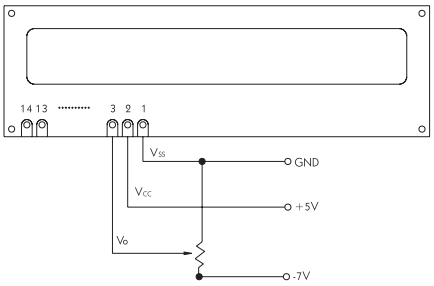
Data transfer are made through all of 8 bus lines from DB0 to DB7.

Example of Power Supply

Normal Temperature Type

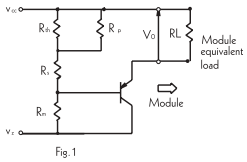


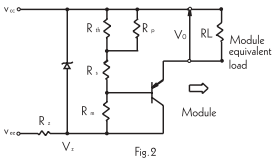
Extended Temperature Type



Examples of Temperature Compensation Circuits for Extended Temp Type. (Only for reference)

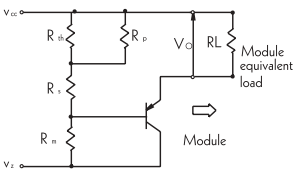
(A) 1/8Duty—1/4Bias

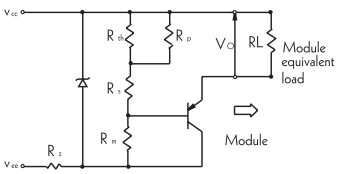




Thermistor: Rth(25°C)=15[k-ohm], B=4200[K]
Resistors: Rp=30[k-ohm]], Rs=6.8[k-ohm], Rm=3.3[k-ohm]
Transistor: PNP Type
Vcc: +5V , Vss: 0V (Logic Supply)
Vz: -8[V] (-7.8 to -8.2[V])
Vee<Vz[V]. Rz=(Vz-Vee) / 5[k-ohm]

(B) 1/16Duty—1/5Bias





Thermistor: Rth(25°C)=15[k-ohm], B=4200[K]
Resistors: Rp=510[k-ohm]], Rs=8.2[k-ohm], Rm=3.9[k-ohm]
Transistor: PNP Type
Vcc: +5V , Vss: 0V (Logic Supply)
Vz: -11[V] (-10.725 to -11.275[V])
Vee<Vz[V]. Rz=(Vz-Vee) / 5[k-ohm]

Instructions

Instruction	Code										Description	Executed Time(max.)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0)	1.64mS
Cursor At Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	1.64mS
Entry Mode Set	0	0	0	0	0	0	0	1	1/D	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40µS
Display On / Off Control	0	0	0	0	0	0	1	D	C	B	Sets ON / OFF of all display (D), cursor NO / OFF (C), and blink of cursor position character (B).	40µS
Cursor / Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing DD RAM contents.	40µS
Function Set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL) number of display lines (L) and character font (F)	40µS
CG RAM Address Set	0	0	0	1	ACG						Sets the CG RAM address. CG RAM data is sent and received after this setting.	40µS
DD RAM Address Set	0	0	1	ADD							Sets the DD RAM address. DD RAM data is sent and received after this setting.	40µS
Busy Flag / Address Read	0	1	BF	AC							Reads Busy flag (FB) indicating internal operation is being performed and reads address counter counts.	0µS
CG RAM / DD RAM Data Write	1	0	WRITE DATA								Writes data into DD RAM or CG RAM.	40µS
CG RAM / DD RAM Data Read	1	1	READ DATA								Reads data from DD RAM or CG RAM.	40µS

Code		Descripion	Executed Time (max)
I/D = 1:Increment I/D = 0:Decrement S= 1:With display shift S/C= 0:cursor movement R/L= 1: Shift to the right R/L= 0: Shift to the left DL = 1: 8-bit	DL = 0:4-bit N = 1:2lines N = 0:1line F = 1:5×10dots F = 0.5×7dots BF = 1:Internal operation is being performed BF = 0:Instruction acceptable	DD RAM:Display Data RAM CG RAM:Character Generator RAM ACG:CG RAM Address ADD:DD RAM Address Corresponds to cursor address. AC:Address Counter, used for both DD RAM and CG RAM *:Invalid	fcp or fosc = 250KHz However. when frequency changes. eecution time also changes Example if fcp or fosc is 270KHz, 70µS x 250 / 270 = 37µS

Power Supply Reset

The internal reset circuit will be operated properly when the following power supply conditions are satisfied. If it is not operated properly, please perform initial setting along with the instruction.

Item	Symbol	Measuring Condition	Standard Value			Unit
			Min.	Typ.	Max.	
Power Supply RISE Time	tree	—	0.1	—	10	mS
Power Supply CFF Time	toff	—	1	—	—	mS

Reset function

◆ **Initialization Made by Internal Reset Circuit**

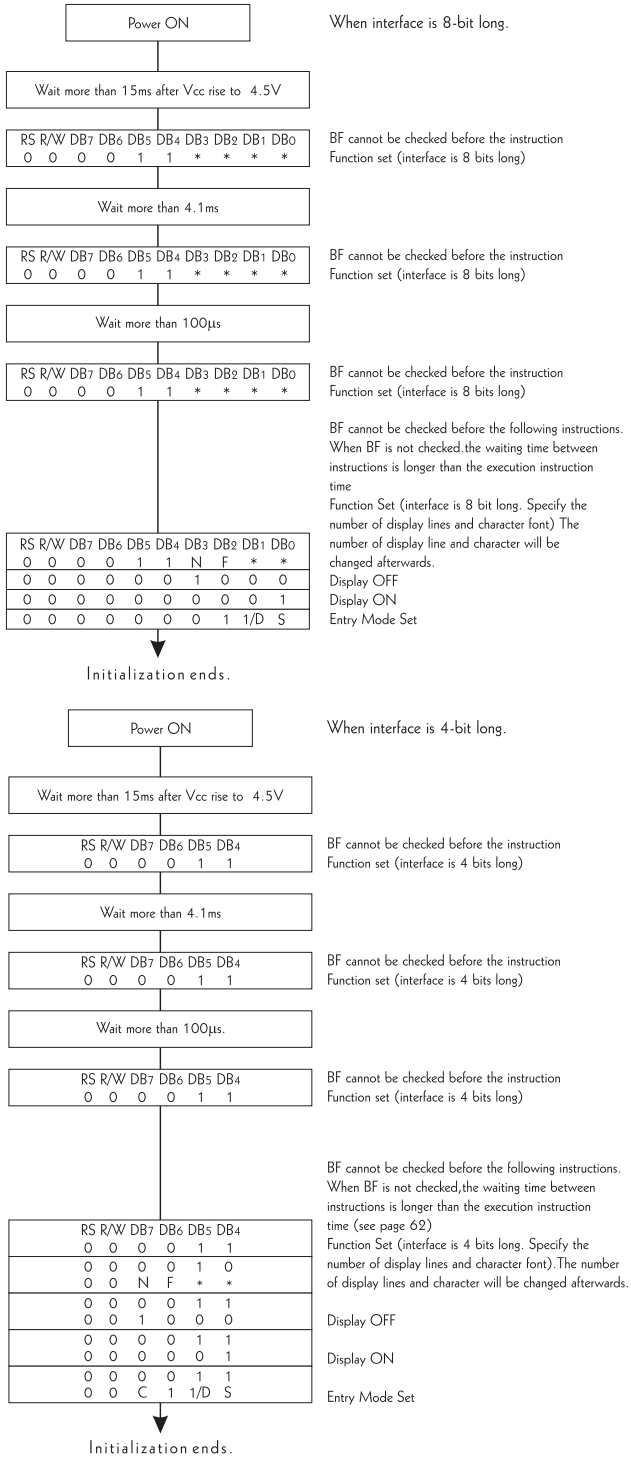
HD44780 automatically initializes (resets) when power is supplied (builtin internal reset circuit). The following instructions are executed in initialization. The busy flag (BF) is kept in busy state until initialization ends. (BF=1) The busy state is 10 ms after Vdd reachs to 4.5V.

(1) Display clear
(2) Function set
DL= 1:8 bit long interface data
DL= 0:4 bit F= 0:5 x 7dots character font
N= 1:2 lines
N= 0:1 line
(3) Display ON / OFF control
D= 0:Display OFF C= 0:Cursor OFF
B= 0:Blink OFF
(4) Entry mode set
1 / D= 1:+1(increment) S= 0:No shift

Note:When conditions stated in power supply conditions using internal reset circuit are not satisfied.The internal reset circuit will not operate properly and initialization will not be performed. Please make initialization using MPU along with instruction.

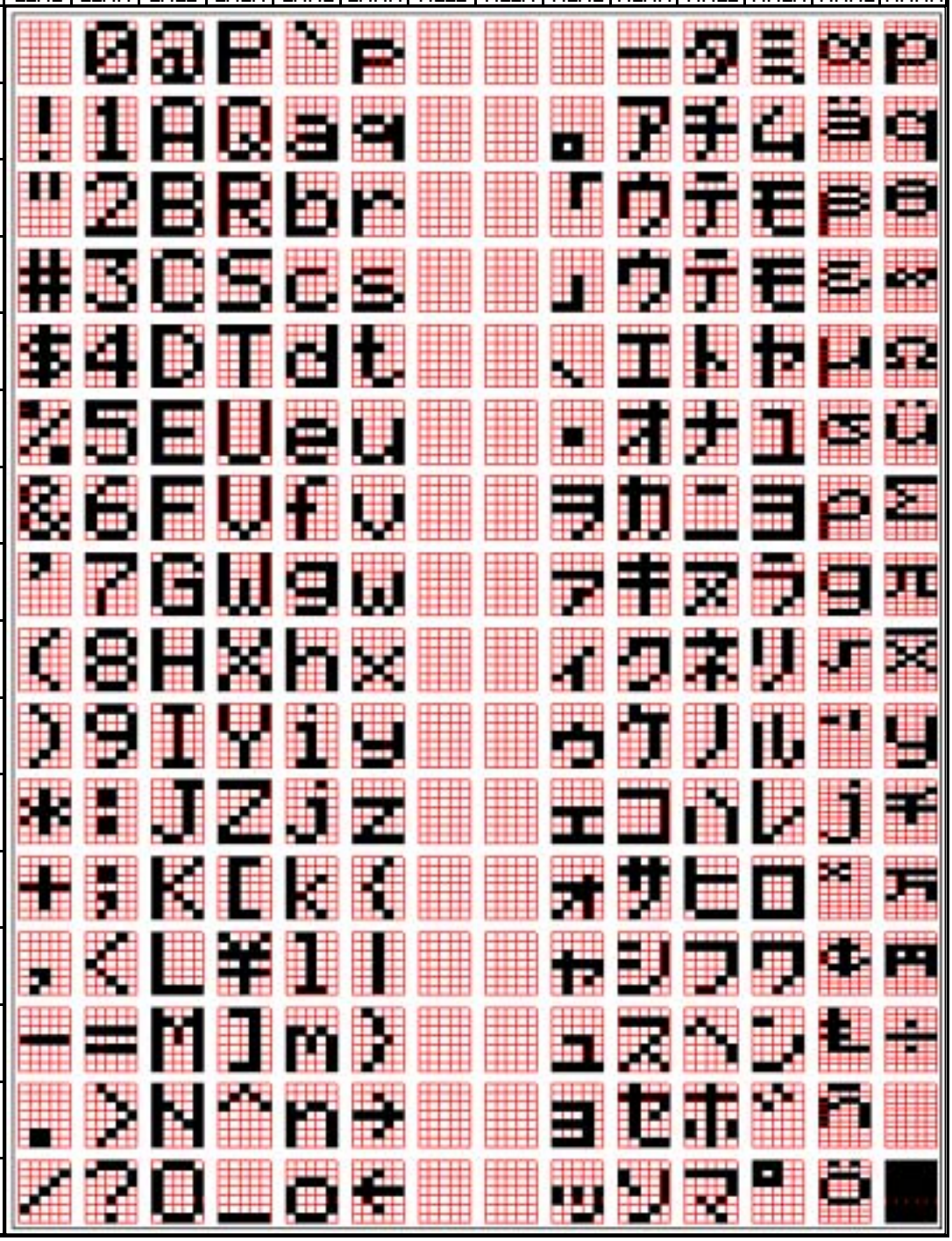
◆ **Initialization along with instruction**

If power supply conditions are not satisfied, which for proper operation of internal rest circuit, it is required to make initialzation along with instruction. Please make following procedures.



Standard Character Pattern (Powertip Module)

Standard Character Pattern (Elec & Eltek Module)

Upper(4bit) / Lower(4bit)	LLLL	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)														
LLLH	(2)														
LLHL	(3)														
LLHH	(4)														
LHLL	(5)														
LHLH	(6)														
LHHL	(7)														
LHHH	(8)														
HLLL	(1)														
HLLH	(2)														
HLHL	(3)														
HLHH	(4)														
HHLL	(5)														
HHLH	(6)														
HHHL	(7)														
HHHH	(8)														

1. Pemograman Mikrokontroller dengan Menggunakan BASCOM AVR

```
$regfile = "m16def.dat"      'Inisialisasi mikrokontroller yang digunakan.
$crystal = 8000000           'Inisialisasi Kristal yang digunakan.
$baud = 9600                 'Inisialisasi baud yang digunakan.

'Konfigurasi LCD
Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 , Db7 =
    Portb.7 , E = Portb.2 , Rs = Portb.0
```

Mengkonfigurasi LCD pin sebagai pin, ***Portb.4*** *sebagai data bit 4,* ***Portb.5*** *sebagai data bit 5,* ***Portb.6*** *sebagai data bit 1,* ***Portb.****7 sebagai data bit 7 lalu* ***Portb.2*** *sebagai enable (mulai menahan data ke LCD), dan* ***Portb.0*** *sebagai pemilih register.*

```
Config Lcd = 16 * 2          'Mengkonfirgurasi LCD 16x2
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Up , Compare B
    Pwm = Clear Up , Prescale = 1

Dim Cek As String * 1        'Deklarasi.
Dim Datas As String * 20     'Deklarasi.
Dim Datas2 As String * 20    'Deklarasi.
Dim Id As String * 20        'Deklarasi.
Dim I As Integer             'Deklarasi.
Dim Nilai As Integer         'Deklarasi.
Dim Nilai2 As Integer        'Deklarasi.
Cls                          'Membersihkan Program.

Cursor Off
Locate 1 , 1                 'Lokasi 1 banding 1.
Lcd "test data"              'Tampilan LCD.
Config Portd = Output        'Mengkonfigurasi port sebagai output.
                             Seluruh Port d sebagai input
Relay1 Alias Portd.6         'Portd.6 adalah nama lain untuk Relay1
Relay2 Alias Portd.7         'Portd.7 adalah nama lain untuk Relay2
Config Pind.7 = Output       'Mengkonfigurasi pin sebagai output.
                             Pind.7 sebagai input
Config Pind.6 = Output       'Mengkonfigurasi pin sebagai output.
                             Pind.6 sebagai input

Mulai:
Datas = ""
I = 0
Do                                   'Awalan Program
Do                                   'Awalan Program
```

```
Cek = Inkey()                          'Syarat Kondisi
'Lcd ; Cek
Loop Until Cek = "r" Or Cek = "g"      'Syarat Kondisi.

If Cek = "r" Then                      'Syarat Kondisi.
Do                                     'Awalan Program
Cek = Inkey()                          'Syarat Kondisi.
If Cek <> "" Then                      'Syarat Kondisi.
Incr I
Datas = Datas + Cek
End If                                 'Akhir dari pernyataan kondisional.
Loop Until Cek = "r"                   'Syarat Kondisi.

Nilai = Val(datas )
Cursor Off
Locate 1 , 1                           'Lokasi 1 banding 1.
Lcd "Data ID:v "
Nilai = Val(datas )
Locate 2 , 1                           'Lokasi 2 banding 1.


Lcd Nilai2 ; "     " ; "    "
If Nilai >= 0 And Nilai < 70 Then      'Jika nilai berlogika >= 0 dan nilai
                                       berlogika <70.
Relay1 = 0                             'maka Relay1 berlogika =0 atau OFF.
Relay2 = 0                             'maka Relay2 berlogika =0 atau OFF.
Cls                                    'Membersihkan Program.

Locate 2 , 10                          'Lokasi 2 banding 10.
Lcd "off"                              'Tampilan LCD "OFF"
End If                                 'Akhir dari pernyataan kondisional.
If Nilai >= 170 And Nilai <= 255 Then  'Jika nilai berlogika >= 170
                                       dan nilai berlogika <255.
Relay1 = 1                             'maka Relay1 berlogika = 1atau ON.
                                       'maka Relay1 berlogika = 1 atau ON.
Cls                                    'Membersihkan Program.

Locate 2 , 10                  'Lokasi 2 banding 10.
Lcd "on "                      'Tampilan LCD "ON"
End If                         'Akhir dari pernyataan kondisional.
Datas = ""
I = 0
End If                         'Akhir dari pernyataan kondisional.
If Cek = "g" Then
Do                             'Awalan Program
```

```
Cek = Inkey()
If Cek <> "" Then
Incr I
Datas = Datas + Cek
End If                    ‘Akhir dari pernyataan kondisional.

Loop Until Cek = "g"      ‘Syarat Kondisi.
Nilai = Val(datas )
Cursor Off

Locate 1 , 1                          ‘Lokasi 1 banding 1.
Lcd "Data ID:v "

Nilai = Val(datas )
Pwm1a = Nilai
Pwm1b = Nilai
Locate 2 , 1                          ‘Lokasi 2 banding 1.
Lcd Nilai ; "     " ; "    "
If Nilai >= 0 And Nilai < 20 Then     ‘Jika nilai berlogika >= 0
                                      dan nilai berlogika <20
Locate 2 , 10                         ‘Lokasi 2 banding 10.
Lcd "MIN "                            ‘Tampilan LCD MIN.
End If                                ‘Akhir dari pernyataan kondisional.

If Nilai >= 220 And Nilai <= 255 Then ‘Jika nilai berlogika >= 220
                                      dan nilai berlogika <255.
Locate 2 , 10                         ‘Lokasi 2 banding 10.
Lcd "MAX "                            ‘Tampilan LCD MAX.
End If                                ‘Akhir dari pernyataan kondisional.
Datas = ""
I = 0
End If                                ‘Akhir dari pernyataan kondisional.
Loop                                  ‘Mengakhiri Perintah
```

2. Program java sistem kendali *interface* untuk input

```
package edu.mit.media.amarino.multicolorlamp;
import android.app.Activity;
public class MultiColorLampInterface extends Activity implements OnClickListener {

        private static final String TAG = "MultiColorLamp";

        public static String DEVICE_ADDRESS;

        EditText idField;
        Button button;

   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.input);

      Log.d(TAG, "Main onStart");

   // get references to views defined in our main.xml layout file
      idField = (EditText) findViewById(R.id.deviceIDField);
      button = (Button) findViewById(R.id.okButton);
      // register listeners
      button.setOnClickListener(this);
      SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(this);
 DEVICE_ADDRESS = prefs.getString("device", "00:06:66:42:21:D6");
      idField.setText(DEVICE_ADDRESS);
   }


        public void onClick(View v)
        {
                DEVICE_ADDRESS = idField.getText().toString();
                PreferenceManager.getDefaultSharedPreferences(this)
                        .edit()
                                .putString("device", DEVICE_ADDRESS)
                        .commit();
                Amarino.connect(this, DEVICE_ADDRESS);
                Intent i = new Intent(this, MultiColorLamp.class);
        startActivity(i);
        }

}
```

3. Program java sistem kendali *interface* untuk ruang kendali

```
package edu.mit.media.amarino.multicolorlamp;
import android.app.Activity;
public class MultiColorLamp extends Activity implements OnSeekBarChangeListener{

        final int DELAY = 150;
        SeekBar powr;
        SeekBar volm;

        int power, volume;
        long lastChange;

   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

        //Amarino.connect(this, DEVICE_ADDRESS);
  // get references to views defined in our main.xml layout file
      powr = (SeekBar) findViewById(R.id.SeekBarPower);
      volm = (SeekBar) findViewById(R.id.SeekBarVolume);

      // register listeners
      powr.setOnSeekBarChangeListener(this);
      volm.setOnSeekBarChangeListener(this);
}

        @Override
        protected void onStart() {
                super.onStart();

         // load last state
      SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(this);
      powr = prefs.getInt("power", 0);
      volm = prefs.getInt("volume", 0);

      // set seekbars and feedback color according to last state
      powr.setProgress(power);
      volm.setProgress(volume);

        }

        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
// do not send to many updates, Arduino can't handle so much
                if (System.currentTimeMillis() - lastChange > DELAY ){
                        updateState(seekBar);
                        lastChange = System.currentTimeMillis();
                }
```

```
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
                lastChange = System.currentTimeMillis();
        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
                updateState(seekBar);
        }

        private void updateState(final SeekBar seekBar) {

                switch (seekBar.getId()){
                        case R.id.SeekBarPower:
                                power = seekBar.getProgress();
                                updatepower();
                                break;
                        case R.id.SeekBarVolume:
                                volume = seekBar.getProgress();
                                updatevolume();
                                break;
                }
                // provide user feedback
                colorIndicator.setBackgroundColor(Color.rgb(red, green, blue));
        }

        private void updatepower() {

        Amarino.sendDataToArduino(this, MultiColorLampInterface.DEVICE_ADDRESS, 'r',
power);
        }

        private void updatevolume(){
                Amarino.sendDataToArduino(this, MultiColorLampInterface.DEVICE_ADDRESS,
'g', volume);
        }

}
```

4. Program Input.xml Sistem Kendali *Interface* untuk Set Device ID

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
android:layout_width="fill_parent"
  android:layout_height="fill_parent"
        android:background="#ffffff"
        >

        <EditText android:id="@+id/deviceIDField"
```

```
                        android:layout_width="fill_parent"
                        android:layout_height="wrap_content"
                        >
        </EditText>

         <Button
                android:id="@+id/okButton"
           android:layout_width="fill_parent"
           android:layout_height="wrap_content"
           android:text="Set Device ID"
   />
<ImageView__
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
android:layout_gravity="center"
android:background="#fff"
   android:src="@drawable/amarino"/>

<ImageView__
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
android:layout_gravity="center"
android:background="#fff"
   android:src="@drawable/buildcircuit"/>
</LinearLayout>
```

5. Program Input.xml Sistem Kendali *Interface* untuk ruang Kontrol

```
        <?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="#cdcdcd">

        <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="2"
                android:clipToPadding="false"
                android:gravity="center"
                android:orientation="vertical">


                <LinearLayout
                        android:layout_width="fill_parent"
                        android:layout_height="0dp"
                        android:layout_weight="1"
                        android:background="#f00"
                        android:paddingRight="10dp">
```

```
        <TextView
            android:layout_width="101dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_weight="0.15”
            android:gravity="center"
            android:text="Power"

android:textAppearance=”?andoid:attr/textAppearanceLarge”
            android:textColor=”#000” >


        </TextView>

        <SeekBar
                android:id="@+id/SeekBarPower"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:marginLeft=”5dp”
                        android:marginLeft=”5dp”
                android:layout_weight="0.12"
                android:max="255"
                android:minHeight="50dp"

android:progressDrawable="@drawable/progress_horizontal"
                android:thumb="@drawable/btn_square_overlay_normal"
                android:thumbOffset="4px">
        </SeekBar>
</LinearLayout>

<LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#fff"
        android:paddingRight="10dp">

        <TextView
                android:layout_widht="98dp"
                android:layout_height="50dp"
                android:layout_gravity="center"
                android:layout_weight="0.26"
                android:gravity="center"
                android:text="volume">

android:textAppearance=”?andoid:attr/textAppearanceLarge”
                android:textColor=”#000” >

        </TextView>

        <SeekBar android:id="@+id/SeekBarvolume"
```

```
                android:layout_width="55dp"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:marginLeft=”5dp”
                        android:marginLeft=”5dp”
                                android:layout_weight="0.09"
                android:max="255"
                android:minHeight="50dp"

        android:progressDrawable="@drawable/progress_horizontal"
                android:thumb="@drawable/btn_square_overlay_normal"
                android:thumbOffset="4px"
            </SeekBar>
        </LinearLayout>
</LinearLayout>
```

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
JalanSrijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polsri.ac.id **E-mail :** info@polsri.ac.id

REKOMENDASI UJIAN LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama	:	NurlistiaPutri
NIM	:	061330330999
Jurusan	:	Teknik Elektro
Program Studi	:	Teknik Telekomunikasi
Judul	:	RancangBangun Audio Amplifier Stereo Berbasis Android (Sub Bahasan : Aplikasi Android, Bluetooth)

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Akhir (LA) pada Tahun Akademik 2015/2016

Palembang,

DosenPembimbing 1

M. ZakuanAgung,S.T.
NIP. 196909291993031004

DosenPembimbingII

Hj. LindawatiS.T. ,M.Kom.
NIP. 197105282006042001

Palembang, Juni 2016

Kepada
Yth. Kepala Laboraturium
Teknik Telekomunikasi
Di
Tempat

Saya yang bertanda tangan dibawah ini:

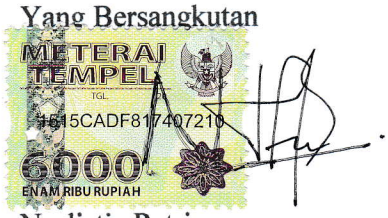
Nama	: Nurlistia Putri
NIM	: 061330330999
Kelas	: 6 TD
Judul	: Rancang Bangun Audio Amplifier Stereo Berbasis Android (Sub : Aplikasi Android, Bluetooth)
Dosen Pembimbing 1	: M. Zakuan Agung, S.T.
Dosen Pembimbing 2	: Hj. Lindawati, S.T., M.Kom

Dengan ini mengajukan permohonan untuk menggunakan laboraturium serta meminjam beberapa peralatan praktikum yang tersedia di Laboraturium. Adapun peralatan yang akan digunakan yaitu :

1. Osiloskop Digital	1 buah
2. Multimeter Digital	1 buah
3. Tetronix	1 buah
4. Kabel BNC to Buaya	1 buah
5. Adaptor	1 buah
6. Kabel jepit buaya	2 buah
7. Obeng	1 buah

Peralatan tersebut digunakan untuk kepentingan pengambilan data serta syarat untk menyelesaikan Laporan Akhir jurusan Teknik Elektro Program Teknik Telekomunikasi. Untuk kepentingan pengambilan data ini mohon kesediaan Bapak/Ibu Pembimbing untuk mendampingi. Demikianlah permohonan ini saya buat sebenar – benarnya. Atas perhatiannya saya ucapkan terima kasih.

Yang Bersangkutan



Nurlistia Putri
NIM.061330330999

No.	Tanggal	Pembimbing I	Pembimbing II	Keterangan
1.	21 - Juni 2016	M. Zakuan Agung S.T		Test alat & pengukuran
2.	19 - Juli - 2016			Test alat & pengukuran
3.	25 - Juli - 2016			Pengukuran alat
4.				
5.				
6.				
7.				
8.				
9.				
10.				
11.				
12.				
13.				

Mengetahui,

Ketua Program Studi

Ciksadan, S.T., M.Kom
NIP. 196809071993031003

Kepala Lab. Telekomunikasi

Sopian Soim, S.T., M.T
NIP.197103142001121001

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polsri.ac.id **E-mail :** info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Nurlistia Putri
NIM : 061330330999
Jurusan/Program Studi : Teknik Elektro / Teknik Telekomunikasi
Judul Laporan Akhir : Rancang Bangun Audio Amplifier Stereo Berbasis Android (Sub Bahasan : Android, Bluetooth)
Pembimbing I : Hj. Lindawati, S.T.,M.Kom

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	27-4-2016	Revisi Bab I	
2.	18-5-2016	ACC Bab I	
3.	22-6-2016	Revisi Bab II & Bab III	
4.	13-7-2016	ACC Bab II & III	
5.	20-7-2016	Revisi Bab IV	
6.	25-4-2016	ACC Bab IV	
7.	25-7-2016	Revisi Bab V	

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
8.	26-7-2016	ACC Bab V	
9.	26-7-2016	Lengkapi daftpus, abstrak, daftisi, lampiran	
10.	28-7-2016	ACC laporan LA	
11.	28-7-2016	ACC sidang	
12.			

Palembang, 28-07-2016

Ketua Program Studi DIII
Telekomunikasi

Ciksadan, S.T., M.Kom
NIP. 196809071993031003

Catatan:
*) melingkari angka yang sesuai.
Ketua Jurusan/Ketua Program Studi harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polsri.ac.id **E-mail :** info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Lembar : 1

Nama : Nurlistia Putri
NIM : 061330330999
Jurusan/Program Studi : Teknik Elektro / Teknik Telekomunikasi
Judul Laporan Akhir : Rancang Bangun Audio Amplifier Stereo Berbasis Android (Sub Bahasan : Android, Bluetooth)
Pembimbing I : M. Zakuan Agung S.T

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	21-3-2016	Proposal acc (judul), lgt lnjt bab I, II & III.	
2.	17-5-2016	acc bab I, II, III. lgt lnjt bab	
3.	7-6-2016	lgt lnjt bab IV perbaiki data	
4.	21-6-2016	Pengambilan data 1	
5.	19-7-2016	Perbaiki data, lanjutkan pd bab IV (flowchart).	
6.	25-7-2016	acc bab IV, lgt lnjt bab V.	
7.	26-7-2016	Langsung ke power point. Siapkan untuk sidang. acc bab V	

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
8.	28-7-2016	Persiapan sidang. acc bab V.	
9.			
10.			
11.			
12.			

Palembang, 20-07-2016

Ketua Program Studi DIII
Telekomunikasi

Ciksadan, S.T., M.Kom
NIP. 196809071993031003

Catatan:
*) melingkari angka yang sesuai.
Ketua Jurusan/Ketua Program Studi harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.

K EMENTERIAN RISET,TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama	: Nurlistia Putri
NIM	: 061330330999
Jurusan	: Teknik Elektro
Program Studi	: Teknik Telekomunikasi

Pihak Kedua

Nama	: Hj. Lindawati, S.T.,M.Kom
NIP	: 197105282006042001
Jurusan	: Teknik Elektro
Program Studi	: Teknik Telekomunikasi

Pada hari ini .....Senin..... tanggal .....7 maret 2016..... telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari .....Rabu.../.Jum'at... pukul .....19:00....., tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Palembang, 7 Maret 2016

Pihak Pertama,

Nurlistia Putri
NIM 061330330999

Pihak Kedua,

Hj. Lindawati, S.T.,M.Kom
NIP 197105282006042001

Mengetahui,
Ketua Jurusan

Yudi Wijanarko S.T.,M.T
NIP. 196705111992031003

KEMENTERIAN RISET,TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama	: Nurlistia Putri
NIM	: 061330330999
Jurusan	: Teknik Elektro
Program Studi	: Teknik Telekomunikasi

Pihak Kedua

Nama	: M. Zakuan Agung,S.T.
NIP	: 196909291993031004
Jurusan	: Teknik Elektro
Program Studi	: Teknik Telekomunikasi

Pada hari ini ..... Senin ..... tanggal ..... 7 maret 2016 ..... telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari ..... Senin / Selasa ..... pukul ..... 09:00-10:00 ..... tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Palembang, 7 Maret 2016

Pihak Pertama,

Nurlistia Putri
NIM 061330330999

Pihak Kedua,

M. Zakuan Agung,S.T
NIP. 196909291993031004

Mengetahui,
Ketua Jurusan

Yudi Wijanarko S.T.,M.T
NIP. 196705111992031003

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id

BUKTI PENYERAHAN HASIL KARYA/RANCANG BANGUN

Pada hari ini Selasa tanggal 23 bulan Agustus tahun 2016 telah diserahkan seperangkat karya/rancang bangun kepada Jurusan Teknik Elektro Program Studi Teknik Telekomunikasi di Politeknik Negeri Sriwijaya,

Nama Perangkat	Spesifikasi
Rancang Bangun Audio Amplifier Stereo Berbasis Android	Speaker : 3 Buah Mikrokontroller : ATMega 16 LCD : M1632 16 x 2 Sumber Tegangan : 220AC Tegangan DC : 18 Volt dan 5 Volt DC Smartphone Android : 2 Buah

Hasil karya/rancang bangun dari,

Nama	NIM	Nama Pembimbing
Nurlistia Putri	061330330999	M. Zakuan Agung, S.T
		Hj. Lindawati, S.T.,M.Kom
Fathia Khairani	061330330271	Nasron, S.T.,M.T
		M. Zakuan Agung, S.T

Palembang, 23 Agustus 2016

Yang menerima *),

M. Zakuan Agung, S.T
NIP 196909291993031004

Yang menyerahkan **),

Nurlistia Putri
NIM 061330330999

Mengetahui,
Ketua Program Studi D-III

Ciksadan, S.T., M.Kom
NIP 196809071993031003

*) pejabat Jurusan/PS yang ditunjuk (Kepala Lab./Bengkel atau Kepala Seksi)
**) perwakilan mahasiswa dari pembuat karya/rancang bangun.

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id **E-mail :** info@polsri.ac.id

PELAKSANAAN REVISI LAPORAN AKHIR

Mahasiswa berikut,

Nama : Nurlistia Putri

NIM : 061330330999

Jurusan/Program Studi : Teknik Elektro / Teknik Telekomunikasi

Judul Laporan Akhir : Rancang Bangun Audio Amplifier Stereo Berbasis Android

Telah melaksanakan revisi terhadap Laporan Akhir yang diujikan pada hari Jumat tanggal 05 bulan Agustus tahun 2016. Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi:

No.	Komentar	Nama Dosen Penguji	Tanggal	Tanda Tangan
1.	Sudah direvisi	Ir. Ali Nurdin, M.T NIP. 196212071991031001	23/8 16	
2.	Acc	Irawan Hadi, S.T.,M.Kom NIP. 196511051990031002	9/8 16	
3.	Acc	Irma Salamah, S.T.,M.T.I NIP. 197410221998022001	9/8 2016	
4.	Acc	Sholihin, S.T.,M.T NIP. 197404252001121001	8/8 2016	
5.	ACC	M. Zakuan Agung, S.T NIP. 196909291993031004	11/8 2016	
6.	Sudah direvisi	Aryanti, S.T.,M.Kom NIP. 197708092002122002	10/8 2016	

Palembang, Agustus 2016

Ketua Penguji

M. Zakuan Agung, S.T
NIP. 196909291993031004