LAMPIRAN

CARA PENGGUNAAN, PROGRAM, DAN DATASHEET

LAMPIRAN 1

CARA PENGGUNAAN PENITI'S

Perlu dilakukan *setting* pada pengendali PENITI's baik pengendali atas maupun pengendali bawah. Hal tersebut perlu dilakukan agar PENITI's dapat berjalan dengan lancer. Adapun *setting* yang perlu dilakukan adalah :

- Pada Pengendali Atas :
  - Hubungkan *Battery* pin sisir pada pin 0 dengan pin negatif (-) dan pin 7 Volt dengan pin positif (+) pada modul regulator L2596
  - Hubungkan *Battery* pin XT60 pada pin XT60 pada modul L298N
  - Hidupkan saklar pada box pengendali atas
  - Tekan Reset (jika program berhenti)

- Pada Pengendali Bawah :
  - Hubungkan *jack* pada pengendali utama pada terminal AC 220 Volt
  - Hidupkan saklar pada box pengendali utama
  - Tekan Reset (jika program berhenti)

- Pada Subjek
  - Subjek yang dihitung nilai IMT menaiki timbangan modifikasi dengan badan tegap
  - PENITI's akan otomatis mengukur tinggi badan, massa tubuh, nilai IMT, massa yang harus dikurangi, indikasi nilai massa subjek sebagai hasil dari pengukuran PENITI's
  - Subjek dapat melihat hasil pengukuran pada LCD box pengendali utama
  - Tekan Reset (jika program berhenti)

LAMPIRAN 2

```
/*****************************************************
This program was created by the
CodeWizardAVR V3.12 Advanced
Automatic Program Generator
© Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project         : PENITI’s (Pengendali Atas)
Version         :
Date            : 24-Dec-2016
Author          : Vendy Dwi Hendra Nugraha
Company         :
Comments        :


Chip type                   : ATmega16
Program type                : Application
AVR Core Clock frequency    : 16.000000 MHz
Memory model                : Small
External RAM size           : 0
Data Stack size             : 256
*****************************************************/

#include <mega16.h>
#include <stdio.h>
#include <delay.h>

// Alphanumeric LCD functions
#include <alcd.h>

//PING))
#define DDR_PING    DDRD.7          //DDR yang dijadikan output dari sensor
PING yaitu DDRD.7
#define PORT_PING   PORTD.7         //PORT yang dijadikan output dari sensor
PING yaitu PORTD 7
#define PIN_PING    PIND.7          //PIN yang dijadikan input dari sensor PING
yaitu PIND 7
unsigned char kata[33];
float jarak_PING;
float jarak_PING2;
```

```
//HC-SR04
#define TRIGGER   PORTD.2
#define ECHO      PIND.3
unsigned char baris_2[33];
long timer = 0, xTimer = 0;
float pulsa, jarakb, jarak_HCSR04;
// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
  xTimer++;
}

// Declare your global variables here

void bacaJarakPING()
{
unsigned int count=0;
DDR_PING=1;     //jadikan PIN output
PORT_PING=1;    // memberi sinyal high selama 5 us
delay_us(5);
PORT_PING=0;   //memberi sinyal low
DDR_PING=0;    //jadikan PIN sebagai input
PORT_PING=1;   //aktifkan internal pullup

while (PIN_PING==0){}         //ketika sinyal low tidak ada perintah
while (PIN_PING==1)           //ketika sinyal high maka nilai counter mencacah
naik setiap 1 us
   {
   count++;
   delay_us(1);
   }

jarak_PING2=((float)count/29.034/2);   //hitung nilai count dan dikalibrasi
menjadi jarak dalam cm
jarak_PING=jarak_PING2+120; // kalibrasi
}

void bacaJarakHCSR04()
{
  TRIGGER = 1;         // Memberikan pulsa high selama 10 microseconds
  delay_us(10);
  TRIGGER = 0;
  while(ECHO == 0);    // Tunggu pulsa high dari sensor
```

```
  TCNT1 = 0;           // Nolkan timer
  xTimer = 0;          // Nolkan variabel xTimer
  while(ECHO == 1);    // Tunggu pulsa low dari sensor
  timer = TCNT1;       // Ambil data dari timer
  pulsa = (float) xTimer * 65535 * 0.5 + (float) timer * 0.5;   // Menghitung pulsa yang masuk dalam satuan microseconds
  jarakb = (float) pulsa / 29.034 / 2;   // Menghitung jarak dalam satuan centimeter
  jarak_HCSR04 = jarakb - 1; // kalibrasi
}

void Tampilan_LCD()
{
   bacaJarakPING();
   bacaJarakHCSR04();
   lcd_gotoxy(0,0);
   lcd_puts("PING  : ");
   lcd_gotoxy(7,0);
   sprintf(kata,"%.1f ", jarak_PING);
   lcd_puts(kata);
   lcd_gotoxy(12,0);
   lcd_puts("cm");
   lcd_gotoxy(0,1);
   lcd_puts("HCSR04: ");
   lcd_gotoxy(7,1);
   sprintf(baris_2,"%.1f ", jarak_HCSR04);
   lcd_puts(baris_2);
   lcd_gotoxy(12,1);
   lcd_puts("cm");
   delay_ms(500);
}
void kirim_udr(float kar);
void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=Out Bit0=Out
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (1<<DDA1) | (1<<DDA0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
```

```
// Port B initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(0<<DDB2) | (0<<DDB1) | (0<<DDB0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=P Bit0=T
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (1<<PORTB1) | (0<<PORTB0);

// Port C initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) |
(1<<DDC2) | (1<<DDC1) | (1<<DDC0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=Out Bit0=In
DDRD=(0<<DDD7) | (1<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) |
(1<<DDD2) | (1<<DDD1) | (0<<DDD0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=P Bit2=0 Bit1=T Bit0=P
PORTD=(0<<PORTD7) | (0<<PORTD6) | (1<<PORTD5) | (0<<PORTD4) |
(1<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (1<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 2000.000 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
```

```
// Input Capture on Rising Edge
// Timer Period: 32.768 ms
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (1<<ICES1) | (0<<WGM13) | (0<<WGM12) |
(0<<CS12) | (1<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) |
(0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (1<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 19200
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) |
(0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) |
(1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x26;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) |
(0<<TWIE);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
```

```
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);
lcd_clear();
while (1)
     {
      // Place your code here
        Tampilan_LCD();

        //motor DC
        if (jarak_HCSR04 <=40 && PORTB.0==1)
         {
           PORTA.0=0;
           PORTA.1=1;
         }

        if (jarak_HCSR04 >40 && PORTB.0==1 || jarak_PING ==125)
        {
           PORTA.0=0;
           PORTA.1=0;
        }

        if (jarak_HCSR04 >40 && PORTB.0==0)
         {
           PORTA.0=1;
           PORTA.1=0;
         }



      if (PIND.0==1)
      {
      lcd_gotoxy(15,0);
      lcd_puts("1");
      }
      else
      {
      lcd_gotoxy(15,0);
      lcd_puts("0");
```

```
      }
      kirim_udr(jarak_PING);
     }
}
void kirim_udr(float kar)
        {
          while (!(UCSRA&0x20))
          {
          };
          UDR =kar;
        }
```

LAMPIRAN 3

```
/*****************************************************
This program was created by the
CodeWizardAVR V3.12 Advanced
Automatic Program Generator
© Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project         : PENITI's (Pengendali Utama)
Version         :
Date            : 07-Jan-2017
Author          : Vendy Dwi Hendra Nugraha
Company         :
Comments        :


Chip type                 : ATmega16
Program type              : Application
AVR Core Clock frequency  : 16.000000 MHz
Memory model              : Small
External RAM size         : 0
Data Stack size           : 256
*****************************************************/

#include <mega16.h>
#include <stdio.h>
#include <delay.h>
#include <math.h>


// Alphanumeric LCD functions
#include <alcd.h>
int a, b, c, e, f;

// Declare your global variables here
char lcd_buffer[33];
float kar;
void terima_data(void);

#define HX711_SCK PORTB.7
#define HX711_DT PINB.6
#define HIGH 1
#define LOW 0
```

```
long HX711_Buffer = 0;
int Weight_Maopi = 0;
int Weight = 0;

unsigned long HX711_Read(void)
{
	unsigned long count;
	unsigned char i;

	HX711_DT= HIGH;

	delay_us(1);

	HX711_SCK=LOW;
	delay_us(1);

	count=0;
	while(HX711_DT);
	for(i=0;i<24;i++)
	{
		HX711_SCK=HIGH;
		delay_us(1);
		count=count<<1;
		HX711_SCK= LOW;
		delay_us(1);
		if (HX711_DT)
		count++;
	}
	HX711_SCK= HIGH;
	count ^= 0x800000;
	delay_us(1);
	HX711_SCK= LOW;
	delay_us(1);
	return(count);
}

void Get_Maopi()
{
	HX711_Buffer = HX711_Read();
	Weight_Maopi = ((HX711_Buffer/100000)-82);
	Weight = (Weight_Maopi + 0.1707) / 0.2219;
}
```

```
void Tampilan_LCD()
{
	terima_data();
	lcd_gotoxy(0,0);
	lcd_puts("Tinggi: ");
	lcd_gotoxy(8,0);
	sprintf(lcd_buffer, "%.1f",kar);
	lcd_puts(lcd_buffer);
	lcd_gotoxy(14,0);
	lcd_puts("cm");
	lcd_gotoxy(0,1);
	lcd_puts("Massa : ");
	lcd_gotoxy(8,1);
	sprintf(lcd_buffer,"%d ",Weight);
	lcd_puts(lcd_buffer);
	lcd_gotoxy(14,1);
	lcd_puts("kg");
}

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (0<<DDA1) | (0<<DDA0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization
// Function: Bit7=Out Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=Out
DDRB=(1<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(0<<DDB2) | (0<<DDB1) | (1 <<DDB0);
// State: Bit7=0 Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTB=(0<<PORTB7) | (1<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
```

```
// Port C initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) |
(1<<DDC2) | (1<<DDC1) | (1<<DDC0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=Out Bit0=In
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) |
(0<<DDD2) | (1<<DDD1) | (0<<DDD0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) |
(0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) |
(0<<CS12) | (0<<CS11) | (0<<CS10);
```

```
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) |
(0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 19200
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) |
(0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) |
(1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
```

```
UBRRH=0x00;
UBRRL=0x26;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) |
(0<<TWIE);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(20);
Get_Maopi();
```

```
while (1)
      {
      // Place your code here
       a=Weight/(kar*kar/10000) ;
       f=fmod(Weight,kar*kar/100000);
       lcd_gotoxy(0,2);
       lcd_puts("IMT   : ");
       lcd_gotoxy(8,2);
       sprintf(lcd_buffer,"%d.%d      ",a,f);
       lcd_puts(lcd_buffer);
       lcd_gotoxy(13,2);
       lcd_puts(";");
       e=(kar*kar/10000)*21;
       b=e-Weight;
       c=fmod(e,Weight);
       lcd_gotoxy(14,2);
       sprintf(lcd_buffer,"%d.%d                  ",b,c);
       lcd_puts(lcd_buffer);


        switch (a)
        {
         case 10 :
         lcd_gotoxy(0,3);
         lcd_puts("Sangat Kurus");
         break;

         case 11 :
         lcd_gotoxy(0,3);
         lcd_puts("Sangat Kurus");
         break;

         case 12 :
         lcd_gotoxy(0,3);
         lcd_puts("Sangat Kurus");
         break;

         case 13 :
         lcd_gotoxy(0,3);
         lcd_puts("Sangat Kurus");
         break;

         case 14 :
```

```
lcd_gotoxy(0,3);
lcd_puts("Sangat Kurus");
break;

case 15 :
lcd_gotoxy(0,3);
lcd_puts("Sangat Kurus");
break;

case 16 :
lcd_gotoxy(0,3);
lcd_puts("Sangat Kurus");
break;

case 17 :
lcd_gotoxy(0,3);
lcd_puts("Kurus");
break;

case 18 :
lcd_gotoxy(0,3);
lcd_puts("Kurus");
break;

case 19 :
lcd_gotoxy(0,3);
lcd_puts("Normal");
break;

case 20 :
lcd_gotoxy(0,3);
lcd_puts("Normal");
break;

case 21 :
lcd_gotoxy(0,3);
lcd_puts("Normal");
break;

case 22 :
lcd_gotoxy(0,3);
lcd_puts("Normal");
break;
```

```
case 23 :
lcd_gotoxy(0,3);
lcd_puts("Normal");
break;

case 24 :
lcd_gotoxy(0,3);
lcd_puts("Normal");
break;

case 25 :
lcd_gotoxy(0,3);
lcd_puts("Gemuk");
break;

case 26 :
lcd_gotoxy(0,3);
lcd_puts("Gemuk");
break;

case 27 :
lcd_gotoxy(0,3);
lcd_puts("Gemuk");
break;

case 28 :
lcd_gotoxy(0,3);
lcd_puts("Gemuk");
break;

case 29 :
lcd_gotoxy(0,3);
lcd_puts("Gemuk");
break;

case 30 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas I");
break;

case 31 :
lcd_gotoxy(0,3);
```

```
lcd_puts("Obesitas I");
break;

case 32 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas I");
break;

case 33 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas I");
break;

case 34 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas I");
break;

case 35 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas II");
break;

case 36 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas II");
break;

case 37 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas II");
break;

case 38 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas II");
break;

case 39 :
lcd_gotoxy(0,3);
lcd_puts("Obesitas II");
break;
```

```
 case 40 :
 lcd_gotoxy(0,3);
 lcd_puts("Obesitas III");
 break;

 case 42 :
 lcd_gotoxy(0,3);
 lcd_puts("Obesitas III");
 break;

 case 43 :
 lcd_gotoxy(0,3);
 lcd_puts("Obesitas III");
 break;

 case 44 :
 lcd_gotoxy(0,3);
 lcd_puts("Obesitas III");
 break;

 case 45 :
 lcd_gotoxy(0,3);
 lcd_puts("Obesitas III");
 break;

    default:
    lcd_gotoxy(0,3);
    lcd_puts("PENITI's");
 };


if(Weight>30)
{
    lcd_gotoxy(19,0);
    lcd_puts("1");
    PORTD.1=1;
}
 else
 {
   lcd_gotoxy(19,0);
   lcd_puts("0");
   PORTD.1=0;
 }
```

```
        Get_Maopi();
        Tampilan_LCD();
        delay_ms(500);

      }
}

void terima_data(void)
{
    while (!(UCSRA&0x80))
    {
    };
    kar = UDR;
}
```

LAMPIRAN 4

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 16K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 512 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 1K Byte Internal SRAM
  - 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 MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega16L
  - 4.5 - 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

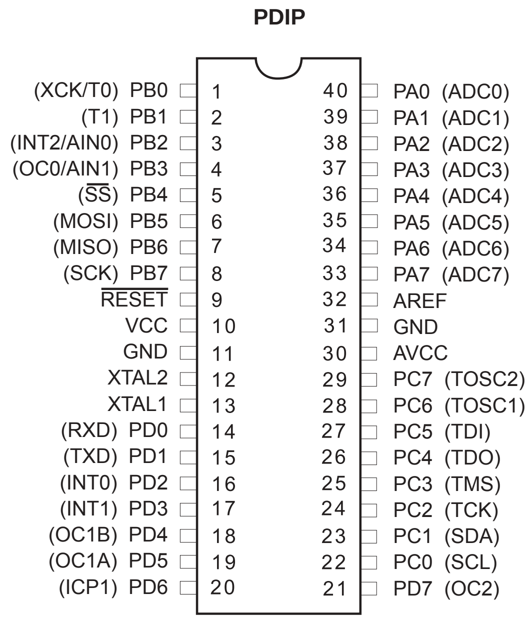
Summary

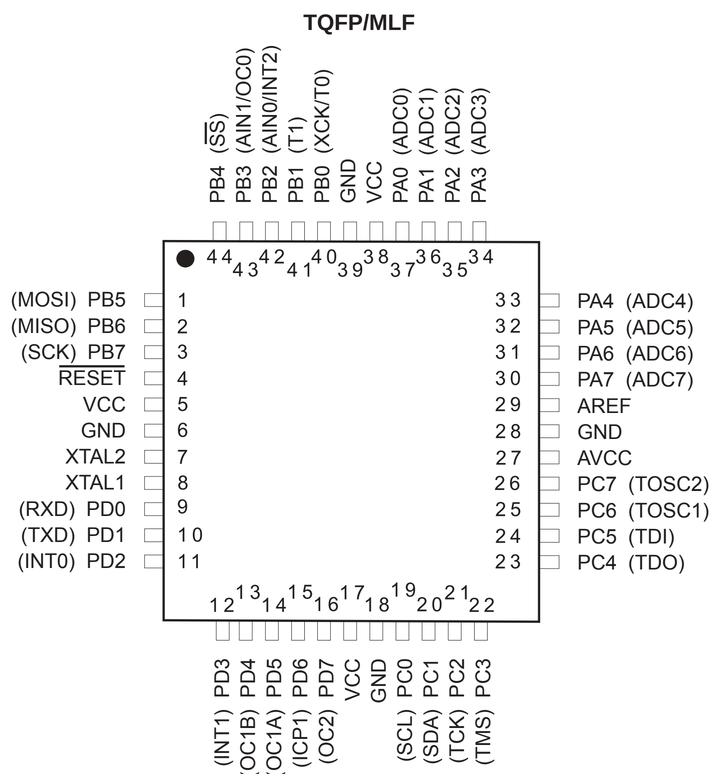
2466HS–AVR–12/03

Note: This is a summary document. A complete document is available on our Web site at www.atmel.com.

Pin Configurations

Figure 1. Pinouts ATmega16





Disclaimer

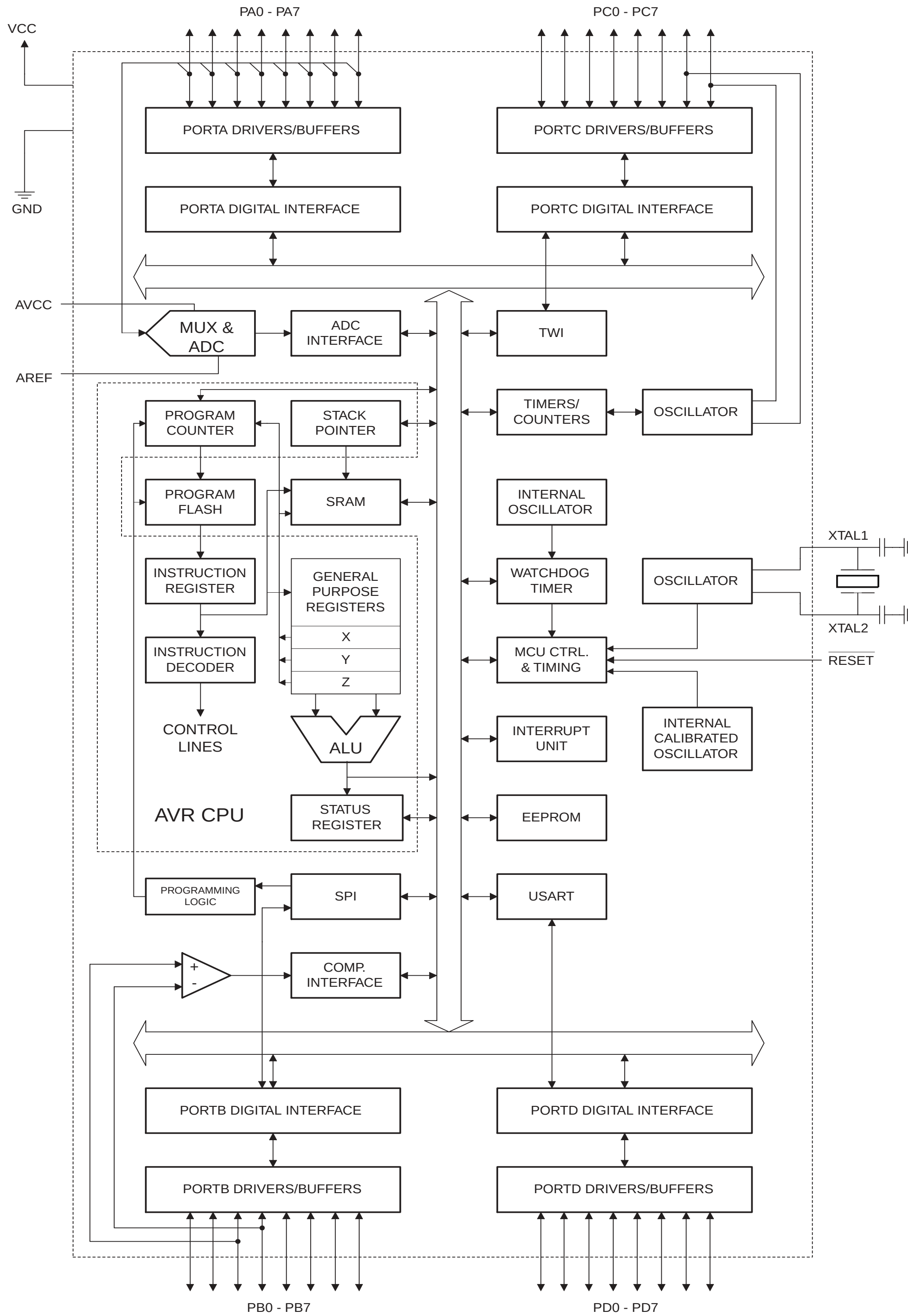
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

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: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte 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 56.

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 59.

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 61.

$\overline{\text{RESET}}$

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 36. 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.

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	7
$3E ($5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	10
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	10
$3C ($5C)	OCR0	Timer/Counter0 Output Compare Register								83
$3B ($5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	46, 67
$3A ($5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	68
$39 ($59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	83, 114, 132
$38 ($58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	84, 115, 132
$37 ($57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	249
$36 ($56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	178
$35 ($55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	30, 66
$34 ($54)	MCUCSR	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	39, 67, 229
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	81
$32 ($52)	TCNT0	Timer/Counter0 (8 Bits)								83
$31[1] ($51)[1]	OSCCAL	Oscillator Calibration Register								28
	OCDR	On-Chip Debug Register								225
$30 ($50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	55,86,133,199,219
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	109
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	112
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								113
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								113
$2B ($4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								113
$2A ($4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								113
$29 ($49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								113
$28 ($48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								113
$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	127
$24 ($44)	TCNT2	Timer/Counter2 (8 Bits)								129
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								129
$22 ($42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	130
$21 ($41)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	41
$20[2] ($40)[2]	UBRRH	URSEL	–	–	–	UBRR[11:8]				165
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	164
$1F ($3F)	EEARH	–	–	–	–	–	–	–	EEAR8	17
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								17
$1D ($3D)	EEDR	EEPROM Data Register								17
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	17
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	64
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	64
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	64
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	64
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	64
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	64
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
$0F ($2F)	SPDR	SPI Data Register								140
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	140
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	138
$0C ($2C)	UDR	USART I/O Data Register								161
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	162
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	163
$09 ($29)	UBRRL	USART Baud Rate Register Low Byte								165
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	200
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	215
$06 ($26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	217
$05 ($25)	ADCH	ADC Data Register High Byte								218
$04 ($24)	ADCL	ADC Data Register Low Byte								218
$03 ($23)	TWDR	Two-wire Serial Interface Data Register								180
$02 ($22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	180

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	179
$00 ($20)	TWBR	Two-wire Serial Interface Bit Rate Register								178

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

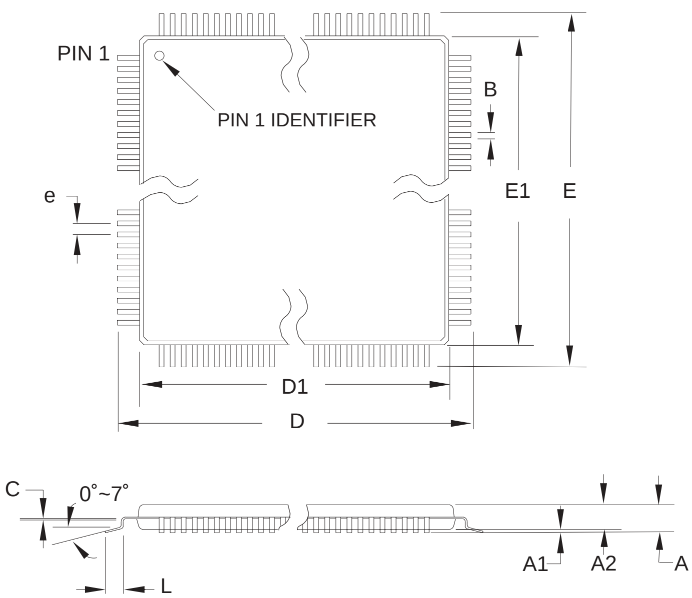
Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7 - 5.5V	ATmega16L-8AC ATmega16L-8PC ATmega16L-8MC	44A 40P6 44M1	Commercial (0^oC to 70^oC)
		ATmega16L-8AI ATmega16L-8PI ATmega16L-8MI	44A 40P6 44M1	Industrial (-40^oC to 85^oC)
16	4.5 - 5.5V	ATmega16-16AC ATmega16-16PC ATmega16-16MC	44A 40P6 44M1	Commercial (0^oC to 70^oC)
		ATmega16-16AI ATmega16-16PI ATmega16-16MI	44A 40P6 44M1	Industrial (-40^oC to 85^oC)

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 x 7 x 1.0 mm body, lead pitch 0.50 mm, Micro Lead Frame Package (MLF)

Packaging Information

44A



COMMON DIMENSIONS
(Unit of Measure = mm)

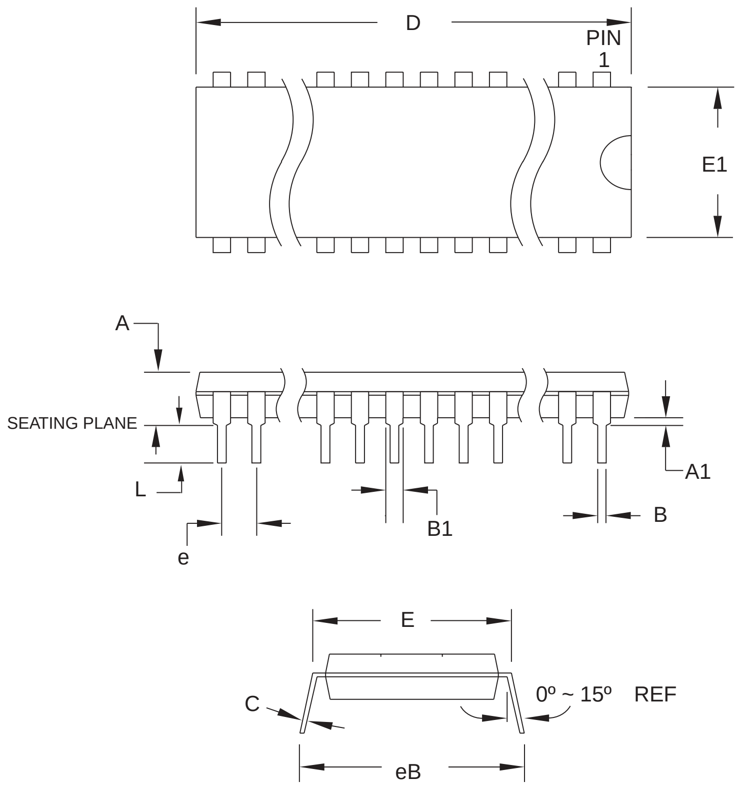
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

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.
ATMEL 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)

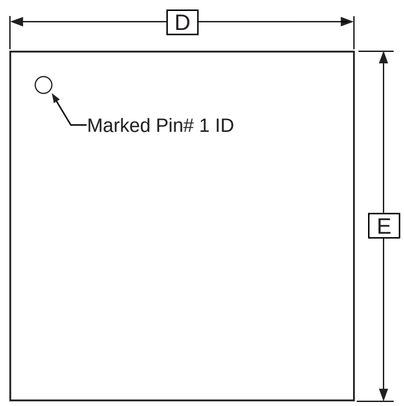
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

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").

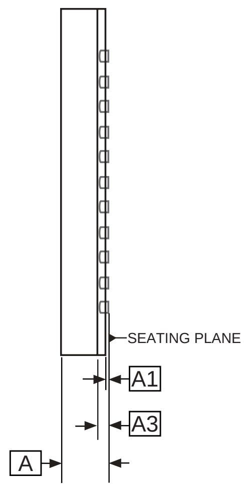
09/28/01

	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

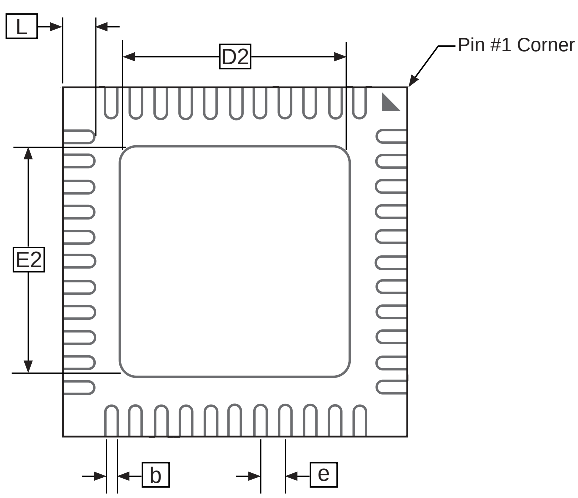
44M1



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A3		0.25 REF		
b	0.18	0.23	0.30	
D		7.00 BSC		
D2	5.00	5.20	5.40	
E		7.00 BSC		
E2	5.00	5.20	5.40	
e		0.50 BSC		
L	0.35	0.55	0.75	

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

01/15/03

	TITLE	DRAWING NO.	REV.
ATMEL 2325 Orchard Parkway San Jose, CA 95131	**44M1**, 44-pad, 7 x 7 x 1.0 mm Body, Lead Pitch 0.50 mm Micro Lead Frame Package (MLF)	44M1	C

Errata

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

ATmega16(L) Rev. I

- **IDCODE masks data from TDI input**

1. 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.

ATmega16(L) Rev. H

- **IDCODE masks data from TDI input**

1. 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.

ATmega16(L) Rev. G

- **IDCODE masks data from TDI input**

1. 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.

Datasheet Change Log for ATmega16

This section contains a log on the changes made to the datasheet for ATmega16.

Changes from Rev. 2466G-10/03 to Rev. 2466H-12/03

All page numbers refer to this document.

1. **Updated "Calibrated Internal RC Oscillator" on page 27.**

Changes from Rev. 2466F-02/03 to Rev. 2466G-10/03

All page numbers refer to this document.

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 34.**
4. **Updated assembly and C code examples in "Watchdog Timer Control Register – WDTCR" on page 41.**
5. **Updated Figure 46 on page 101.**
6. **Updated Table 15 on page 36, Table 82 on page 215 and Table 115 on page 274.**
7. **Updated "Test Access Port – TAP" on page 220 regarding JTAGEN.**
8. **Updated description for the JTD bit on page 229.**
9. **Added note 2 to Figure 126 on page 251.**
10. **Added a note regarding JTAGEN fuse to Table 105 on page 259.**
11. **Updated Absolute Maximum Ratings* and DC Characteristics in "Electrical Characteristics" on page 289.**
12. **Updated "ATmega16 Typical Characteristics" on page 297.**
13. **Fixed typo for 16 MHz MLF package in "Ordering Information" on page 11.**
14. **Added a proposal for solving problems regarding the JTAG instruction IDCODE in "Errata" on page 15.**

Changes from Rev. 2466E-10/02 to Rev. 2466F-02/03

All page numbers refer to this document.

1. **Added note about masking out unused bits when reading the Program Counter in "Stack Pointer" on page 10.**
2. **Added Chip Erase as a first step in "Programming the Flash" on page 286 and "Programming the EEPROM" on page 287.**
3. **Added the section "Unconnected pins" on page 53.**

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 252 about writing to the EEPROM during an SPM Page Load.**

8. **Removed ADHSM completely.**

9. **Added Table 73, "TWI Bit Rate Prescaler," on page 180 to describe the TWPS bits in the "TWI Status Register – TWSR" on page 179.**

10. **Added section "Default Clock Source" on page 23.**

11. **Added note about frequency variation when using an external clock. Note added in "External Clock" on page 29. An extra row and a note added in Table 118 on page 291.**

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 20.**

15. **Added note about Differential Mode with Auto Triggering in "Prescaling and Conversion Timing" on page 205.**

16. **Added updated "Packaging Information" on page 12.**

Changes from Rev. 2466D-09/02 to Rev. 2466E-10/02

All page numbers refer to this document.

1. **Updated "DC Characteristics" on page 289.**

Changes from Rev. 2466C-03/02 to Rev. 2466D-09/02

All page numbers refer to this document.

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

2. **Updated the following tables: Table 4 on page 24, Table 15 on page 36, Table 42 on page 83, Table 45 on page 110, Table 46 on page 110, Table 59 on page 141, Table 67 on page 165, Table 90 on page 233, Table 102 on page 257, "DC Characteristics" on page 289, Table 119 on page 291, Table 121 on page 293, and Table 122 on page 295.**

3. **Updated "Errata" on page 15.**

Changes from Rev. 2466B-09/01 to Rev. 2466C-03/02

All page numbers refer to this document.

1. **Updated typical EEPROM programming time, Table 1 on page 18.**

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

   Table 3 on page 23, Table 5 on page 25, Table 6 on page 26, Table 8 on page 27, Table 9 on page 27, and Table 10 on page 28.

3. **Updated Table 17 on page 41 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 36, Table 16 on page 40, Table 116 on page 272 (table removed in document review #D), "Electrical Characteristics" on page 289, Table 119 on page 291, Table 121 on page 293, and Table 122 on page 295.

5. **Updated TWI Chapter.**

   Added the note at the end of the "Bit Rate Generator Unit" on page 176.

6. **Corrected description of ADSC bit in "ADC Control and Status Register A – ADCSRA" on page 217.**

7. **Improved description on how to do a polarity check of the ADC doff results in "ADC Conversion Result" on page 214.**

8. **Added JTAG version number for rev. H in Table 87 on page 227.**

9. **Added not regarding OCDEN Fuse below Table 105 on page 259.**

10. **Updated Programming Figures:**

    Figure 127 on page 261 and Figure 136 on page 272 are updated to also reflect that AVCC must be connected during Programming mode. Figure 131 on page 268 added to illustrate how to program the fuses.

11. **Added a note regarding usage of the "PROG_PAGELOAD ($6)" on page 278 and "PROG_PAGEREAD ($7)" on page 278.**

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

    See "Leaving Programming Mode" on page 286.

13. **Added Calibrated RC Oscillator characterization curves in section "ATmega16 Typical Characteristics" on page 297.**

14. **Corrected ordering code for MLF package (16MHz) in "Ordering Information" on page 11.**

15. **Corrected Table 90, "Scan Signals for the Oscillators(1)(2)(3)," on page 233.**

Atmel Corporation

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

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

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

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

2466HS–AVR–12/03

LAMPIRAN 5

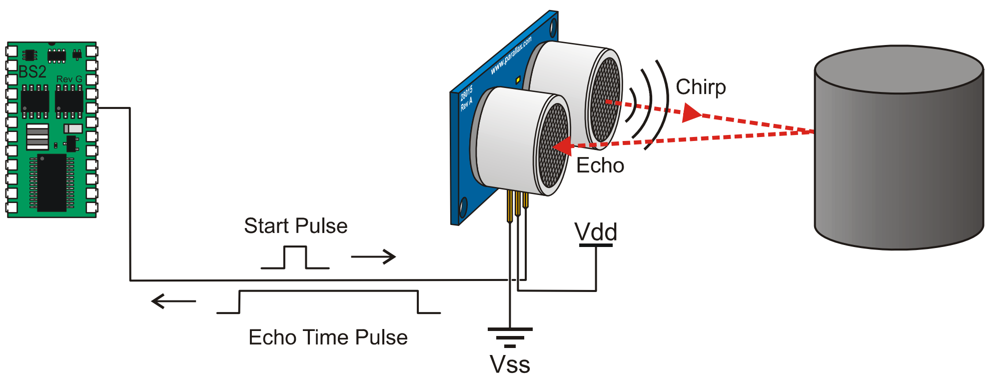
Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

PING))) Ultrasonic Distance Sensor (#28015)

The Parallax PING)))™ ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, Propeller chip, or Arduino, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2 µs min, 5 µs typ.
- Echo pulse: positive TTL pulse, 115 µs minimum to 18.5 ms maximum.
- RoHS Compliant

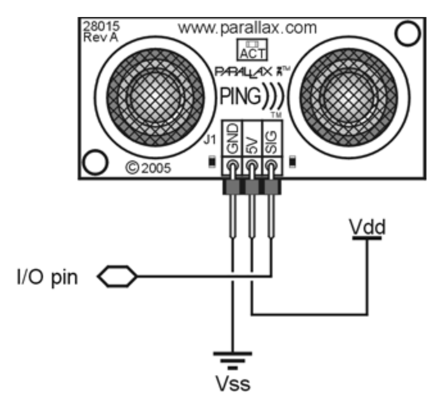
Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

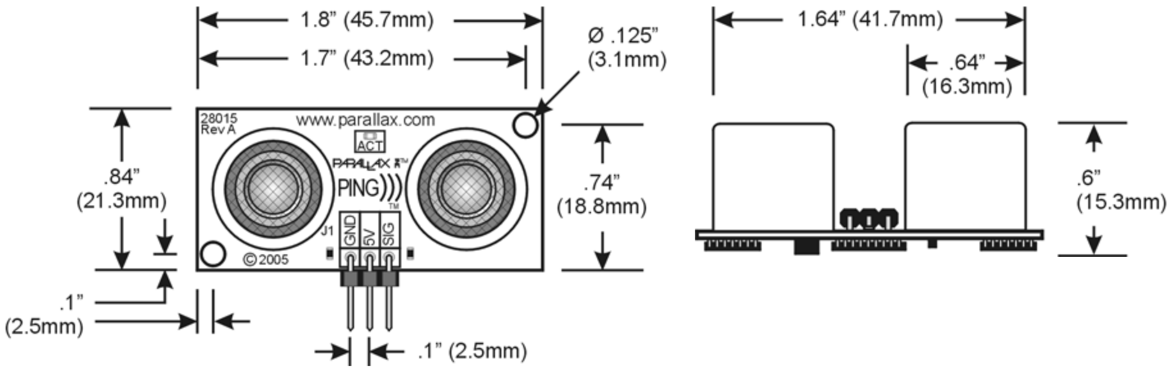
Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged into a directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #800-00120).

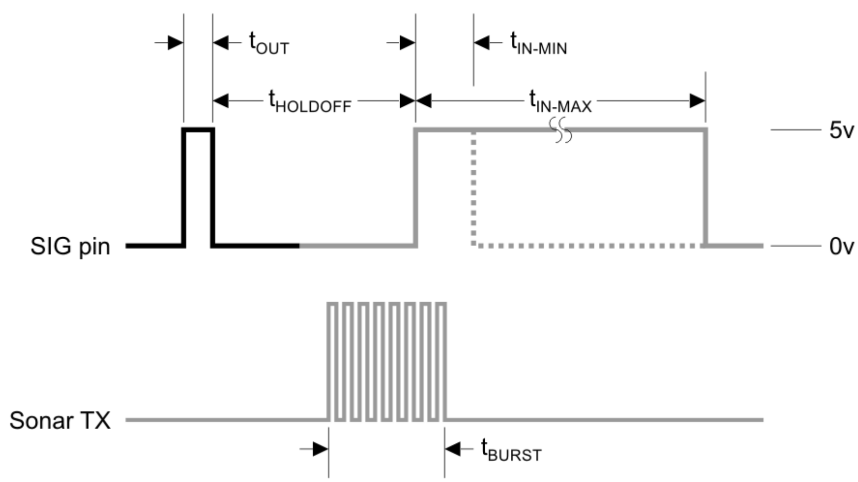


Dimensions



Communication Protocol

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.

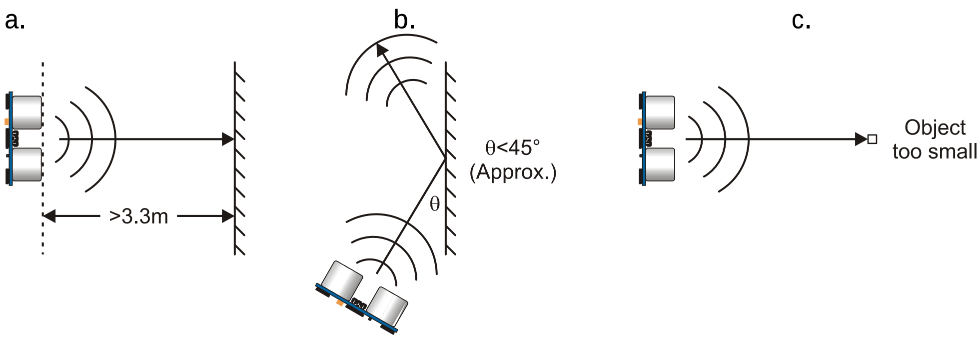


	Host Device	Input Trigger Pulse	t_{OUT}	2 µs (min), 5 µs typical
	PING))) Sensor	Echo Holdoff	$t_{HOLDOFF}$	750 µs
		Burst Frequency	t_{BURST}	200 µs @ 40 kHz
		Echo Return Pulse Minimum	t_{IN-MIN}	115 µs
		Echo Return Pulse Maximum	t_{IN-MAX}	18.5 ms
		Delay before next measurement		200 µs

Practical Considerations for Use

Object Positioning

The PING))) sensor cannot accurately measure the distance to an object that: a) is more than 3 meters away, b) that has its reflective surface at a shallow angle so that sound will not be reflected back towards the sensor, or c) is too small to reflect enough sound back to the sensor. In addition, if your PING))) sensor is mounted low on your device, you may detect sound reflecting off of the floor.



Target Object Material

In addition, objects that absorb sound or have a soft or irregular surface, such as a stuffed animal, may not reflect enough sound to be detected accurately. The PING))) sensor will detect the surface of water, however it is not rated for outdoor use or continual use in a wet environment. Condensation on its transducers may affect performance and lifespan of the device.

Air Temperature

Temperature has an effect on the speed of sound in air that is measurable by the PING))) sensor. If the temperature (°C) is known, the formula is:

$$C_{air} = 331.5 + (0.6 \times T_C) \text{ m/s}$$

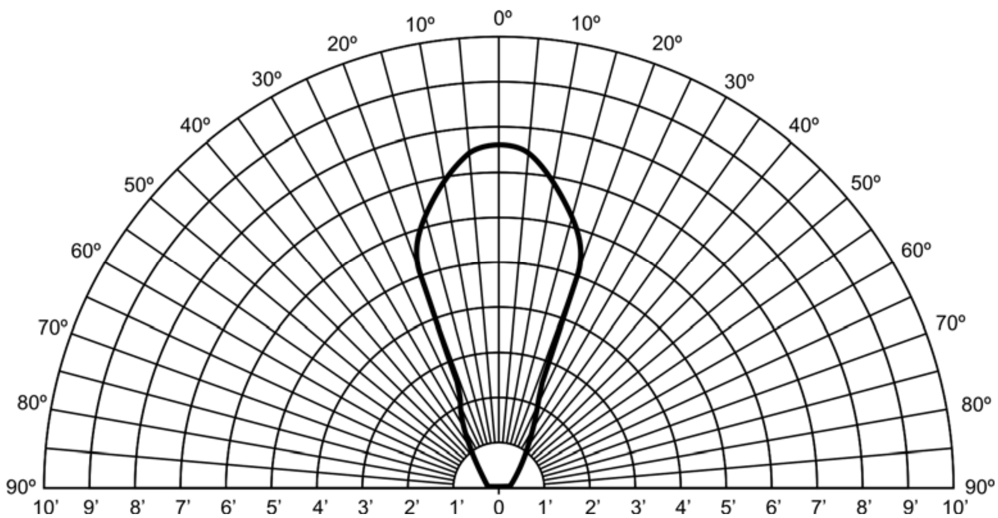
The percent error over the sensor's operating range of 0 to 70 ° C is significant, in the magnitude of 11 to 12 percent. The use of conversion constants to account for air temperature may be incorporated into your program (as is the case in the example BS2 program given in the Example Programs section below). Percent error and conversion constant calculations are introduced in Chapter 2 of *Smart Sensors and Applications*, a Stamps in Class text available for download from the 28029 product page at www.parallax.com.

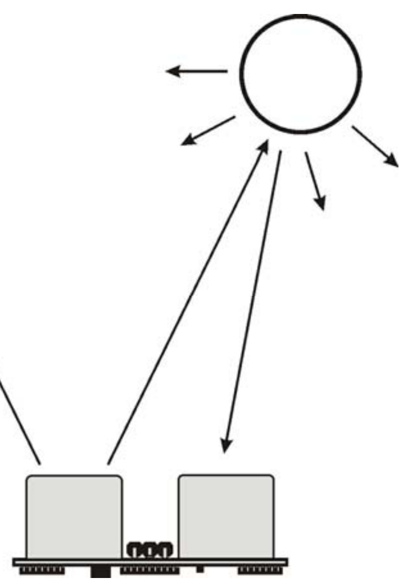
Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

Test 1

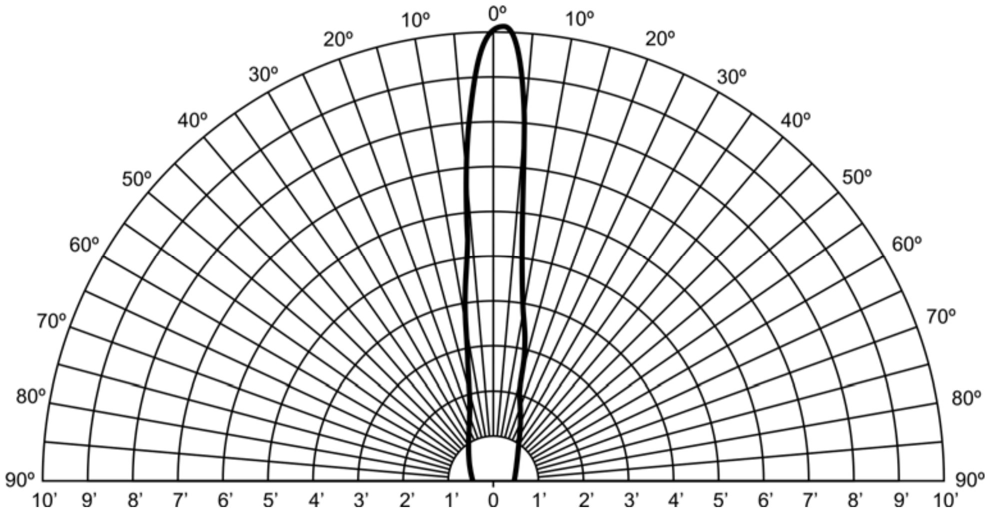
Sensor Elevation: 40 in. (101.6 cm)
Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation

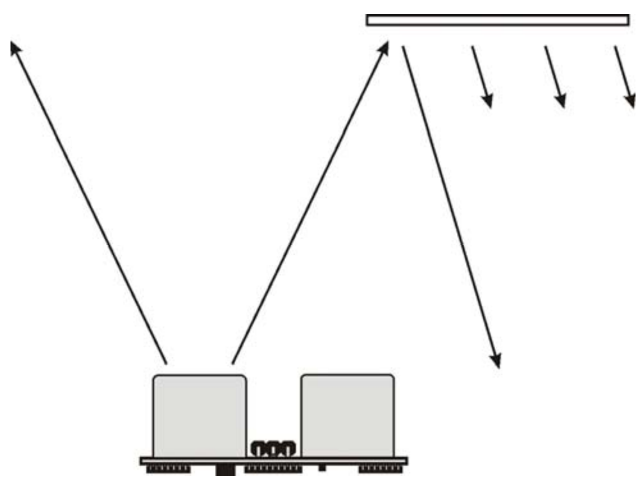




Test 2

Sensor Elevation:	40 in. (101.6 cm)
Target:	12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole Target positioned parallel to backplane of sensor

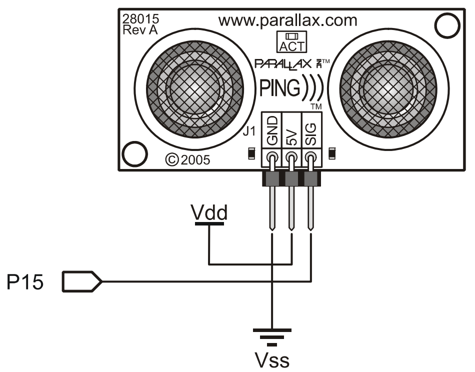


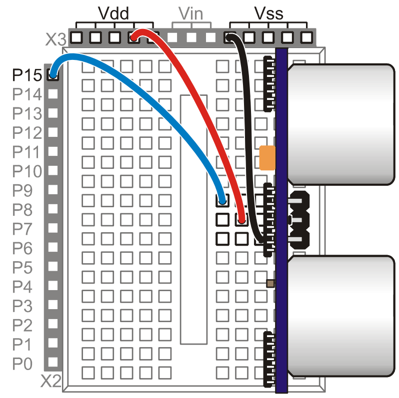


Example Programs

BASIC Stamp 2

This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp® 2 via the Board of Education® breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example BASIC Stamp program listed below.

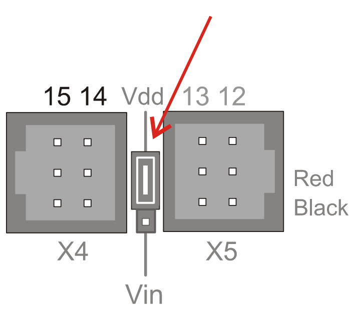




Extension Cable and Port Cautions for the Board of Education

If you are connecting your PING))) sensor to a Board of Education platform using an extension cable, follow these steps:

1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown. Then plug the cable into the port, matching the wire color to the labels next to the port.
4. If your Board of Education servo ports do not have a jumper, **do not use them with the PING))) sensor**. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the cable directly to the breadboard with a 3-pin header as shown above. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.



Board of Education Servo Port Jumper, Set to Vdd

Example Program: PingMeasureCmAndIn.bs2

This program for the BASIC Stamp 2 displays distance measurements in both inches and centimeters in the BASIC Stamp Debug Terminal. The example program can be downloaded from the 28015 product page at www.parallax.com. The BASIC Stamp Editor software, which includes the Debug Terminal, is a free download from www.parallax.com/basicstampsoftware.

```
' Smart Sensors and Applications - PingMeasureCmAndIn.bs2
' Measure distance with Ping))) sensor and display in both in & cm

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

' Conversion constants for room temperature measurements.
CmConstant    CON   2260
InConstant    CON   890

cmDistance  VAR   Word
inDistance  VAR   Word
time        VAR   Word

DO

  PULSOUT 15, 5
  PULSIN 15, 1, time

  cmDistance = cmConstant ** time
  inDistance = inConstant ** time

  DEBUG HOME, DEC3 cmDistance, " cm"
  DEBUG CR, DEC3 inDistance, " in"

  PAUSE 100

LOOP
```

Propeller Microcontroller

```
{{
*************************************
*       Ping))) Object V1.1         *
*      (C) 2006 Parallax, Inc.      *
* Author:  Chris Savage & Jeff Martin *
* Started: 05-08-2006               *
*************************************

Interface to Ping))) sensor and measure its ultrasonic travel time.  Measurements can be in
units of time or distance.  Each method requires one parameter, Pin, that is the I/O pin that
is connected to the Ping)))'s signal line.

  ┌───────────────────┐
  │┌───┐         ┌───┐│    Connection To Propeller
  ││ ▮ │ PING))) │ ▮ ││    Remember PING))) Requires
  │└───┘         └───┘│    +5V Power Supply
  │    GND +5V SIG    │
  └─────┬───┬───┬─────┘
        │  ┌┘   ≷ 1K
        ▽  └┘   └┤ Pin

--------------------------REVISION HISTORY--------------------------
 v1.1 - Updated 03/20/2007 to change SIG resistor from 10K to 1K
}}

CON
  TO_IN = 73_746                                                 ' Inches
  TO_CM = 29_034                                                 ' Centimeters

PUB Ticks(Pin) : Microseconds | cnt1, cnt2
''Return Ping)))'s one-way ultrasonic travel time in microseconds

  outa[Pin]~                                                     ' Clear I/O Pin
  dira[Pin]~~                                                    ' Make Pin Output
  outa[Pin]~~                                                    ' Set I/O Pin
  outa[Pin]~                                                     ' Clear I/O Pin (> 2 µs pulse)
  dira[Pin]~                                                     ' Make I/O Pin Input
  waitpne(0, |< Pin, 0)                                          ' Wait For Pin To Go HIGH
  cnt1 := cnt                                                    ' Store Current Counter Value
  waitpeq(0, |< Pin, 0)                                          ' Wait For Pin To Go LOW
  cnt2 := cnt                                                    ' Store New Counter Value
  Microseconds := (||(cnt1 - cnt2) / (clkfreq / 1_000_000)) >> 1    ' Return Time in µs

PUB Inches(Pin) : Distance
''Measure object distance in inches

  Distance := Ticks(Pin) * 1_000 / TO_IN                          ' Distance In Inches

PUB Centimeters(Pin) : Distance
''Measure object distance in centimeters

  Distance := Millimeters(Pin) / 10                               ' Distance In Centimeters

PUB Millimeters(Pin) : Distance
''Measure object distance in millimeters

  Distance := Ticks(Pin) * 10_000 / TO_CM                         ' Distance In Millimeters
```

The ping.spin object is used in an example project with the Parallax 4 x 20 Serial LCD (#27979) to display distance measurements. The complete Project Archive can be downloaded from the Propeller Object Exchange at http://obex.parallax.com. The Propeller Tool software can be downloaded from www.parallax.com/propellertool.

```
─────────────────────────────────────
Parallax Propeller Chip Project Archive
─────────────────────────────────────

 Project :  "ping_demo"

Archived :  Tuesday, December 18, 2007 at 3:29:46 PM

    Tool :  Propeller Tool version 1.05.8

            ping_demo.spin
              │
              ├──Debug_Lcd.spin
              │    │
              │    ├──Serial_Lcd.spin
              │    │    │
              │    │    └──Simple_Serial.spin
              │    │
              │    └──Simple_Numbers.spin
              │
              └──ping.spin
```

Reources and Downloads

For additional example code downloads and links to videos, tutorials and robotics projects that use the Ping))) Ultrasonic Distance Sensor, visit www.parallax.com and search “28015.”

Product Change Notice

Rev A: original release
Rev B: resonator added to the SX-28 co-processor circuit. No changes to functionality
Rev C: SX-28 co-processor changed to PIC16F57. No changes to functionality.

Revision History

Version 2.0: Added revision history. Removed Javelin Stamp examples. Added URLs for programming software. Added Product Change Notice section with PCB revision information.

LAMPIRAN 6

HC-SR04 User Guide

Part 1 Ultrasonic Introduction

1. 1 Ultrasonic Definition

The human ear can hear sound frequency around 20HZ ~ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ .

1.2 Ultrasonic distance measurement principle

Ultrasonic transmitter emitted an ultrasonic wave in one direction, and started timing when it launched. Ultrasonic spread in the air, and would return immediately when it encountered obstacles on the way. At last, the ultrasonic receiver would stop timing when it received the reflected wave. As Ultrasonic spread velocity is 340m / s in the air, based on the timer record **t**, we can calculate the distance (s) between the obstacle and transmitter, namely: s = 340t / 2, which is so- called time difference distance measurement principle

The principle of ultrasonic distance measurement used the already-known air spreading velocity, measuring the time from launch to reflection when it encountered obstacle, and then calculate the distance between the transmitter and the obstacle according to the time and the velocity. Thus, the principle of ultrasonic distance measurement is the same with radar.

Distance Measurement formula is expressed as: L = C X T

In the formula, L is the measured distance, and C is the ultrasonic spreading velocity in air, also, T represents time (T is half the time value from transmitting to receiving).

1.3 Ultrasonic Application

Ultrasonic Application Technology is the thing which developed in recent decades. With the ultrasonic advance, and the electronic technology development, especially as high-power semiconductor device technology matures, the application of ultrasonic has become increasingly widespread:

- Ultrasonic measurement of distance, depth and thickness;
- Ultrasonic testing;
- Ultrasound imaging;
- Ultrasonic machining, such as polishing, drilling;
- Ultrasonic cleaning;
- Ultrasonic welding;

Part 2 HC-SR04 Ultrasonic Module Introduction

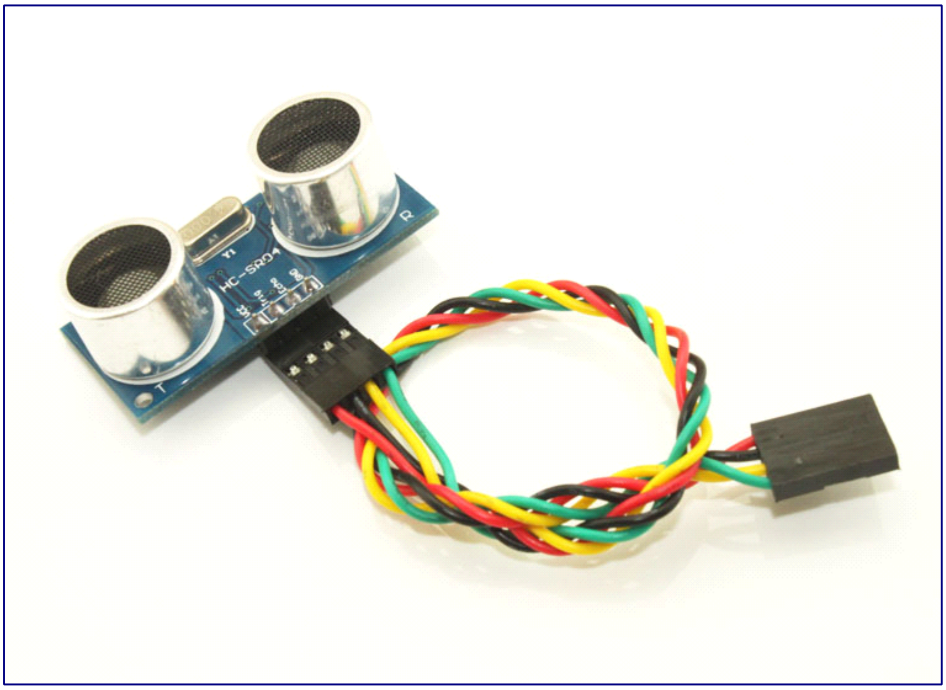
2.1 Product Features

- Stable performance
- Accurate distance measurement
- High-density
- Small blind

Application Areas:

- Robotics barrier
- Object distance measurement
- Level detection
- Public security
- Parking detection

2.2 Product Image



2.3、Module pin definitions

Types	Pin Symbol	Pin Function Description
HC-SR04	VCC	5V power supply
	Trig	Trigger pin
	Echo	Receive pin
	GND	Power ground

2.4、Electrical parameters

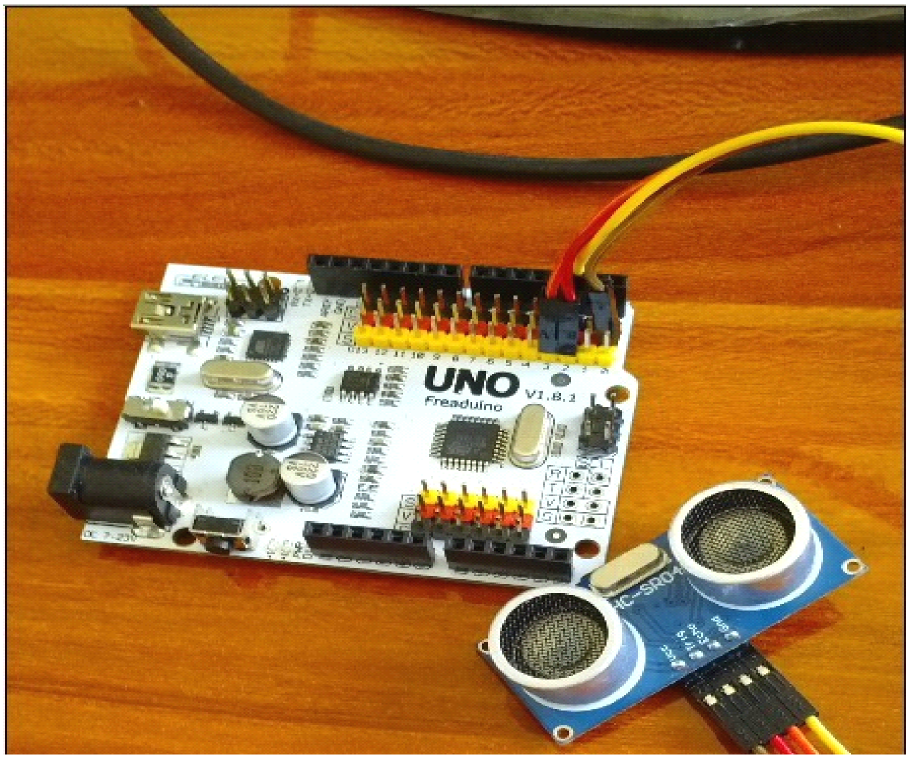
Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	DC-5V
Operating Current	15mA
Operating Frequency	40KHZ
Farthest Range	4m
Nearest Range	2cm
Measuring Angle	15 Degree
Input Trigger Signal	10us TTL pulse
Output Echo Signal	Output TTL level signal, proportional with range
Dimensions	45*20*15mm

2.5 Module operating Principle

Set low the Trig and Echo port when the module initializes , firstly, transmit at least 10us high level pulse to the Trig pin (module automatically sends eight 40K square wave), and then wait to capture the rising edge output by echo port, at the same time, open the timer to start timing. Next, once again capture the falling edge output by echo port, at the same time, read the time of the counter, which is the ultrasonic running time in the air. According to the formular: test distance = (high level time * ultrasonic spreading velocity in air) / 2, you can calculate the distance to the obstacle.

Part3 Use Freaduino UNO to test HC-SR04

3.1 Freaduino uno and HC-SR04 Connection



Connection Description：D2<------>Trig D3<------>Echo (The users can define the connection pin by themselves)

Note： You need to set the Freaduino UNO switch in 5V Side when use together with HC-SR04 Module.

3.2 HCSR04 library function description

Long timing()

Function name: timing
Parameters: None
Return Value: the time of ultrasonic from the transmitter to the receiver

float CalcDistance(long microsec,int metric)

Function name：CalcDistance
- microsec: the time of ultrasonic from the transmitter to the receiver
- metric: Set the unit of the return value (the value of 1 for cm, and the value of 0 for in)

Return Value: the measured distance

3.3 Add the HC-SR04 Library

Step1:Download the Demo Code of HCSR04 Ultrasonic from address http://www.elecfreaks.com/store/download/product/Sensor/HC-SR04/HCSR04Ultrasonic_demo.zip and then unpack it to get the file of HCSR04 Ultrasonic.

Step2: Add the file of HCSR04 Ultrasonic in the file of Arduino-1.0.X / libraries.

Step3:If you can see the Example of HCSR04 Ultrasonic in Arduino IDE, the adding of HC-SR04 library has been successful.

3.4 Test the Module with the Examples of Library File

1. Open Arduino IDE 1.0.X, and choose the corresponding board and serial port.
2. Click file/ examples/ HCSR04Ultrasonic until the code pop up.
3. Compiling sketch until Done uploading appears, which represents the uploading has been successful.
4. Open serial monitor and set the corresponding BaudRate.
5. If you see similar information in serial monitor as below, you succeeded.

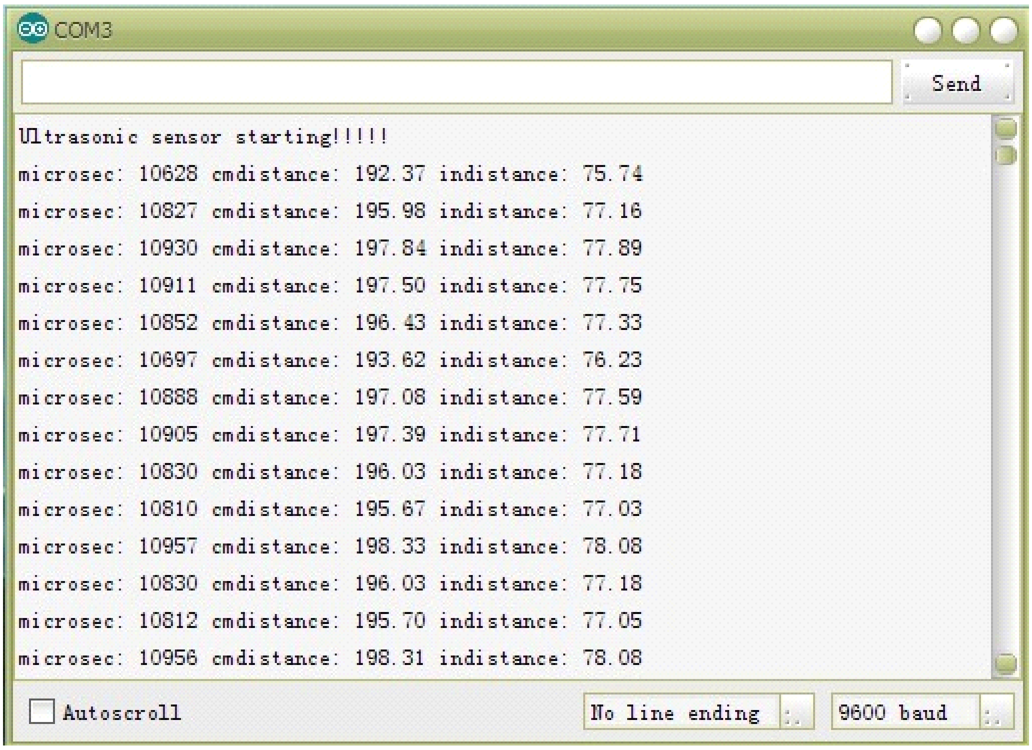


Chart 3、HC-SR04 testing results

LAMPIRAN 7

LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator

1 Features

- 3.3-V, 5-V, 12-V, and Adjustable Output Versions
- Adjustable Version Output Voltage Range: 1.2-V to 37-V ± 4% Maximum Over Line and Load Conditions
- Available in TO-220 and TO-263 Packages
- 3-A Output Load Current
- Input Voltage Range Up to 40 V
- Requires Only 4 External Components
- Excellent Line and Load Regulation Specifications
- 150-kHz Fixed-Frequency Internal Oscillator
- TTL Shutdown Capability
- Low Power Standby Mode, I_Q, Typically 80 μA
- High Efficiency
- Uses Readily Available Standard Inductors
- Thermal Shutdown and Current-Limit Protection
- Create a Custom Design Using the LM2596 with the WEBENCH Power Designer

2 Applications

- Simple High-Efficiency Step-Down (Buck) Regulator
- On-Card Switching Regulators
- Positive to Negative Converter

3 Description

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed-frequency oscillator.

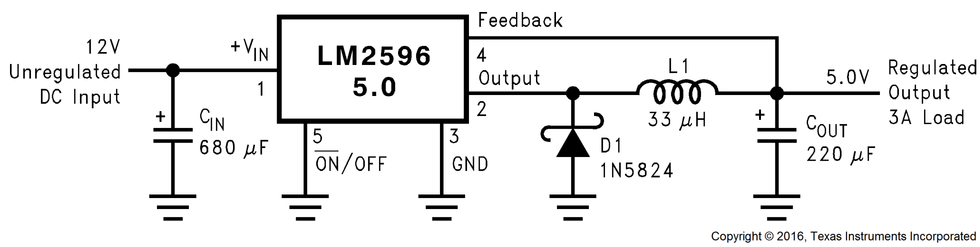
The LM2596 series operates at a switching frequency of 150 kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators. Available in a standard 7-pin TO-220 package with several different lead bend options, and a 7-pin TO-263 surface mount package.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM2596	TO-220 (7)	14.986 mm × 10.16 mm
	TO-263 (7)	10.10 mm × 8.89 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Typical Application



(Fixed Output Voltage Versions)

Table of Contents

4 Revision History

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

Changes from Revision C (April 2013) to Revision D — **Page**

- Added *ESD Ratings* table, *Feature Description* section, *Device Functional Modes*, *Application and Implementation* section, *Power Supply Recommendations* section, *Layout* section, *Device and Documentation Support* section, and *Mechanical, Packaging, and Orderable Information* section. 1
- Removed all references to design software *Switchers Made Simple* 1

Changes from Revision B (April 2013) to Revision C — **Page**

- Changed layout of National Semiconductor Data Sheet to TI format 10

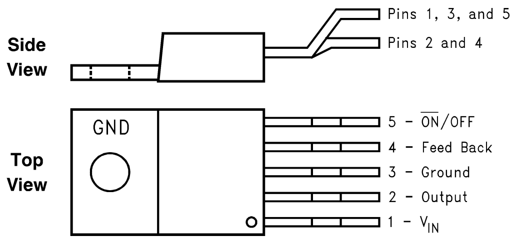
5 Description (continued)

A standard series of inductors are available from several different manufacturers optimized for use with the LM2596 series. This feature greatly simplifies the design of switch-mode power supplies.

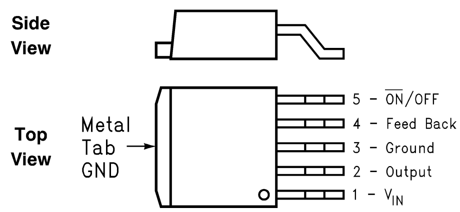
Other features include a ±4% tolerance on output voltage under specified input voltage and output load conditions, and ±15% on the oscillator frequency. External shutdown is included, featuring typically 80 μA standby current. Self-protection features include a two stage frequency reducing current limit for the output switch and an overtemperature shutdown for complete protection under fault conditions.

6 Pin Configuration and Functions

NDH Package
7-Pin TO-220
Top View



KTT Package
7-Pin TO-263
Top View



Pin Functions

PIN		I/O	DESCRIPTION
NO.	NAME		
1	V_{IN}	I	This is the positive input supply for the IC switching regulator. A suitable input bypass capacitor must be present at this pin to minimize voltage transients and to supply the switching currents required by the regulator.
2	Output	O	Internal switch. The voltage at this pin switches between approximately ($+V_{IN} - V_{SAT}$) and approximately −0.5 V, with a duty cycle of V_{OUT} / V_{IN}. To minimize coupling to sensitive circuitry, the PCB copper area connected to this pin must be kept to a minimum.
3	Ground	—	Circuit ground.
4	Feedback	I	Senses the regulated output voltage to complete the feedback loop.
5	$\overline{\text{ON}}$/OFF	I	Allows the switching regulator circuit to be shut down using logic signals thus dropping the total input supply current to approximately 80 µA. Pulling this pin below a threshold voltage of approximately 1.3 V turns the regulator on, and pulling this pin above 1.3 V (up to a maximum of 25 V) shuts the regulator down. If this shutdown feature is not required, the $\overline{\text{ON}}$/OFF pin can be wired to the ground pin or it can be left open. In either case, the regulator will be in the ON condition.

7 Specifications

7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)(2)

			MIN	MAX	UNIT
Maximum supply voltage (V_{IN})				45	V
$\overline{SD}$/SS pin input voltage(3)				6	V
Delay pin voltage(3)				1.5	V
Flag pin voltage			–0.3	45	V
Feedback pin voltage			–0.3	25	V
Output voltage to ground, steady-state				–1	V
Power dissipation			Internally limited		
Lead temperature	KTW package	Vapor phase (60 s)		215	°C
		Infrared (10 s)		245	
	NDZ package, soldering (10 s)			260	
Maximum junction temperature				150	°C
Storage temperature, T_{stg}			–65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
(3) Voltage internally clamped. If clamp voltage is exceeded, limit current to a maximum of 1 mA.

7.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001(1)	±2000	V

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

7.3 Operating Conditions

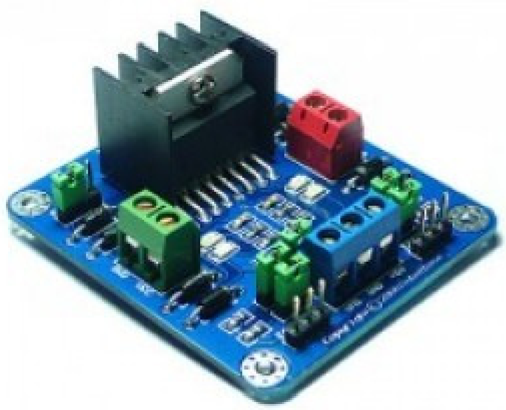
	MIN	MAX	UNIT
Supply voltage	4.5	40	V
Temperature	–40	125	°C

7.4 Thermal Information

THERMAL METRIC(1)			LM2596		UNIT
			KTW (TO-263)	NDZ (TO-220)	
			7 PINS	7 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance(2)(3)	See(4)	—	50	°C/W
		See(5)	50	—	
		See(6)	30	—	
		See(7)	20	—	
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance		2	2	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.
(2) The package thermal impedance is calculated in accordance to JESD 51-7.
(3) Thermal Resistances were simulated on a 4-layer, JEDEC board.
(4) Junction to ambient thermal resistance (no external heat sink) for the package mounted TO-220 package mounted vertically, with the leads soldered to a printed circuit board with (1 oz.) copper area of approximately 1 in^2.
(5) Junction to ambient thermal resistance with the TO-263 package tab soldered to a single sided printed circuit board with 0.5 in^2 of 1-oz copper area.
(6) Junction to ambient thermal resistance with the TO-263 package tab soldered to a single sided printed circuit board with 2.5 in^2 of 1-oz copper area.
(7) Junction to ambient thermal resistance with the TO-263 package tab soldered to a double sided printed circuit board with 3 in^2 of 1-oz copper area on the LM2596S side of the board, and approximately 16 in^2 of copper on the other side of the PCB.

LAMPIRAN 8

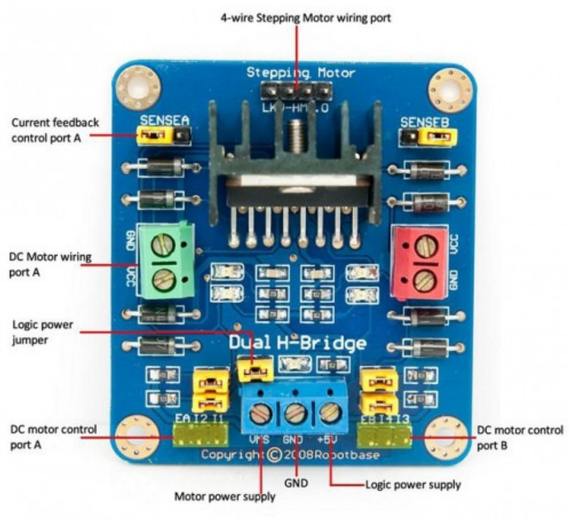


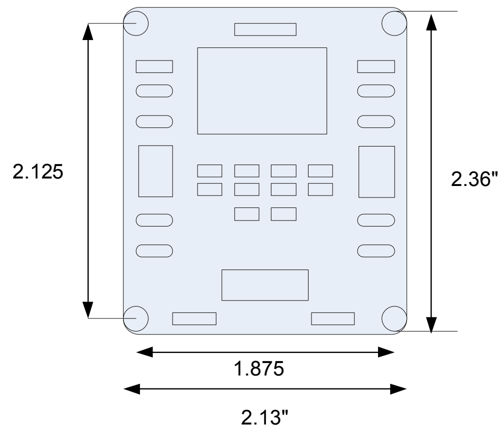
L298 Dual H-Bridge Motor Driver

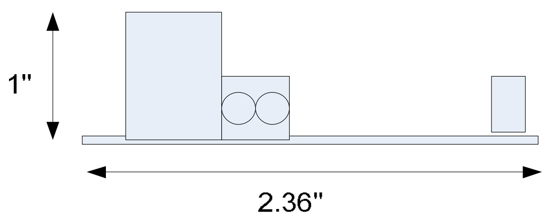
Technical Data Sheet

L298 Dual H-Bridge Motor Driver

Technical data sheet







<u>Motor Driver Features:</u>
Compact and Light Weight
High Capacity Heat Sink
Motor Direction LEDs (indicates direction of motor)
Current Feedback for both Ports
Four Pull Up Resistor Switch
Four Standard Mounting Holes

<u>Motor Driver Specifications:</u>

Driver Voltage:	5VDC-46VDC
Driver Peak Current:	2A
Logic Voltage:	5VDC
Logic Power Output:	5VDC at 1A max
Logic Current:	0-36 mA
Logic Levels:	Low: -.3V to 1.5V ; High: 2.3V to VCC
Interface:	TTL
Interface connectors:	Terminal blocks, .100 headers
Dimensions:	2.36” x 2.13” x 1” (not including connector)
Weight:	48 grams
Mounting:	4 corner holes, 0.125” diameter

Performance Specifications:

Max Drive Power: 25W
Commutation Frequency: 40KHz
Driver Peak Current: 2A
Working Temperature: -25 ℃ to +130 ℃

Description: The L298 Dual H-Bridge Motor Driver is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors.

The Dual H-bridge motordriver has the ability to drive two motors at the same time. Both DC motor inputs have three pins. DC motor input A has pins EA, I2, and I1, while DC motor input B has pins EB, I4, and I3. I1, I2, I3, and I4 are used to control the direction the motor will spin, while EA and EB connect to the PWM port of the control board in order to control the speed of the rotating motors.

EA/EB	I1/I3	I2/I4	Motor A/B direction
0	0	1	Clockwise rotation
0	1	0	Counterclockwise rotation

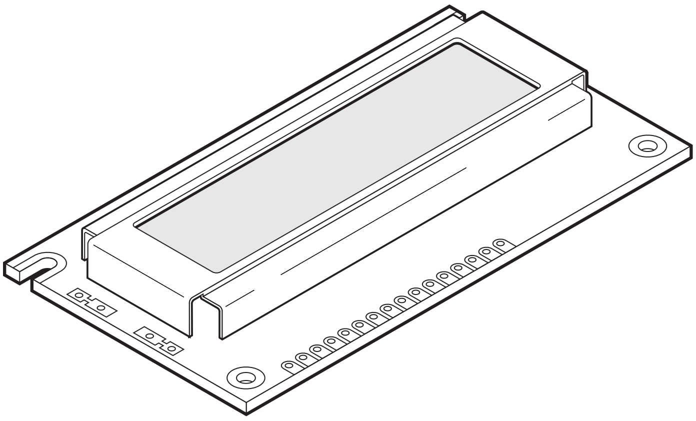
Refer to documents:
- Manufacturer Data Sheet
- QuickStart Guide

LAMPIRAN 9

Alphanumeric LCD Display (16 x 2)

Order Code

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



Contents

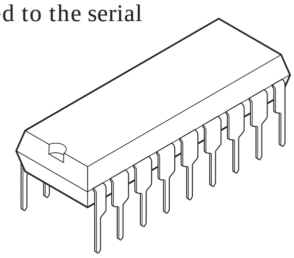
1 x 16x2 Alphanumeric Display
1 x data booklet

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

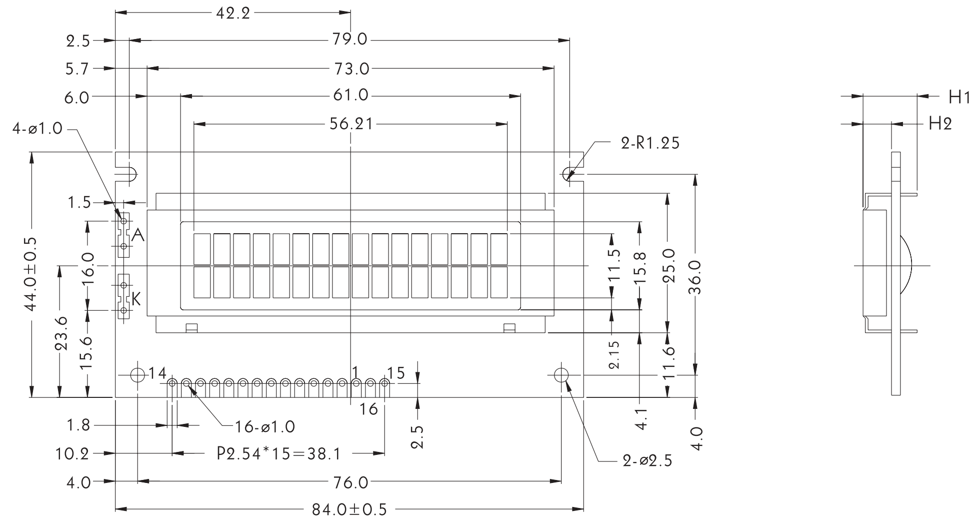
Available as an optional extra is the Serial LCD Firmware, which allows serial control of the display. This option provides much easier connection and use of the LCD module. The firmware enables microcontrollers (and microcontroller based systems such as the PICAXE) to visually output user instructions 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.

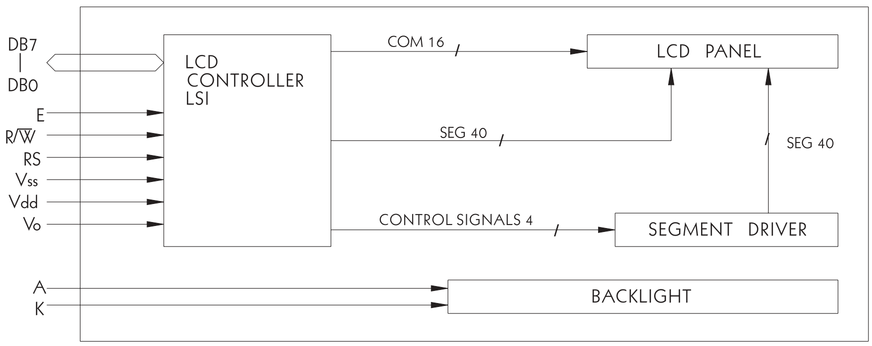


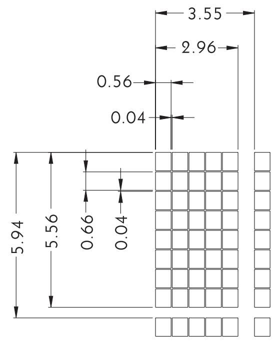
An example PICAXE instruction to print the text 'Hello' using the **serout** command is as follows:

```
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/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/W,E DB0～DB7
Supply current	Idd	Vdd = 5V	2	—	—	mA	Vdd
LCD operating voltage	VLCD	Vdd—V0	3.0	—	11.0	V	V0

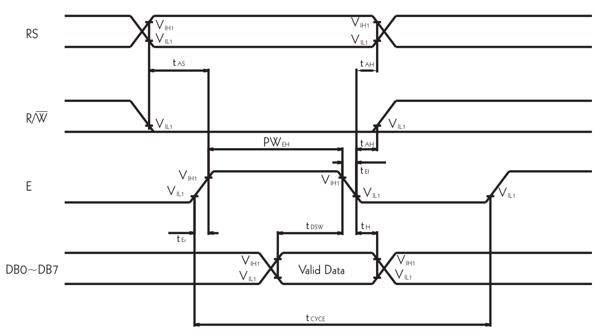
Timing Characteristics

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/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

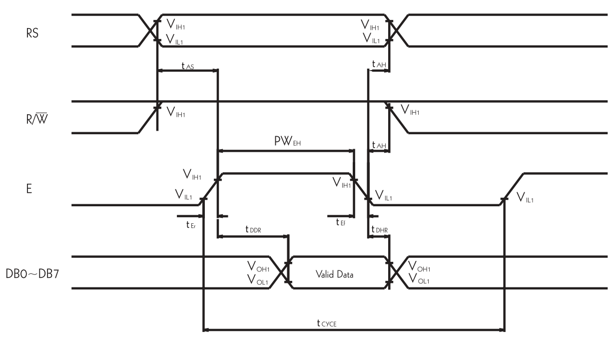
Timing Chart

◆ FIG.1 WRITE OPERATION



(Write Data from MPU to MODULE)

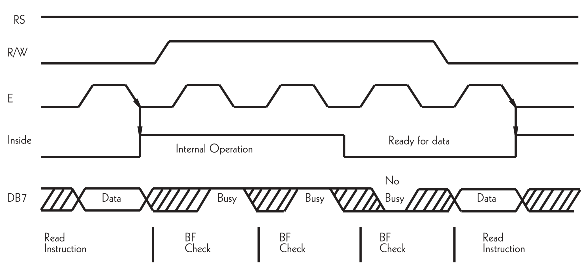
◆ FIG.2 READ OPERATION

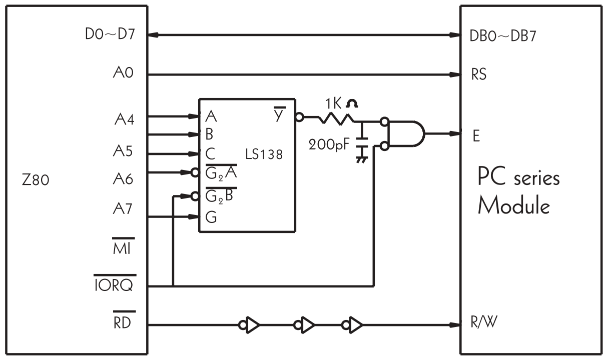


(Read Data from MODULE to MPU)

Interface with MPU

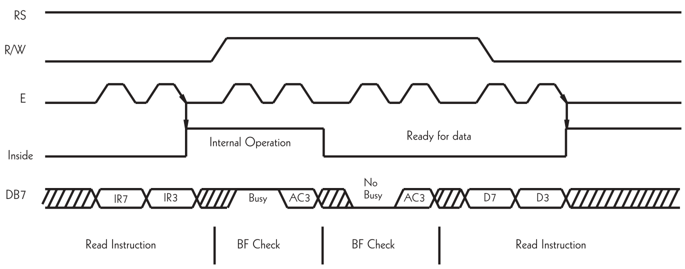
◆ Example of Interface with 8-bit MPU (Z80)



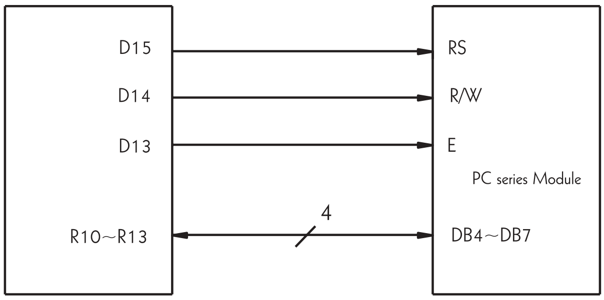


◆ Example of interface with 4-bit MPU

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.



Note:IR7,IR3:7th bit,3rd bit of instruction
AC3:3th bit of Address Counter



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

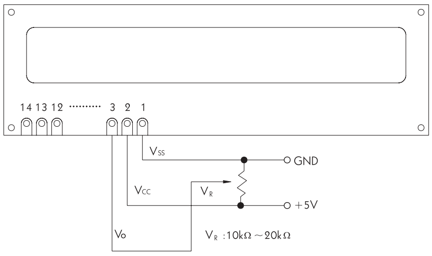
Data transfer are made through 4 bus lines from DB4 to DB7. (while the rest of 4 bus lines fromDB0 to DB3 are not used.) Data transfer with MPU are completed when 4-bit data are transfered in twice.
(first upper 4-bit data. then lower 4-bit data.)

◆ If interface data is 8-bit long

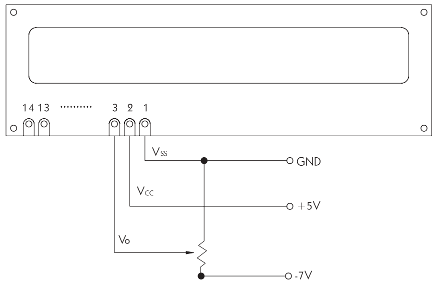
Data transfer are made through all of 8 bus lines from DB0 to DB7.

Example of Power Supply

◆ Normal Temperature Type

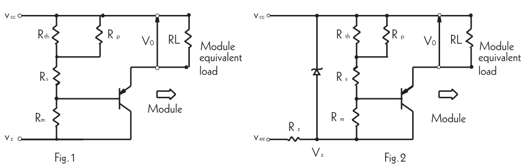


◆ Extended Temperature Type



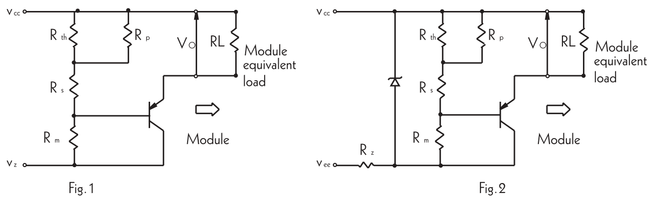
◆ Examples of Temperature Compensation Circuits for Extended Temp Type. (Only for reference)

(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**

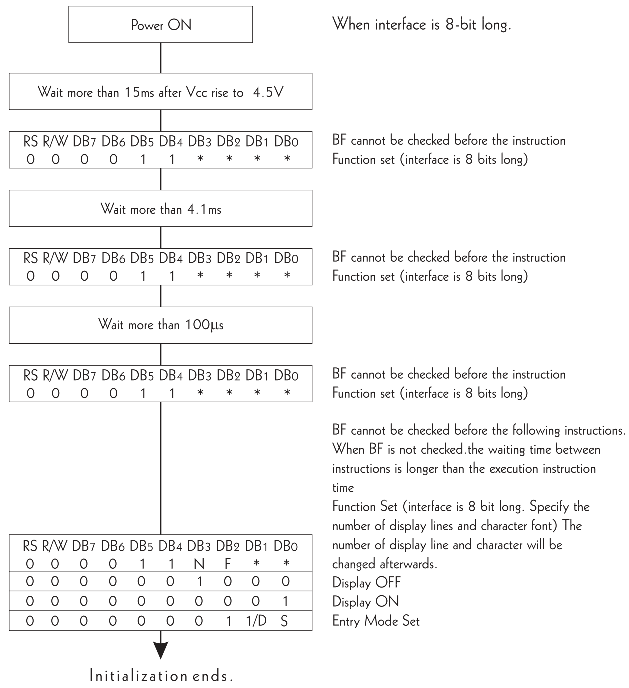
HD44780 automatically initializes (resets) when power is supplied (builtin internal reset circuit). The following instructions are executed in initialization. The busy flag (BF) is kept in busy state until initialization ends. (BF=1) The busy state is 10 ms after Vdd reachs to 4.5V.

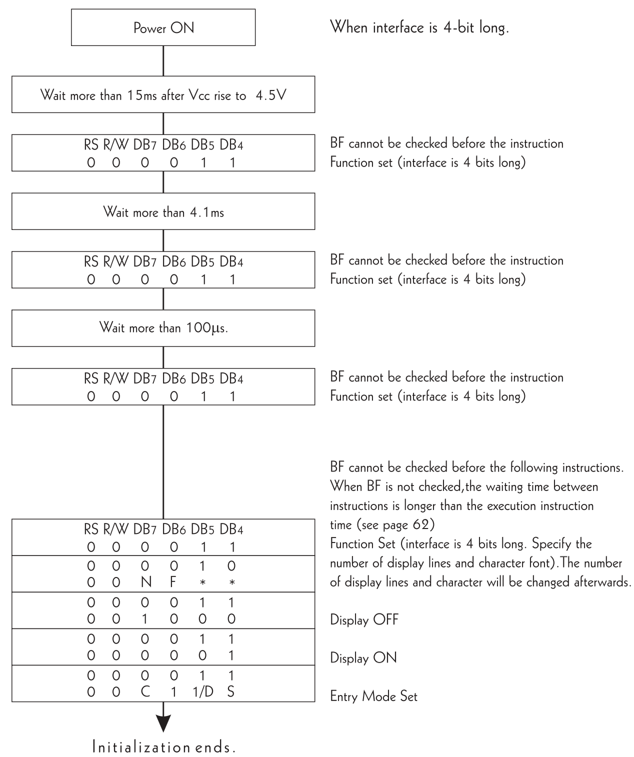
(1) Display clear
(2) Function set
DL= 1:8 bit long interface data
DL= 0:4 bit F= 0:5 x 7dots character font
N= 1:2 lines
N= 0:1 line
(3) Display ON / OFF control
D= 0:Display OFF C= 0:Cursor OFF
B= 0:Blink OFF
(4) Entry mode set
1 / D= 1:+1(increment) S= 0:No shift

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.

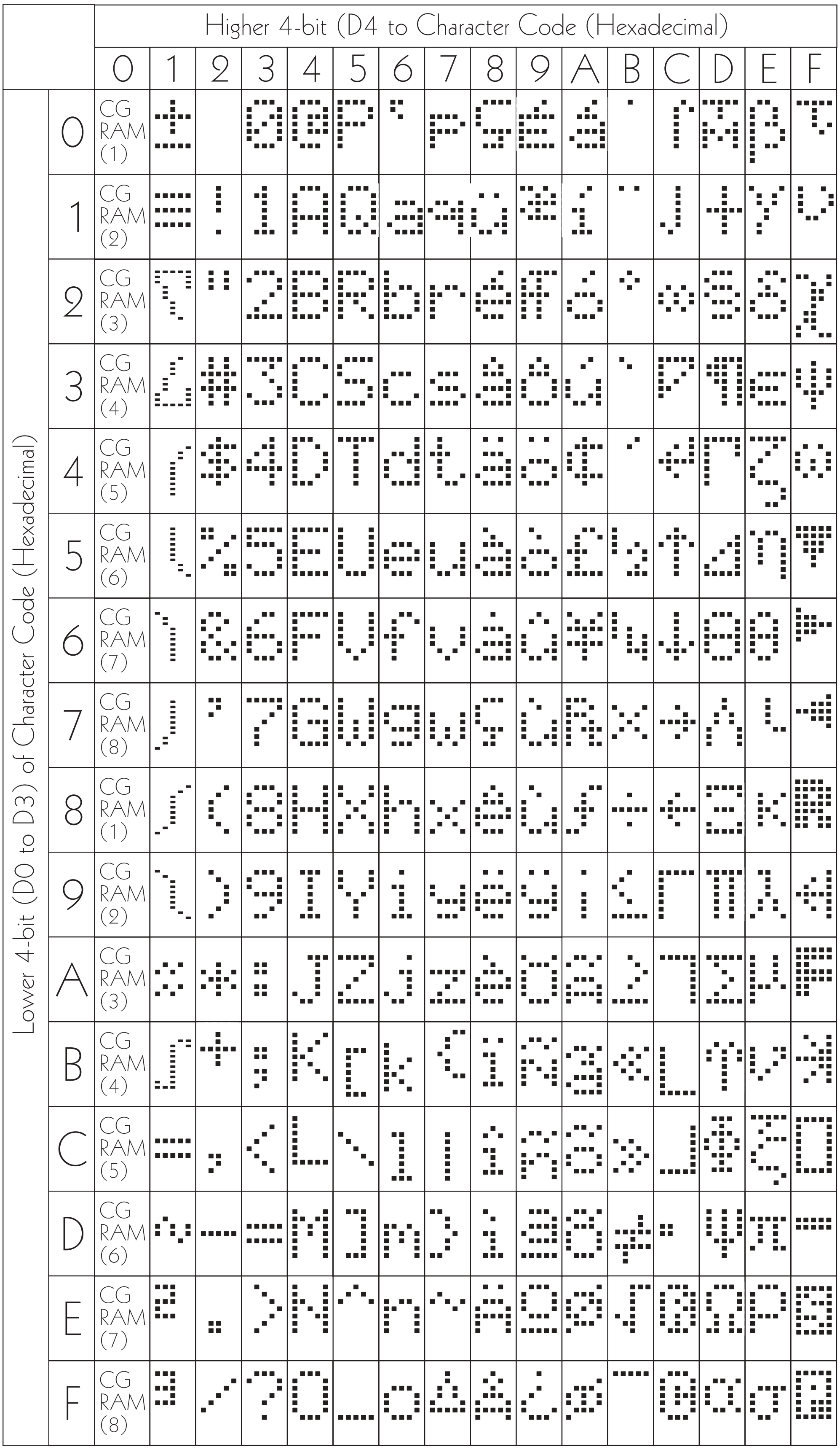
◆ **Initialization along with instruction**

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.

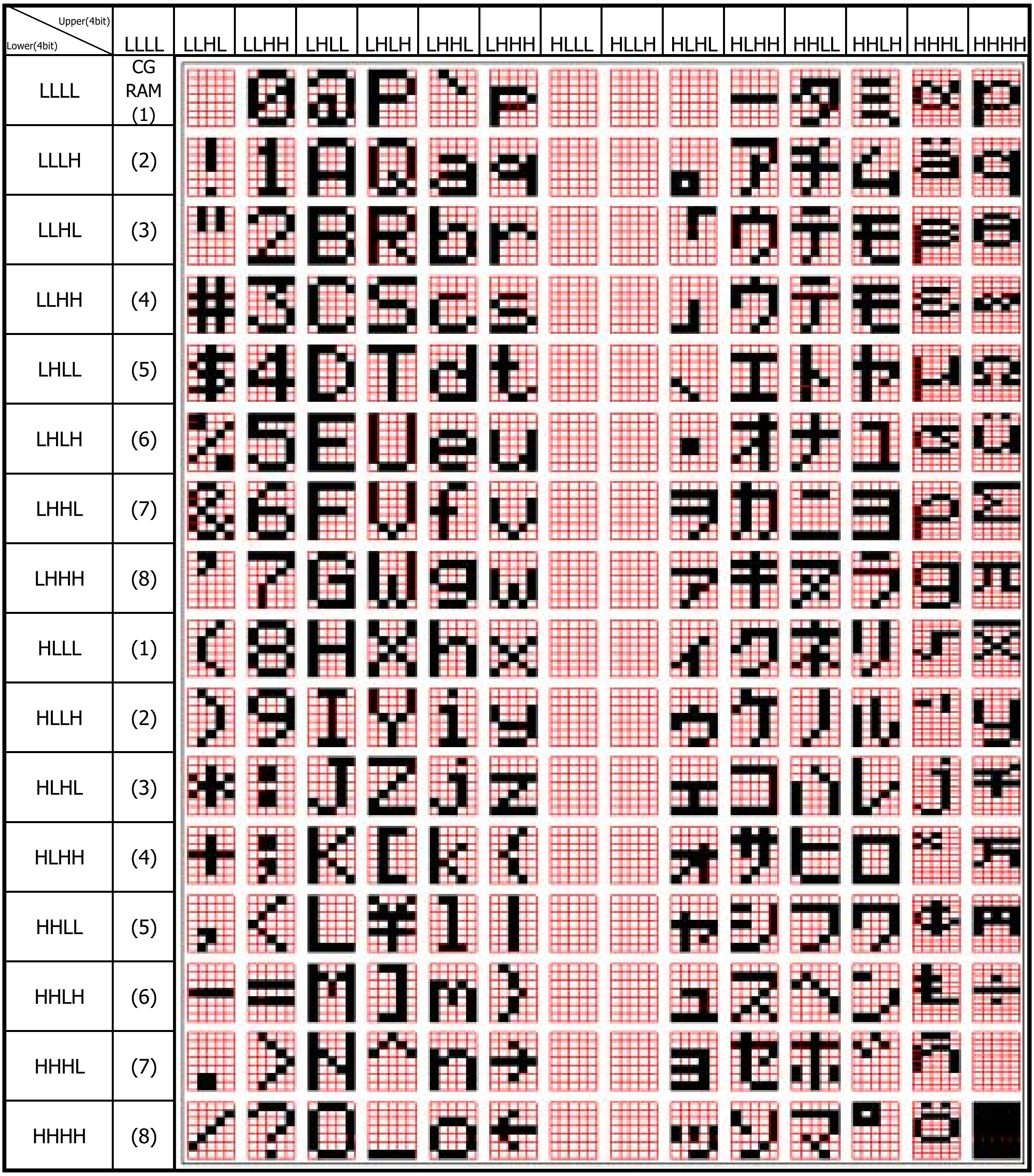




Standard Character Pattern (Powertip Module)

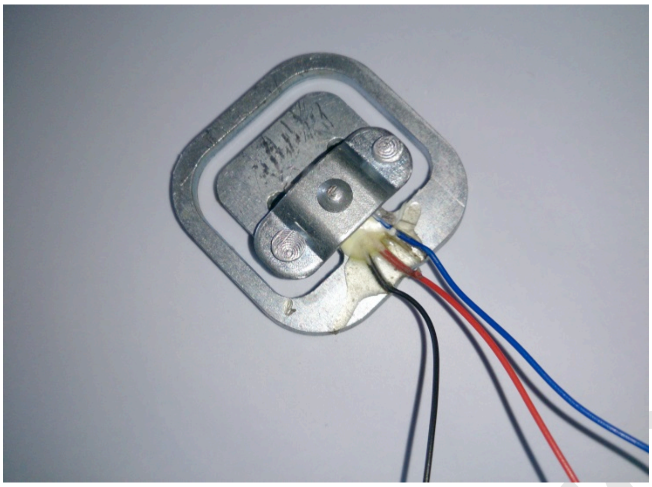


Standard Character Pattern (Elec & Eltek Module)



LAMPIRAN 10

Load Cell 50KG



This weight sensor (load cell half bridge 50KG) is suitable for electronic balance and other high accuracy electronic weighing devices.

When measuring, the correct force is applied to the outer side of the strain E-shaped beam portion of the sensor and the outside edges to form a shear force in the opposite direction.

Tips:

1-The sensor use the following three methods:

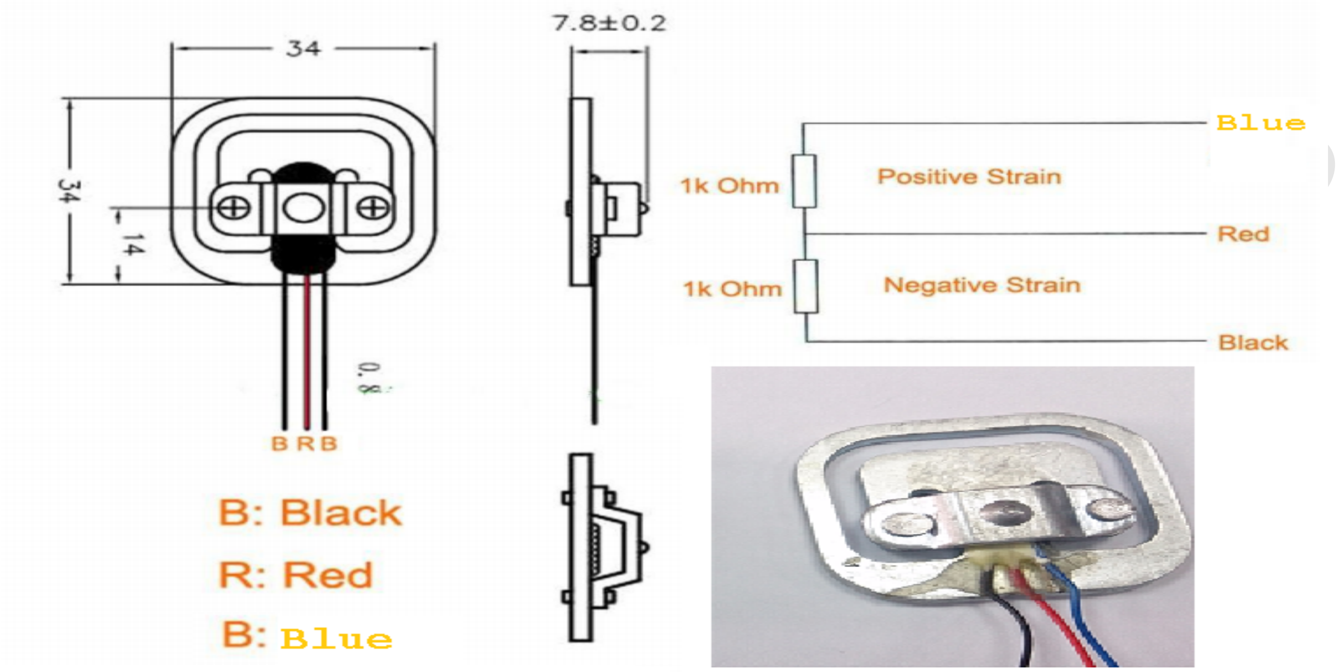
Using a sensor with external resistors to make full bridge measurement range 50kg.

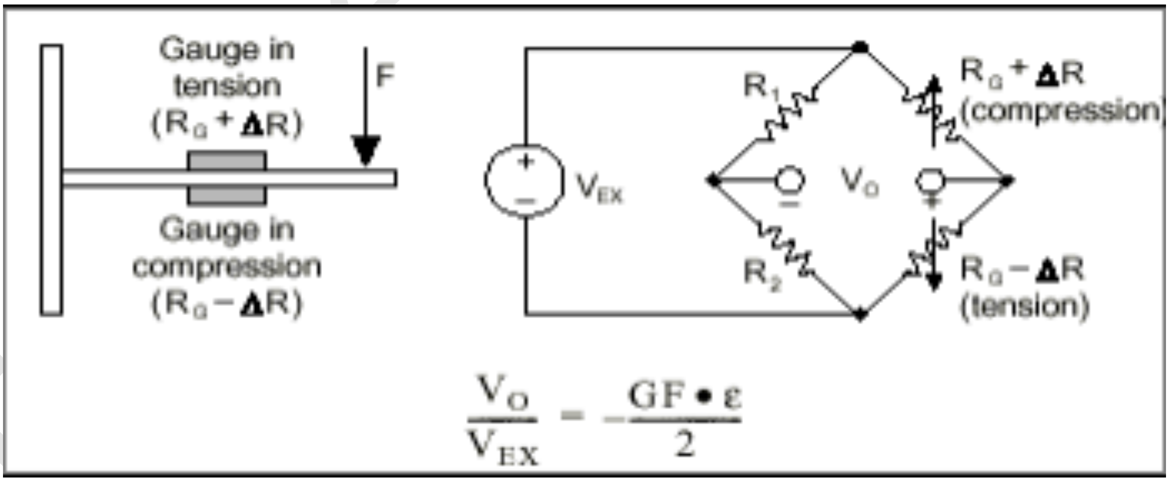
The use of only two full-bridge sensors measuring range is the range of the two sensors and: 50kgx2 = 100kg.

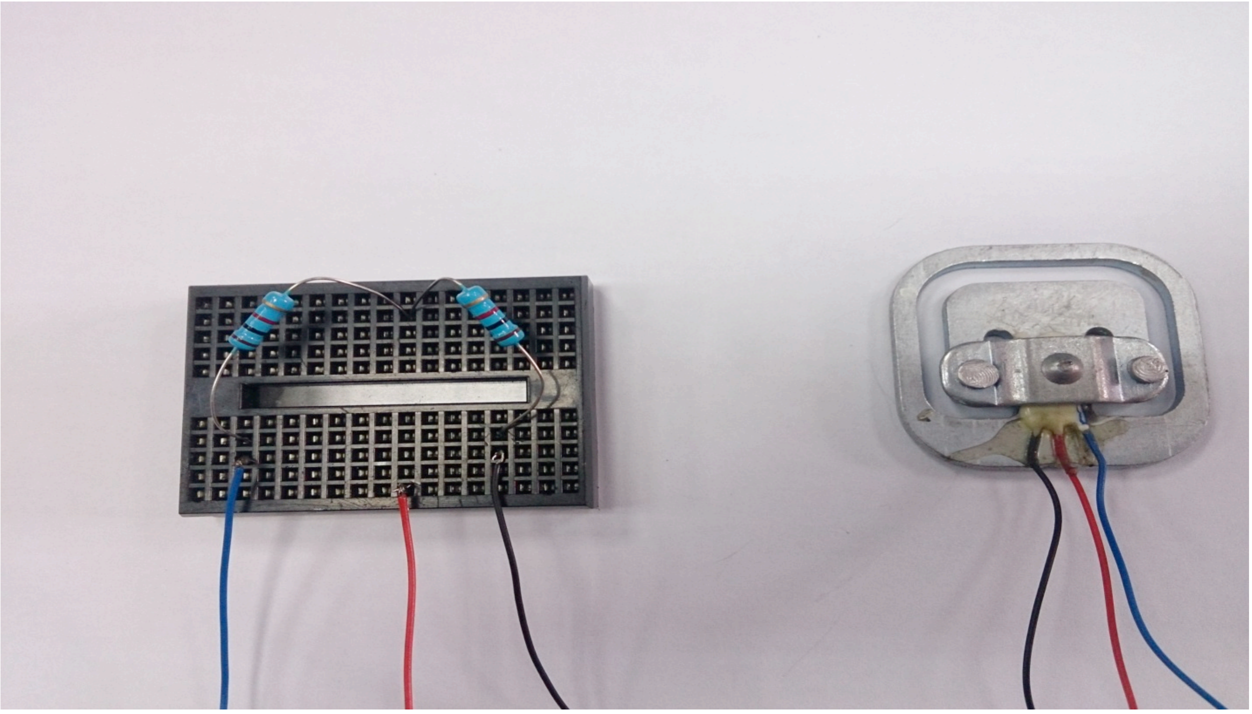
The use of four full-bridge sensors measuring range is the range of four sensors and: 50kgx4 = 200kg

2-When measuring, the correct force is applied to the outer side of the strain E-shaped beam portion of the sensor and the outside edges to form a shear force in the opposite direction.

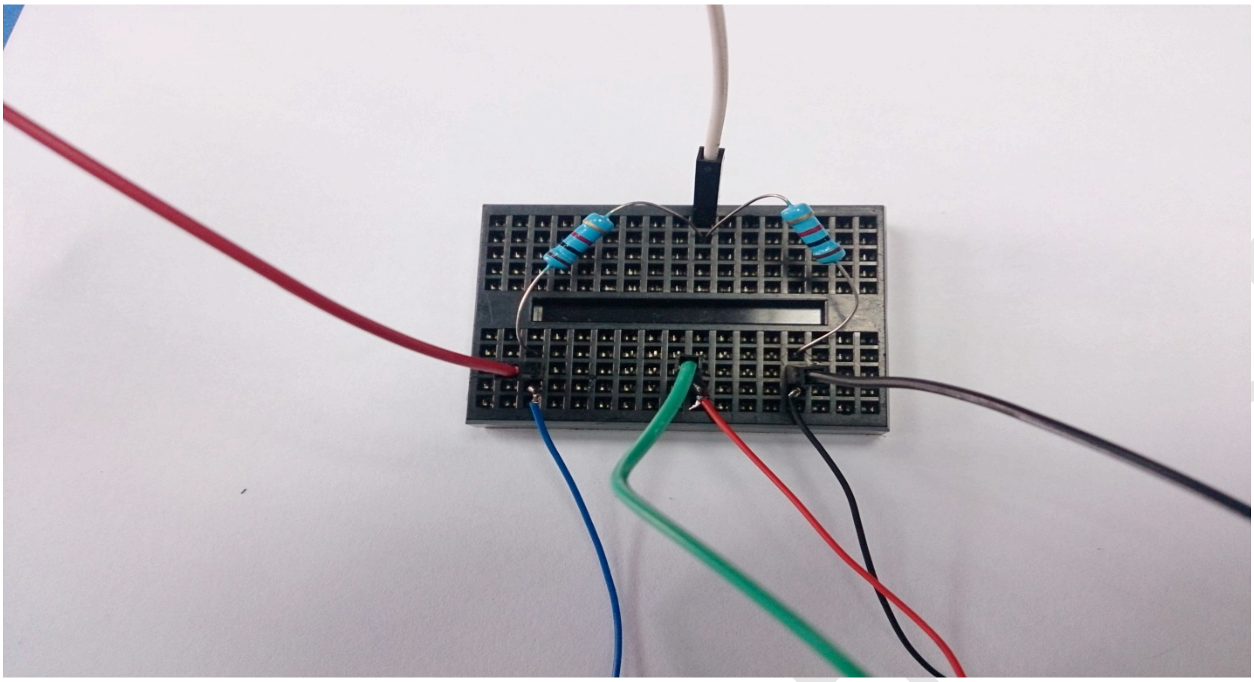
Pin Connection







VCC:	**Red wire**
GND:	**Black wire**
Signal + :	**Green wire**
Signal - :	**White wire**



Connect Load Cell Half bridge 50KG with Weight Scales Analog-to-Digital Converter (ADC) 24-bit and arduino

Load Cell to Weight Scales

Out +: **sensor power (connect to red wire of weight sensor**
A- **:** **Signal -ve (connect to white wire of first weight sensor)**
A+ **:** **Signal +ve (connect to green wire of first weight sensor)**
GND **:** **Weight sensor ground (connect to black wire of weight sensor)**

Weight Scales to micro controller (arduino) side:

VCC: 5 VDC
DO : Digital Output
CK : Clock
GND: Ground

LAMPIRAN 11

HX711

24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of ±20mV or ±40mV respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

FEATURES

- **Two selectable differential input channels**
- **On-chip active low noise PGA with selectable gain of 32, 64 and 128**
- **On-chip power supply regulator for load-cell and ADC analog power supply**
- **On-chip oscillator requiring no external component with optional external crystal**
- **On-chip power-on-reset**
- **Simple digital control and serial interface: pin-driven controls, no programming needed**
- **Selectable 10SPS or 80SPS output data rate**
- **Simultaneous 50 and 60Hz supply rejection**
- **Current consumption including on-chip analog power supply regulator:**
  normal operation < 1.5mA, power down < 1uA
- **Operation supply voltage range: 2.6 ~ 5.5V**
- **Operation temperature range: -40 ~ +85℃**
- **16 pin SOP-16 package**

APPLICATIONS

- **Weigh Scales**
- **Industrial Process Control**

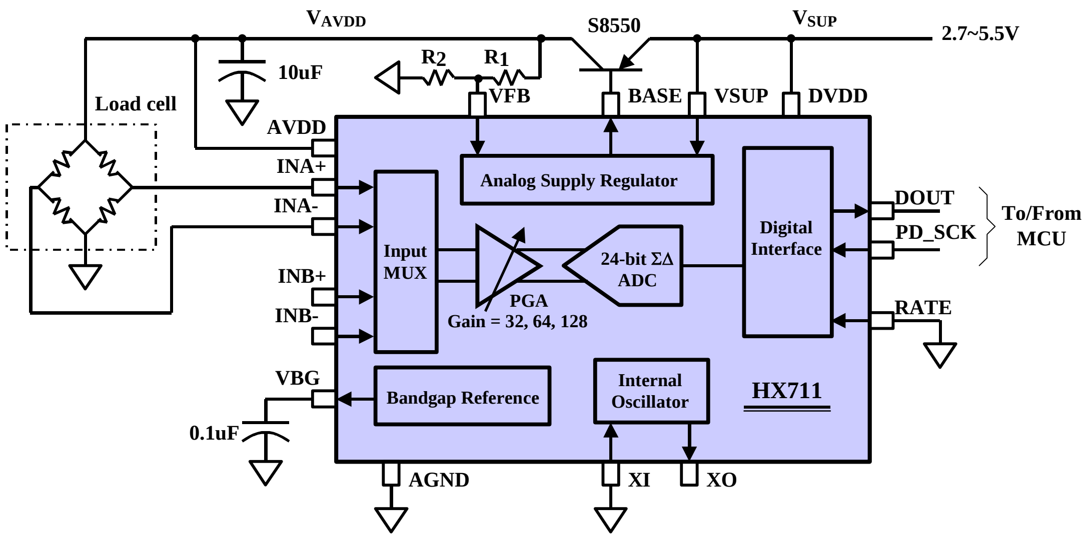
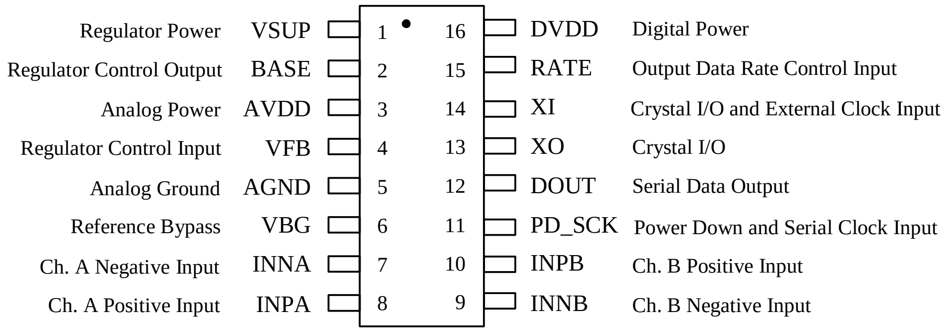


Fig. 1 Typical weigh scale application block diagram

Pin Description



SOP-16L Package

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output（NC when not used）
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input（connect to AGND when not used）
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used）
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

Table 1 Pin Description

KEY ELECTRICAL CHARACTERISTICS

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	V(inp)-V(inn)		±0.5(AVDD/GAIN)		V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0		10		Hz
	Internal Oscillator, RATE = DVDD		80		
	Crystal or external clock, RATE = 0		f_{clk}/1,105,920		
	Crystal or external clock, RATE = DVDD		f_{clk}/138,240		
Output data coding	2's complement	800000		7FFFFF	HEX
Output settling time (1)	RATE = 0		400		ms
	RATE = DVDD		50		
Input offset drift	Gain = 128		0.2		mV
	Gain = 64		0.4		
Input noise	Gain = 128， RATE = 0		50		nV(rms)
	Gain = 128， RATE = DVDD		90		
Temperature drift	Input offset（Gain = 128）		±6		nV/℃
	Gain（Gain = 128）		±5		ppm/℃
Input common mode rejection	Gain = 128， RATE = 0		100		dB
Power supply rejection	Gain = 128， RATE = 0		100		dB
Reference bypass（V_{BG}）			1.25		V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6		5.5	V
	AVDD，VSUP	2.6		5.5	
Analog supply current (including regulator)	Normal		1400		µA
	Power down		0.3		
Digital supply current	Normal		100		µA
	Power down		0.2		

（1）Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

Table 2 Key Electrical Characteristics

Analog Inputs

Channel A differential input is designed to interface directly with a bridge sensor's differential output. It can be programmed with a gain of 128 or 64. The large gains are needed to accommodate the small output signal from the sensor. When 5V supply is used at the AVDD pin, these gains correspond to a full-scale differential input voltage of ±20mV or ±40mV respectively.

Channel B differential input has a fixed gain of 32. The full-scale input voltage range is ±80mV, when 5V supply is used at the AVDD pin.

Power Supply Options

Digital power supply (DVDD) should be the same power supply as the MCU power supply.

When using internal analog supply regulator, the dropout voltage of the regulator depends on the external transistor used. The output voltage is equal to $V_{AVDD}=V_{BG}*(R1+R2)/R1$ (Fig. 1). This voltage should be designed with a minimum of 100mV below VSUP voltage.

If the on-chip analog supply regulator is not used, the VSUP pin should be connected to either AVDD or DVDD, depending on which voltage is higher. Pin VFB should be connected to Ground and pin BASE becomes NC. The external 0.1uF bypass capacitor shown on Fig. 1 at the VBG output pin is then not needed.

Clock Source Options

By connecting pin XI to Ground, the on-chip oscillator is activated. The nominal output data rate when using the internal oscillator is 10 (RATE=0) or 80SPS (RATE=1).

If accurate output data rate is needed, crystal or external reference clock can be used. A crystal can be directly connected across XI and XO pins. An external clock can be connected to XI pin, through a 20pF ac coupled capacitor. This external clock is not required to be a square wave. It can come directly from the crystal output pin of the MCU chip, with amplitude as low as 150 mV.

When using a crystal or an external clock, the internal oscillator is automatically powered down.

Output Data Rate and Format

When using the on-chip oscillator, output data rate is typically 10 (RATE=0) or 80SPS (RATE=1).

When using external clock or crystal, output data rate is directly proportional to the clock or crystal frequency. Using 11.0592MHz clock or crystal results in an accurate 10 (RTE=0) or 80SPS (RATE=1) output data rate.

The output 24 bits of data is in 2's complement format. When input differential signal goes out of the 24 bit range, the output data will be saturated at 800000h (MIN) or 7FFFFFh (MAX), until the input signal comes back to the input range.

Serial Interface

Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls.

When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted out from the DOUT output pin. Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25^{th} pulse at PD_SCK input will pull DOUT pin back to high (Fig.2).

Input and gain selection is controlled by the number of the input PD_SCK pulses (Table 3). PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Table 3 Input Channel and Gain Selection

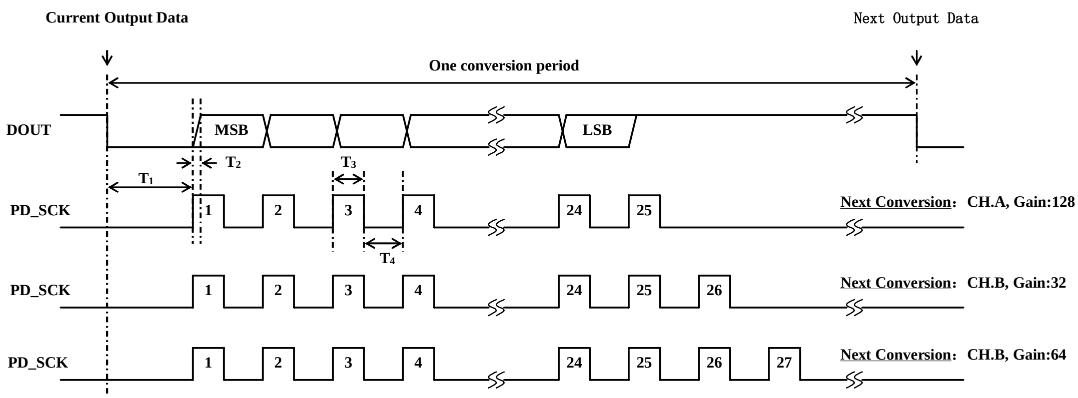


Fig.2 Data output, input and gain selection timing and control

Symbol	Note	MIN	TYP	MAX	Unit
T_1	DOUT falling edge to PD_SCK rising edge	0.1			µs
T_2	PD_SCK rising edge to DOUT data ready			0.1	µs
T_3	PD_SCK high time	0.2	1	50	µs
T_4	PD_SCK low time	0.2	1		µs

Reset and Power-Down

When chip is powered up, on-chip power on rest circuitry will reset the chip.

Pin PD_SCK input is used to power down the HX711. When PD_SCK Input is low, chip is in normal working mode.

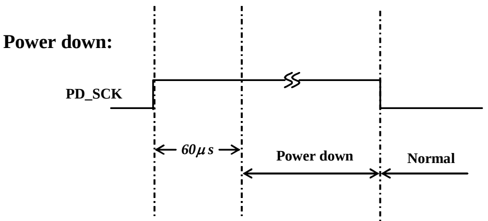


Fig.3 Power down control

When PD_SCK pin changes from low to high and stays at high for longer than 60µs, HX711 enters power down mode (Fig.3). When internal regulator is used for HX711 and the external transducer, both HX711 and the transducer will be powered down. When PD_SCK returns to low, chip will reset and enter normal operation mode.

After a reset or power-down event, input selection is default to Channel A with a gain of 128.

Application Example

Fig.1 is a typical weigh scale application using HX711. It uses on-chip oscillator (XI=0), 10Hz output data rate (RATE=0). A Single power supply （2.7～5.5V） comes directly from MCU power supply. Channel B can be used for battery level detection. The related circuitry is not shown on Fig. 1.

Reference PCB Board (Single Layer)

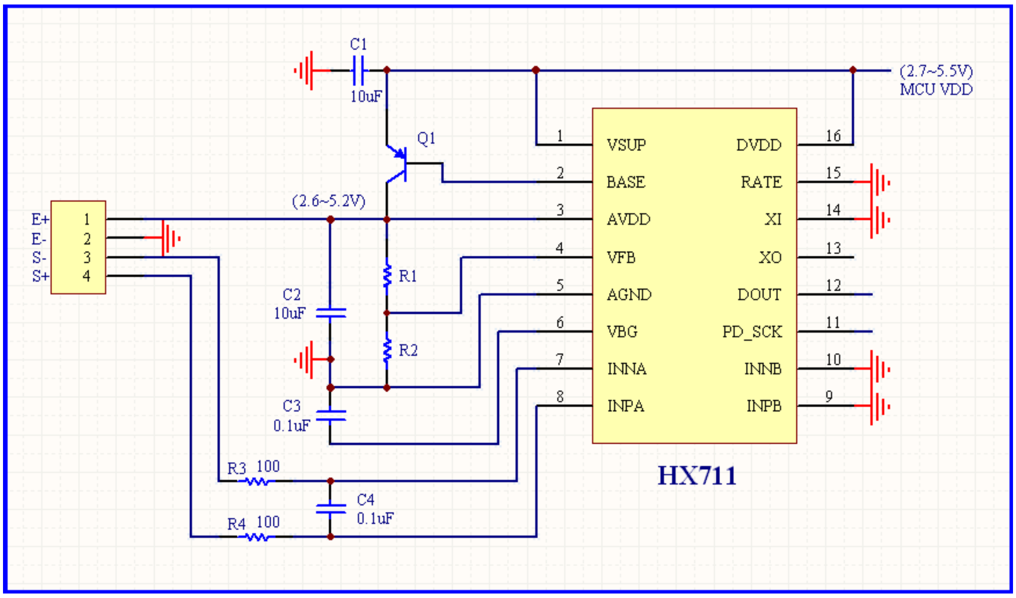


Fig.4 Reference PCB board schematic

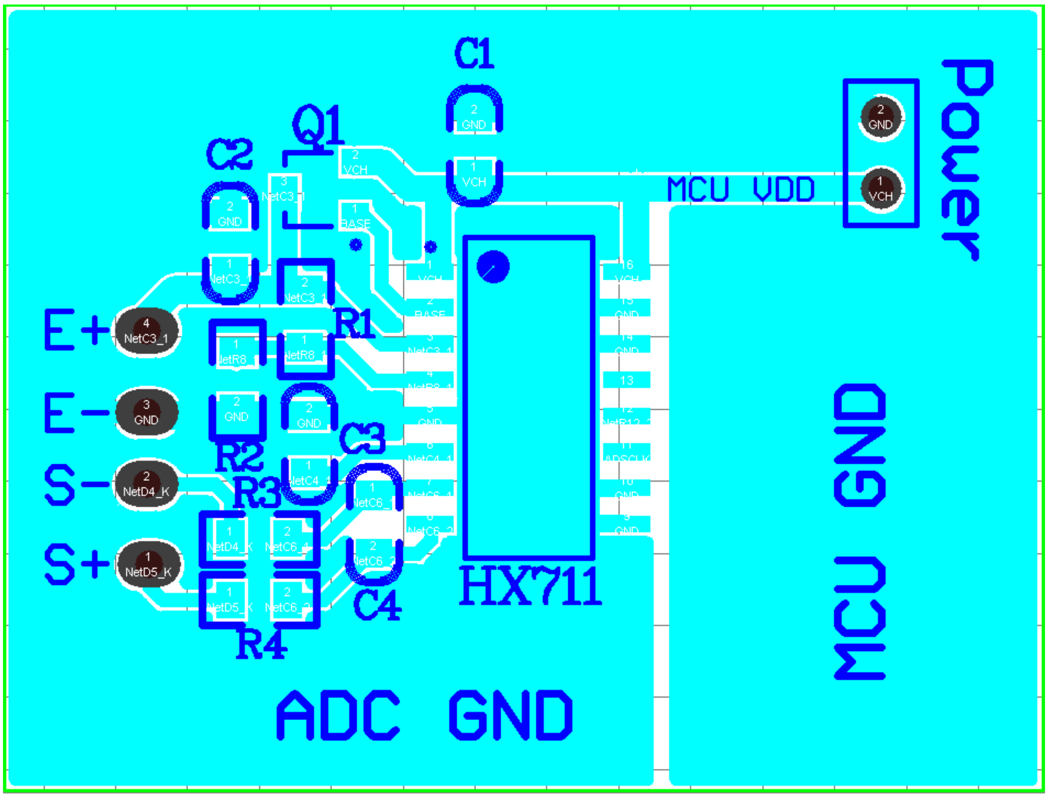


Fig.5 Reference PCB board layout

Reference Driver (Assembly)

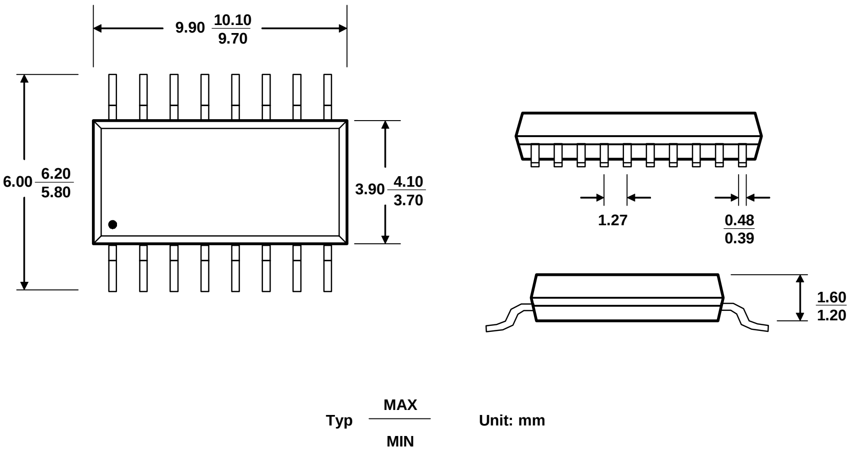
```
/*---------------------------------------------------------------------
Call from ASM:      LCALL    ReaAD
Call from C:    extern unsigned long ReadAD(void);
                   .
                   .
                  unsigned long data;
                  data=ReadAD();
                   .
                   .
----------------------------------------------------------------------*/
PUBLIC        ReadAD
HX711ROM      segment  code
rseg          HX711ROM

sbit           ADDO = P1.5;
sbit           ADSK = P0.0;
/*---------------------------------------------------
 OUT:   R4, R5, R6, R7   R7=>LSB
---------------------------------------------------*/
ReadAD:
    CLR    ADSK                //AD Enable（PD_SCK set low）
    SETB   ADDO                //Enable 51CPU I/0
    JB     ADDO,$              //AD conversion completed?
    MOV    R4,#24
ShiftOut:
    SETB   ADSK                //PD_SCK set high（positive pulse）
    NOP
    CLR    ADSK                //PD_SCK set low
    MOV    C,ADDO              //read on bit
    XCH    A,R7                //move data
    RLC    A
    XCH    A,R7
    XCH    A,R6
    RLC    A
    XCH    A,R6
    XCH    A,R5
    RLC    A
    XCH    A,R5
    DJNZ   R4,ShiftOut          //moved 24BIT?
    SETB   ADSK
    NOP
    CLR    ADSK
    RET
    END
```

Reference Driver（C）

```
//-------------------------------------------------------------------
sbit   ADDO = P1^5;
sbit   ADSK = P0^0;
unsigned long ReadCount(void){
  unsigned long Count;
  unsigned char i;
  ADDO=1;
  ADSK=0;
  Count=0;
  while(ADDO);
  for (i=0;i<24;i++){
    ADSK=1;
    Count=Count<<1;
    ADSK=0;
    if(ADDO) Count++;
  }
  ADSK=1;
  Count=Count^0x800000;
  ADSK=0;
  return(Count);
}
```

Package Dimensions



SOP-16L Package