LAMPIRAN A
FOTO SISTEM CRANE

Tampak Depan

Tampak Samping

Tampak Tengah

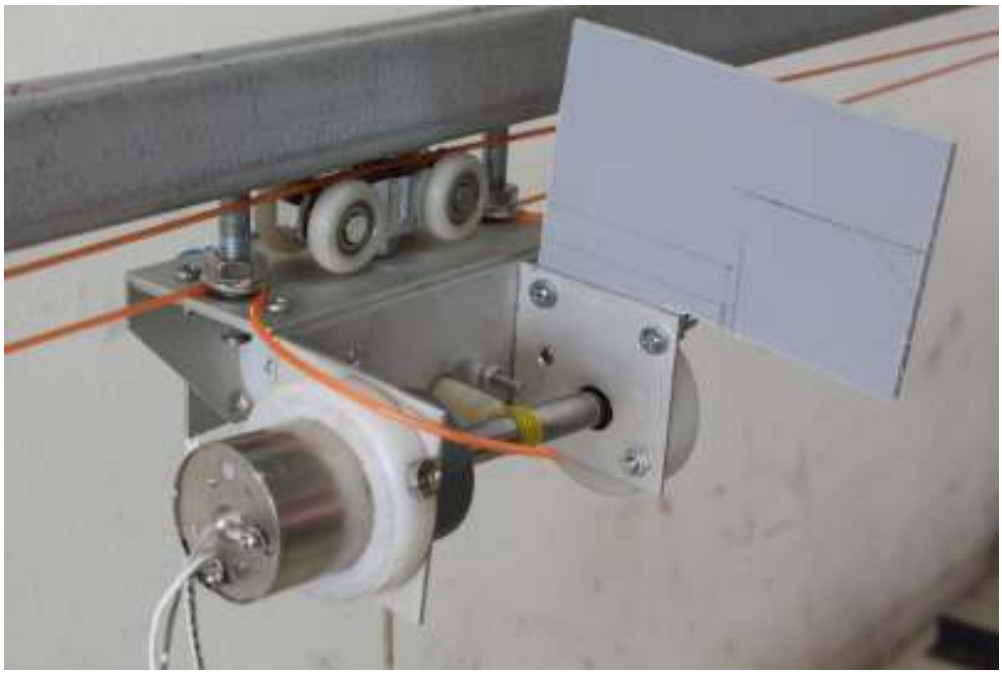
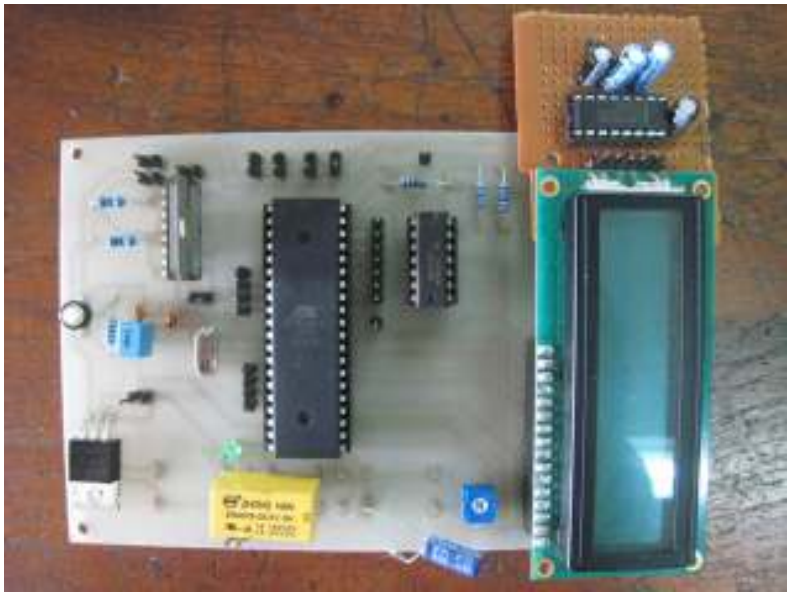


Foto *Board* Pengontrol Mikro



LAMPIRAN B
PROGRAM PADA PENGONTROL MIKRO ATMEGA16

```
/*****************************************************
This program was produced by the
CodeWizardAVR V1.25.3 Standard
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project : Final Project
Version :
Date    : 6/4/2010
Author  : Irwing Antonio
Company : Lab Robotik
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 11.059200 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

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


// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

// Standard Input/Output functions
#include <stdio.h>

#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}

// Declare your global variables here


//=====buat LCD=======
char buffer1[8];
char buffer2[8];
char buffer3[16];

//====buat posisi=====
unsigned int t,i;
```

```
float posisi,DOMposisizero,DOMclose,DOMmedium,DOMfar;

//====buat sudut======
unsigned int adc;
float sudut, DOMnegbig,DOMnegsmall,DOMsudutzero,DOMpossmall,DOMposbig;

//====buat output=====
float out,outocr,atas,bawah,imp1,imp2,imp3,imp4,imp5,imp6,imp7,imp8,imp9,imp10;
int output;




//=====================================================================================
//                  MULAI
//=====================================================================================
      void fuzzy(void)
            {



//=====================================================================================
//                PROGRAM BACA POSISI
//=====================================================================================



      //PORTD.0 digunakan sebagai signal input dan output dari dan ke Sensor Ping
     mulai:
    t=1;
    DDRD.2=1;
    PORTD.2=1;
    delay_us(5);
    PORTD.2=0;
    DDRD.2=0;
    PORTD.2=1 ;
      for(i=0;i<1050;i++)
      {
      if(PIND.2==1)
      goto coun;
      }
goto mulai;
coun:
      if(PIND.2==0)
      goto hitung;
      t=t+1 ;
      delay_us(1);
goto coun;
hitung:
      posisi=(0.0322*t)-14.8;  //KUSTOMISASI PROGRAM KONVERSI PING

goto bacasudut;
```

```
//=====================================================================================
 //              PROGRAM BACA SUDUT

//=====================================================================================
       bacasudut:

       adc=read_adc(0);
       sudut=(0.74*adc)-83.7 ;    //konversi dari nilai ADC ke sudut dgn range sudut -80 s/d 80
       goto negbig;


//=====================================================================================
 //              FUZZIFIKASI SUDUT

//=====================================================================================

      //////inisialisasi////////
      /*
      DOMnegbig=0;
      DOMnegsmall=0;
      DOMsudutzero=0;
      DOMpossmall=0;
      DOMposbig=0;
      */
 ////////////////////NEG_BIG//////////////////
 negbig:
      if (sudut<=-30)
            {
            DOMnegbig=1;
            goto negsmall;
            }

  else if (sudut>-30 && sudut<-15)
            {
            DOMnegbig=(-15-sudut)/15;
            DOMnegbig=fabs(DOMnegbig);
            goto negsmall;
            }
  else if (sudut>=-15)
            {
            DOMnegbig=0;
            goto negsmall;
            };



 ////////////////NEG_SMALL//////////////////
 negsmall:
      if (sudut>-30 && sudut<=-10)
            {
            DOMnegsmall=(sudut+30)/20;
            DOMnegsmall=fabs(DOMnegsmall);
            goto sudutzero;
            }

  else if (sudut>-10 && sudut<10)
            {
```

```
            DOMnegsmall=(10-sudut)/20;
            DOMnegsmall=fabs(DOMnegsmall);
            goto sudutzero;
            }
 else if (sudut>=10 || sudut <=-30)
            {
            DOMnegsmall=0;
            goto sudutzero;
            };


/////////////////ZERO//////////////////////
sudutzero:
     if (sudut>-15 && sudut<=0)
            {
            DOMsudutzero=(sudut+15)/15;
            DOMsudutzero=fabs(DOMsudutzero);
            goto possmall;
            }

 else if (sudut>0 && sudut<15)
            {
            DOMsudutzero=(15-sudut)/15;
            DOMsudutzero=fabs(DOMsudutzero);
            goto possmall;
            }
 else if (sudut>=15 || sudut <=-15)
            {
            DOMsudutzero=0;
            goto possmall;
            };


 ////////////////POS_SMALL//////////////////////
 possmall:
     if (sudut>-10 && sudut<=10)
            {
            DOMpossmall=(sudut+10)/20;
            DOMpossmall=fabs(DOMpossmall);
            goto posbig;
            }

 else if (sudut>10 && sudut<30)
            {
            DOMpossmall=(30-sudut)/20;
            DOMpossmall=fabs(DOMpossmall);
            goto posbig;
            }
 else if (sudut>=30 || sudut <=-10)
            {
            DOMpossmall=0;
            goto posbig;
            };


 //////////////////POS_BIG//////////////////////
 posbig:
      if (sudut>15 && sudut<=30)
            {
            DOMposbig=(sudut-15)/15;
            DOMposbig=fabs(DOMposbig);
            goto posisizero;
```

```
            }

  else  if (sudut>30)
            {
            DOMposbig=1;
            goto posisizero;
            }
   else if (sudut<=15)
            {
            DOMposbig=0;
            goto posisizero;
            };




//=================================================================================
==================================
 //             FUZZIFIKASI POSISI

//=================================================================================
==================================

     /////////inisialisasi//////////////
     /*
     DOMposisizero=0;
     DOMclose=0;
     DOMmedium=0;
     DOMfar=0;
     */
 ////////////////////ZERO//////////////////
 posisizero:
      if (posisi<1)
            {
            DOMposisizero=(1-posisi)/1;
            goto close;
            }

 else if (posisi>=1)
            {
            DOMposisizero=0;
            goto close;
            };



 ////////////////CLOSE//////////////////
 close:
      if (posisi>1 && posisi<=15)
            {
            DOMclose=(posisi-5)/10;
            goto medium;
            }

  else if (posisi>15 && posisi<30)
            {
            DOMclose=(30-sudut)/15;
            goto medium;
            }

   else if (posisi>=30 || posisi<=1)
```

```
            {
            DOMclose=0;
            goto medium;
            };

 ///////////////MEDIUM/////////////////
 medium:
     if (posisi>15 && posisi<=60)
            {
            DOMmedium=(posisi-15)/45;
            goto far;
            }

  else if (posisi>60 && posisi<120)
            {
            DOMmedium=(120-sudut)/60;
            goto far;
            }

  else if (posisi>=120 || posisi<=15)
            {
            DOMmedium=0;
            goto far;
            };


  ///////////////FAR/////////////////
  far:
     if (posisi>80 && posisi<=160)
            {
            DOMfar=(posisi-80)/80;
            goto rule;
            }

  else if (posisi>160)
            {
            DOMfar=1;
            goto rule;
            }

  else if (posisi<=80)
            {
            DOMfar=0;
            goto rule;
            };



//=================================================================================
==================================
 //                        RULE EVALUATION

//=================================================================================
==================================

    /*
      1. IF posisi = far      AND sudut = zero        THEN Tegangan = pos_medium            pos_high  =21 V
      2. IF posisi = far      AND sudut = neg_small   THEN Tegangan = pos_high              pos_medium=15 V
      3. IF posisi = far      AND sudut = neg_big     THEN Tegangan = pos_high            pos_low   =12 V
      4. IF posisi = medium   AND sudut = neg_small   THEN Tegangan = pos_medium              zero      =0 V
```

```
      5. IF posisi = medium   AND sudut = neg_big     THEN Tegangan = pos_low                neg_medium=-
10 V
      6. IF posisi = medium   AND sudut = pos_small   THEN Tegangan = pos_medium
      7. IF posisi = close    AND sudut = zero        THEN Tegangan = pos_low
      8. IF posisi = close    AND sudut = neg_small   THEN Tegangan = neg_medium
      9. IF posisi = close    AND sudut = pos_small   THEN Tegangan = pos_low
      10. IF posisi = zero    AND sudut = zero        THEN Tegangan = zero
   */

   ////////////////////////////////IMPLIKASI DENGAN AND//////////////////////////////////////
    rule:



    imp1=fmin(DOMfar,DOMsudutzero);

    imp2=fmin(DOMfar,DOMnegsmall);

    imp3=fmin(DOMfar,DOMnegbig);

    imp4=fmin(DOMmedium,DOMnegsmall);

    imp5=fmin(DOMmedium,DOMnegbig);

    imp6=fmin(DOMmedium,DOMpossmall);

    imp7=fmin(DOMclose,DOMsudutzero);

    imp8=fmin(DOMclose,DOMnegsmall);

    imp9=fmin(DOMclose,DOMpossmall);

    imp10=fmin(DOMposisizero,DOMsudutzero);

   ////////////////////////////////AGREGASI OUTPUT////////////////////////////////////////////

    atas=(imp1*15)+(imp2*21)+(imp3*21)+(imp4*15)+(imp5*12)+(imp6*15)+(imp7*12)+(imp8*(-
10))+(imp9*12)+(imp10*0);
    bawah=(imp1+imp2+imp3+imp4+imp5+imp6+imp7+imp8+imp9+imp10)   ;

    out=atas/bawah;


    if (out>0)
      {
      //konfigurasi arah motor maju
      PORTB.0=1;
      PORTB.1=0;
      PORTB.2=1;
      PORTB.3=0;
      outocr=115-(5.66*out);  //ubah jadi nilai ocr
      }
    else if (out<0)
       {
       //konfigurasi motor mundur
       PORTB.0=0;
       PORTB.1=1;
       PORTB.2=0;
       PORTB.3=1;
       out=fabs(out);
       outocr=115-(5.66*out);  //ubah jadi nilai ocr
```

```
    }
  else if (out==0)
    {
     //konfigurasi motor berhenti
     PORTB.0=0;
     PORTB.1=0;
     PORTB.2=0;
     PORTB.3=0;
     }

  output=outocr; //ubah tipe float outocr jadi tipe int output



  return;
  }




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

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=Out Func4=Out Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=0 State4=0 State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x30;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1382.400 kHz
```

```
// Mode: Ph. correct PWM top=00FFh
// OC1A output: Inverted
// OC1B output: Inverted
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xF1;
TCCR1B=0x02;
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: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 691.200 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: None
```

```
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// LCD module initialization
lcd_init(16);

printf("FINAL PROJECT");
putchar(0x0D);
while (1)
      {
      // Place your code here
       fuzzy();

        //masukin ke OCR

       OCR1A=output;
       OCR1B=output;

        lcd_clear();
        sprintf(buffer1,"sdt:%2f",sudut);
        sprintf(buffer2,"pos:%2f",posisi);
        sprintf(buffer3,"out:%2f",out);
        printf("sdt:%2f       ",sudut);
        printf("pos:%2f       ",posisi);
        printf("out:%2f       ",out);
        putchar(0x0D);
        lcd_gotoxy(0,0);
        lcd_puts(buffer1);
        lcd_gotoxy(8,0);
        lcd_puts(buffer2);
        lcd_gotoxy(0,1);
        lcd_puts(buffer3);
        delay_ms(10);

      };
}
```

LAMPIRAN C
DATASHEET

SENSOR JARAK ULTRASONIK (PING)

PING)))™ Ultrasonic Distance Sensor (#28015)

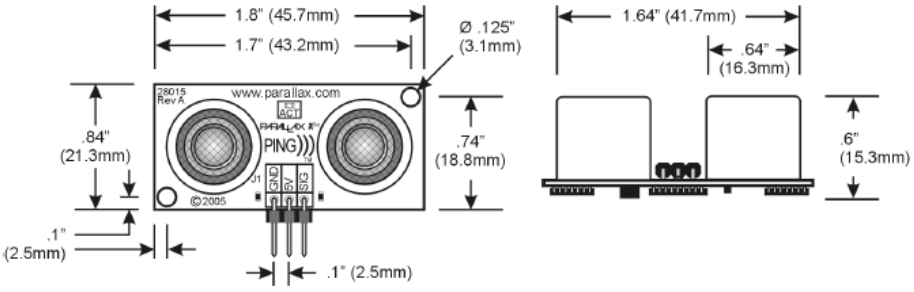
The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to BASIC Stamp® or Javelin Stamp modules, SX or Propeller chips, or other microcontrollers, requiring only one I/O pin.

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

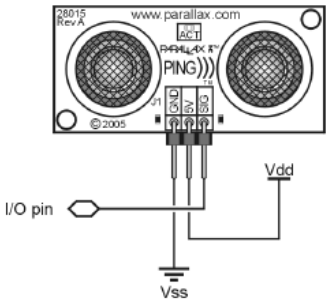
- Supply voltage – +5 VDC
- Supply current – 30 mA typ; 35 mA max
- Range – 2 cm to 3 m (0.8 inches to 3.3 yards)
- Input trigger – positive TTL pulse, 2 µs min, 5 µs typ.
- Echo pulse – positive TTL pulse, 115 µs to 18.5 ms
- Echo hold-off – 750 µs from fall of trigger pulse
- Burst frequency – 40 kHz for 200 µs
- Burst indicator LED shows sensor activity
- Delay before next measurement – 200 µs
- Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Operating temperature: 0 – 70° C.

Dimensions



Pin Definitions

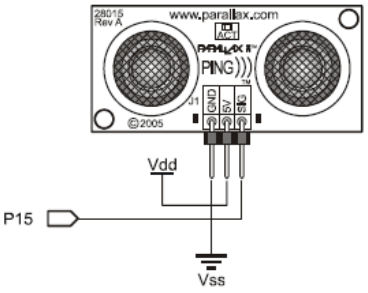
GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

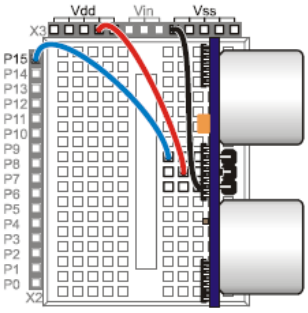


The PING))) sensor has a male 3-pin header used to supply ground, power (5 VDC) and signal. The header allows the sensor to be plugged into a solderless breadboard, or to be located remotely through the use of a standard servo extender cable (Parallax part #805-00002). Standard connections are shown in the diagram to the right.

Quick-Start Circuit

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.

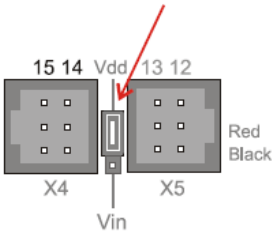




Extension Cable and Port Cautions

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

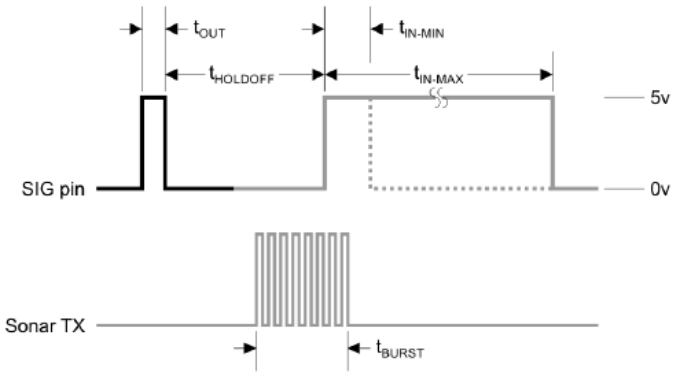
1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown. Then plug the cable into the 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

Theory of Operation

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 at about 1130 feet per second, 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.



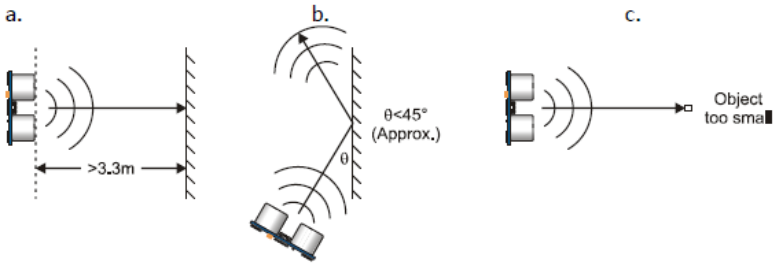
Host Device
PING))) Sensor

t_{OUT}	2 µs (min), 5 µs typical
$t_{HOLDOFF}$	750 µs
t_{BURST}	200 µs @ 40 kHz
t_{IN-MAX}	115 µs
t_{IN-MIN}	18.5 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. See the Water Level with PING))) document on the 28015 product page.

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:

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

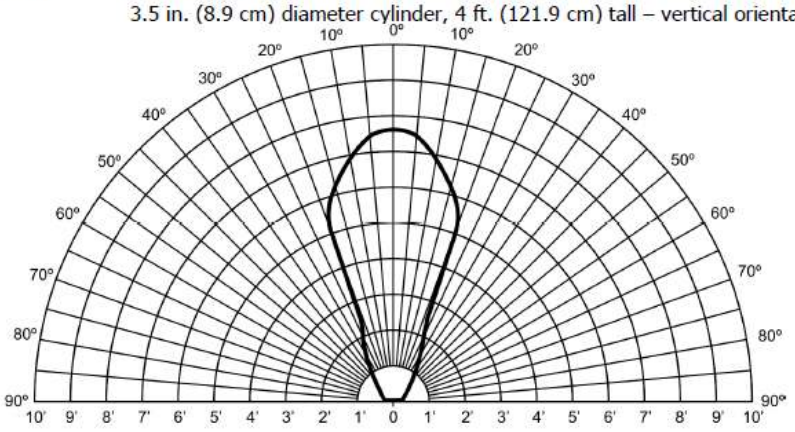
The percent error over the sensor's operating range of 0 to 70 ° C is significant, in the magnitude of 11 to 12 percent. The use of conversion constants to account for air temperature may be incorporated into your program (as is the case in the BS2 program below). Percent error and conversion constant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

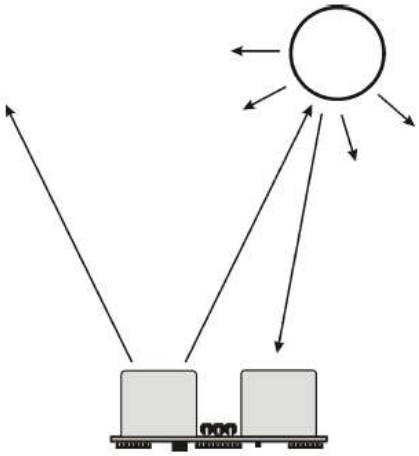
Test Data

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