Distance Calculator

Using LPC 1768 and Ultrasonic Sensor

Aditya Jajodia 150953050 aadityajajodia@gmail.com
Chaudhary Shyamal 150953066 chaudaryshyamal0908@gmail.com
Rishav Kumar 150953070 kumarrishav7625@gmail.com

INDEX

INTRODUCTION:

The purpose of this project is to calculate the distance of an obstacle by using an ultrasonic sensor. We have included the sketches regarding the working of the sensor as underlined in the methodology part.

The sensors are also widely used to measure distances. Thus they have provided a reliable source of obstacle detections. Since they are not vision-based, they are useful under conditions of poor lighting and transparent objects. However, ultrasonic sensors have limitations due to their wide beam-width, sensitivity to specular surfaces, and the inability to discern objects within 3 cm. Because of the typical specular nature of the ultrasonic waves reflection, only reflecting objects that are almost normal to the sensor acoustic axis may be accurately detected. The ultrasonic sensors have precision of less than 1 cm in distance measurements of up to 6m. However, the time of flight (ToF) measurement is the most accurate method among the measurements used (*). This ToF is the time elapsed between the emission and subsequent collection of a ultrasonic pulse train traveling at the speed of sound, which is approximately 340 m/s, after reflection from an object. For single measurement, this causes large response time, for example, 35ms for objects placed 6m away. In addition, they offer poor angular resolution. They have low cost and very simple to use.

*→Taken from reference #1

TECHNICAL SPECIFICATION:

1) Ultrasonic frequency
2) Maximal range 4m.
3) Minimal range 3cm.
4) Resolution 1cm.

The underlying phenomenon is using the Ultrasonic sensing principle.

A short ultrasonic pulse is transmitted at time 0, reflected by an object. The sensor receives this signal and converts into an electric signal. The next pulse can be transmitted when the echo is faded away. This time period is called cycle period. The recommended cycle period should not be less than 50ms.

This sensor cannot be used to detect the object made of a cloth, cotton and wool as these objects absorb majority of the waves. The time gap between the transmitted wave and the reflected wave is used to calculate the distance of the obstacle from the source using the formula

$$d = t * v/2$$

Where d = distance, t = time gap, v = velocity of sound in air.

*As velocity of sound in air is affected by the Temperature of the surrounding, it can be given by

$$v = 331.3 + 0.606 * T$$

Where v = apparent velocity, T = Room temperature ≈ 24°C

*Not used by us. We took speed of ultrasonic wave as 340 m/s

INNOVATIVENESS AND USEFULLNESS:

1) Completely insensitive to interfering factors (such as extraneous light, dust, smoke, mist, vapor, lint, oily air, etc.)
2) They are best suited for the detection of transparent and dark objects, reflective surfaces, shiny objects and of bulk materials and liquids.
3) Reliable detection of and measurement of objects, independent of their materials, color, transparency and texture.
4) The key part of application related to distance measurement by using ultrasonic sensor are:-
   - Automatic door opener
   - Electronic tape measurement
   - Robotics

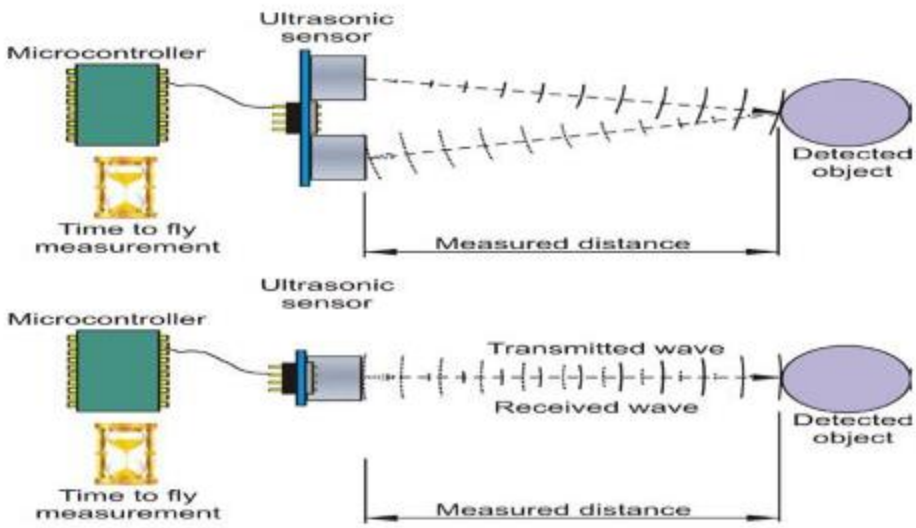
*→Taken from reference #2

REFERENCE & LITERATURE SURVEY:

1. http://waset.org/publications/6833/using-ultrasonic-and-infrared-sensors-for-distance-measurement
2. services@elefreaks.com

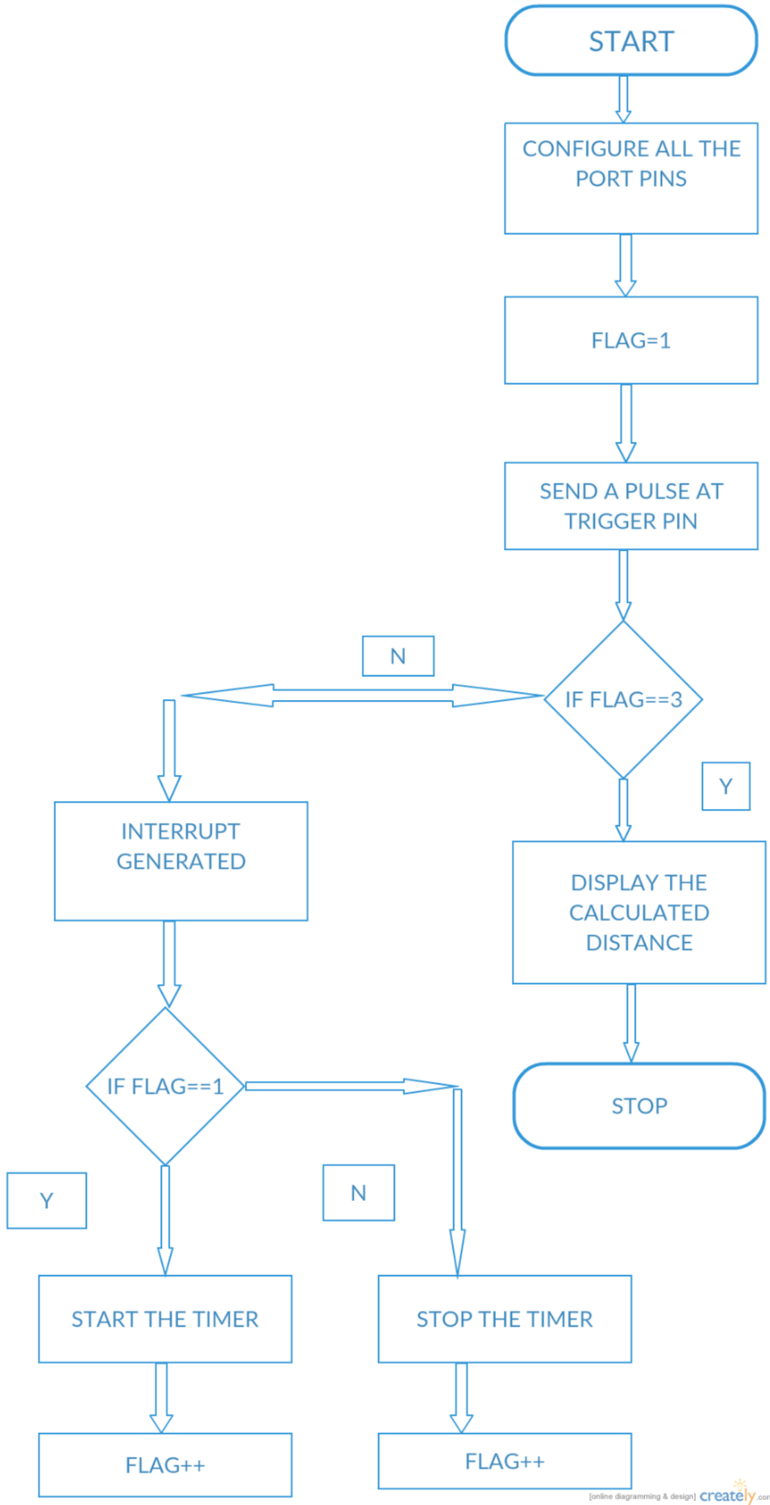
METHODOLOGY:

In this work, distance of the object is measured through ultrasonic distance sensor and the sensor output is connected to signal conditioning unit and after that it is processed through LPC 1768 ARM microcontroller. The measured results are displayed in liquid crystal display. This device is used to measure the distance from an object. It can detect objects that are within a range of 3cm – 350cm. The device uses two digital pins to communicate the distance found. Ultrasonic Range Detection Sensor, works by sending an ultrasound pulse at around 40 KHz, it then waits and listens for the pulse to echo back, calculating the time taken in microseconds. We can trigger a pulse as fast as 20 times a second and it can determine objects up to 3 metres away and as near as 3cm. The snapshot of the sensor and working process of the sensor is given below. The sensor needs a 5V power supply to run. The Timing diagram is shown in the figure pulse is required to the trigger input and start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. **Then calculate the range through the time interval between rising edge and falling edge of echo pin**. We suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal. There are only four connections, +5v, Ground, trigger and Echo.

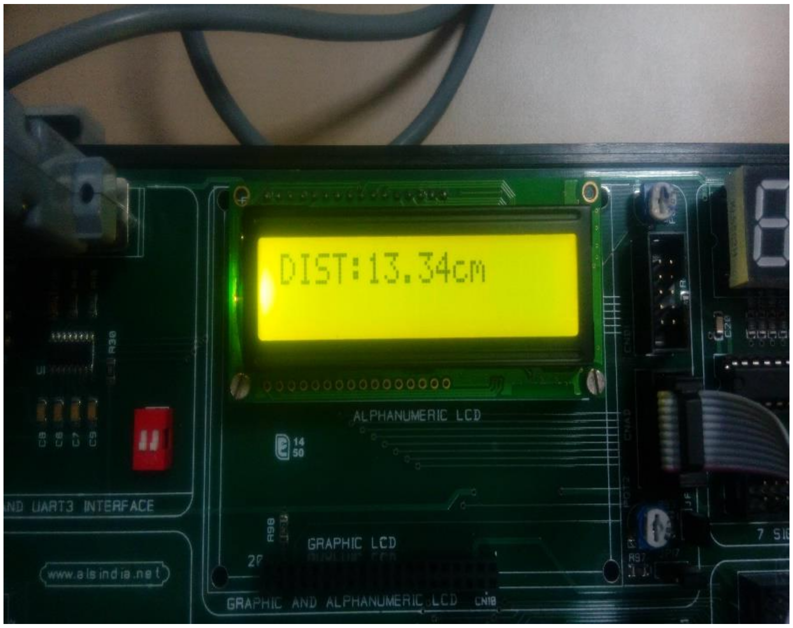


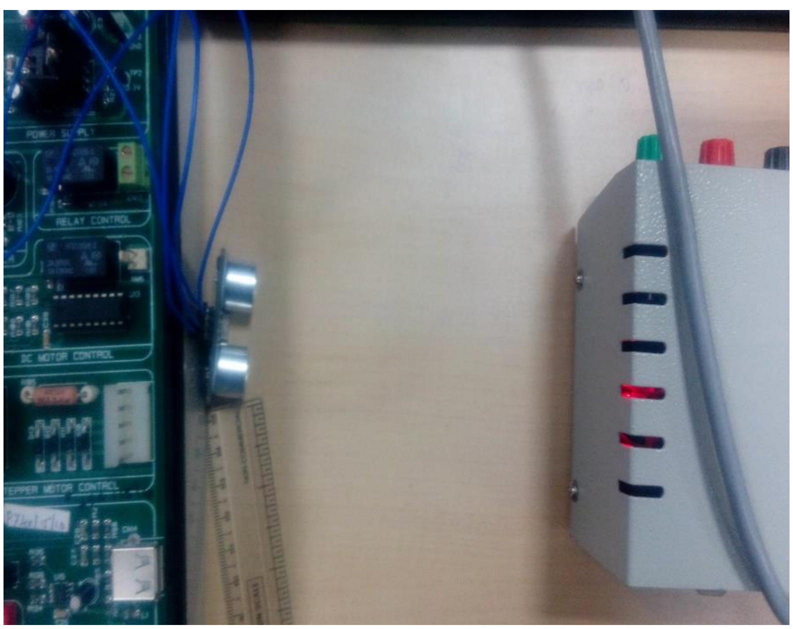
The Ultra Sonic sensor works as a burst signal is transmitted for short duration (is emitted) by the emitter. After that there will be a silent period. This period is actually called “response time” and is the time waiting for reflected waves. The acoustic emitted signal may find an obstacle or not. If an obstacle is found, the acoustic signal will be bounced back from the obstacle. This back bounced signal is called “echo”. The echo is received by the receiving transducer and is converted into electrical signal. Usually this signal is amplified, filtered and can be converted into digital format. Using the elapsed time between transmission and reception, the distance between the source and destination can easily be calculated.

Pseudo Code and Flow Chart



SNAPSHOT:


DIST:13.34cm
ALPHANUMERIC LCD
GRAPHIC LCD
www.aisindia.net
GRAPHIC AND ALPHANUMERIC LCD


RELAY CONTROL
DC MOTOR CONTROL

APPENDIX (CODE):

```
//ESD mini project to calculate distance

#include<lpc17xx.h>

#include<stdio.h>

#include<stdlib.h>

void lcd_init();//function to initialize lcd

void lcd_cmd_wrt(int);//function for commands

void lcd_data_wrt(int );//function for data

void delay();//delay

void lcd_string(char *);//function to write char array

void clear_ports();//for clearing ports

char dismessage[]={"distance in cm"};

char message1[]={"abcdefgh"};

unsigned int x,i=0,j=0,flag=1;

double value,distance;

int main(void)

{

        SystemInit();

        SystemCoreClockUpdate();

        LPC_PINCON->PINSEL1=0x0;

        LPC_GPIO0->FIODIR=0x3F<<23;

        lcd_init();  // External Interrupt at P2.11 (for ECHO ) and GPIO at P2.10 (for
TRIGGER)
```

```
 LPC_PINCON->PINSEL4=1<<22; // Configuring P2.11 as Fun1 and P2.10 as
GPIO

LPC_GPIO2->FIODIR=0x1<<10;     //External Interrupt Configuration

 //LPC_SC->EXTINT

  LPC_SC->EXTMODE=0x2;

 LPC_SC->EXTPOLAR=0x02;

//Timer configuration

LPC_TIM0->TCR=0x02;

LPC_TIM0->CTCR=0x00;

LPC_TIM0->MCR=0x1;

LPC_TIM0->PR=0x2;

 lcd_cmd_wrt(0x01);

lcd_cmd_wrt(0x80);

lcd_string(dismessage);

delay();

lcd_cmd_wrt(0xC0);

 while(1)

{

LPC_GPIO2->FIOPIN=0;

for(i=0;i<10000;i++);

LPC_GPIO2->FIOPIN=0x1<<10;

for(i=0;i<25000;i++);

LPC_GPIO2->FIOPIN=0;

NVIC_EnableIRQ(EINT1_IRQn);
```

```
            while(flag!=3); // Waiting for the interrupt
            delay();
            //value = LPC_TIM0->TC;
            //distance = (value/1000000.0)*340;
            distance=value;
            sprintf(message1,"%3.2f",distance);
             lcd_cmd_wrt(0xC0);
            lcd_string("       ");
            lcd_cmd_wrt(0xC0);
            lcd_string(message1);
            flag=1;
            for(i=0;i<25000;i++);
            }
            }
void EINT1_IRQHandler(void)
{
            LPC_SC->EXTPOLAR=~(LPC_SC->EXTPOLAR);
            if(flag==1)
            {
          flag++;
          LPC_TIM0->TCR=0x1;
              }
        else if(flag==2)
```

```
	{
		LPC_TIM0->TCR=0x00;
		flag++;
		value=LPC_TIM0->TC/58.0;
		LPC_TIM0->TCR=0x02;
		//LPC_SC->EXTMODE=0x2;
		}
		LPC_SC->EXTINT=0xFF;
	}
void lcd_string(char *buff)
{
	i=0;
	while(buff[i]!='\0')
	{
		//lcd_data_wrt(0x34);
		lcd_data_wrt(buff[i]);
		delay();
		i++;
	}
}
void lcd_init()
{
	for(i=0;i<3;i++)
```

```
{
	LPC_GPIO0->FIOCLR=0x1<<27;
	LPC_GPIO0->FIOPIN=0x03<<23;
	for(j=0;j<30000;j++);
	LPC_GPIO0->FIOSET=0x1<<28;
	for(j=0;j<100;j++);
	LPC_GPIO0->FIOCLR=0x1<<28;
	for(j=0;j<30000;j++);

}
LPC_GPIO0->FIOPIN=0x2<<23;
LPC_GPIO0->FIOSET=0x1<<28;
for(j=0;j<100;j++);
LPC_GPIO0->FIOCLR=0x1<<28;
delay();
lcd_cmd_wrt(0x28);
delay();
lcd_cmd_wrt(0x06);
delay();
lcd_cmd_wrt(0x0E);
delay();
lcd_cmd_wrt(0x80);
delay();
```

```
}
void lcd_cmd_wrt(int y)
{
	clear_ports();
	LPC_GPIO0->FIOCLR=0x1<<27;
	x=y&0xF0;
	LPC_GPIO0->FIOPIN=x<<19;
	LPC_GPIO0->FIOSET=0x1<<28;
	for(j=0;j<100;j++);
	LPC_GPIO0->FIOCLR=0x1<<28;
	delay();
	x=y&0x0F;
	LPC_GPIO0->FIOPIN=x<<23;
	LPC_GPIO0->FIOSET=0x1<<28;
	for(j=0;j<100;j++);
	LPC_GPIO0->FIOCLR=0x1<<28;
	delay();
}
void lcd_data_wrt(int y)
{
	clear_ports();
	LPC_GPIO0->FIOSET=0x1<<27;
	x=y&0xF0;
```

```
        LPC_GPIO0->FIOPIN|=x<<19;
        LPC_GPIO0->FIOSET=0x1<<28;
        for(j=0;j<100;j++);
        LPC_GPIO0->FIOCLR=0x1<<28;
        delay();
        clear_ports();
        LPC_GPIO0->FIOSET=0x1<<27;
        x=y&0x0F;
        LPC_GPIO0->FIOPIN|=x<<23;
        LPC_GPIO0->FIOSET=0x1<<28;
        for(j=0;j<100;j++);
        LPC_GPIO0->FIOCLR=0x1<<28;
        delay();
}
void delay()
{
        for(j=0;j<800;j++);
}
void clear_ports()
{
        LPC_GPIO0->FIOPIN=0x0;
}
```