

Distance Measurement Using Ultrasonic Sensor (HC-SRO4)



PROJECT BY:- PROJECT FROM:-

Rishav Gupta (150953224) (Roll no. 55) Mr. Santhosha Rao

Krish Kalro (150953222) (Roll no. 54) Assistant Professor-

Amit Kumbar (150953216) (Roll no. 53) Selection Grade,ICT Dept.

Contents

|  |  |  |
| --- | --- | --- |
| **Serial Number** | **TOPIC** | **Page Number** |
| **1** | Introduction | **1** |
| **2** | Objectives | **2** |
| **3** | Working Principle | **5** |
| **4** | Methodology of Hardware | **7** |
| **5** | Code | **11** |
| **6** | Results | **17** |
| **7** | References | **21** |
| **8** | Conclusion | **22** |

Introduction

Ultrasonic waves in today’s industry play a key role in a wide range of applications from medical, to automobile to military uses. Most of the uses like SONAR and RADAR use the reflection of sound waves through the medium or echo to sense the distance between objects and the transmitter. In this project, we use the HC-SRO4 sensor to find the distance of an object in its path from the sensor. Sound which travels at the speed of 220m/s in air is mapped as it travels to and fro from a particular object to accurately calculate the distance from the object.

Ultrasonic sensors are widely used in cars as parking sensors to aid the driver in reversing into parking spaces. They are being tested for a number of other automotive uses including ultrasonic people detection and assisting in autonomous UAV navigation. They are even installed in industrial robots as an enabler to their artificial intelligence as it allows them to detect objects in their path. In medicine ultrasonic sensors are used to get clear scans of organs through scans such as echo-cardiograms which map the organs of the body with the use of ultrasonic waves. High and ultra-high ultrasound waves are used in acoustic microscopy, with frequencies up to 4 gigahertz. Ultrasonic imaging applications include industrialnon-destructive testing, quality control

Since these ultrasonic sensors use sound rather than light for detection, they work in applications where photoelectric sensors cannot be used. Ultrasonics are a great solution to detect clear objects, clear labels and even for liquid level measurement, applications that photo electrics struggle with because of target translucence. Also, the target colour and/or reflectivity do not affect ultrasonic sensors, which can operate reliably in high-glare environments.

Objectives of Range Sensing

Range sensing have been used in the following applications:

* Organ/mass detection and scanning in medical fields
* Chemical viscosity and testing experiments for industrial purposes
* Artificial intelligence for robots, automatic cars, reversing mechanisms in automobiles, automatic lights on detection of movement and people
* Water level sensing
* Automatic wheelchairs to improve quality of life for the disabled
* Plant protection systems in vineyards
* Automatic lighting and sensing in security systems

Machinery and processes in a wide range of industries use distance measurement sensors where size or position feedback is required: printing, converting, robotics, material handling, transportation and more. Here are just some of the ways they are used:

**Positioning & Locating**

Distance measurement sensors are used to control or indicate the position of objects and materials.

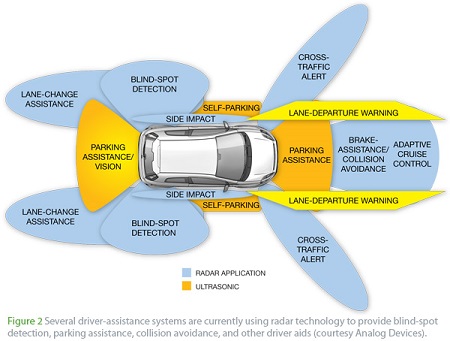
**Dimensioning & Selecting**

Distance measurement sensors can determine the dimensions of objects such as height, width and diameter, using one or more sensors. Items can be selected or rejected based on their dimensions or profiles, such as determining whether an object on a conveyor is upside down or not, by using the sensor’s timing and distance features.

**Profiling & Multi-Sensor Systems**

Applications requiring multiple measurements are handled by networks or groups of standard sensors.

A rather interesting application is the development of the autonomous or driver-less cars which use technology such as radar, laser light, GPS, odometer, and computer vision to detect objects, obstacles and even people in its path. These cars if perfectly operational will help ease traffic, reduce pollution, and even a large reduction in crime and accidents.



A model of the proposed Tesla autonomous vehicle.

Working Principle

The model created using HC-SRO4 is a non-contact measurement function which ranges between 2cm-400cm and uses the ARM controller’s LCD to display the distance of the object rounded up to the nearest millimeter.

Wire connections are as follows:

* 5V Supply
* Trigger Pulse Input
* Echo Pulse Output
* 0V Ground

The basic working principle is that:

* An IO trigger is used for at least 10us high level signal,
* The Module automatically sends eight 40 kHz signals and detects whether there is a pulse signal back.
* IF the signal comes back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time × velocity of sound (340M/S) / 2

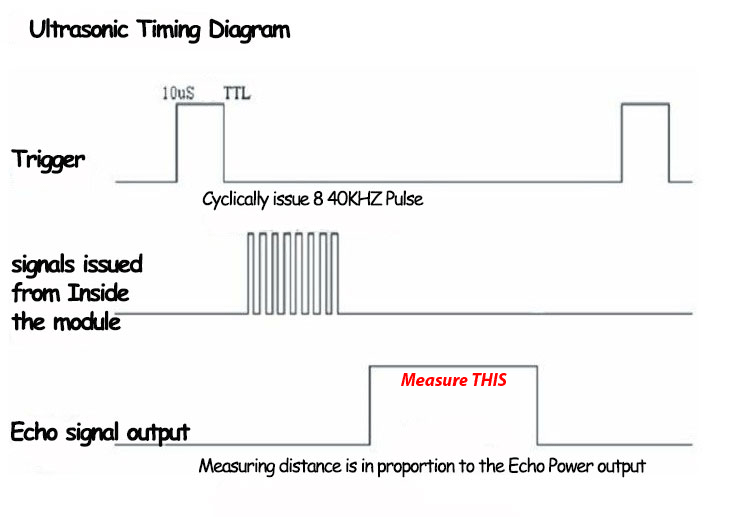
TIMING DIAGRAM:-

In the Timing diagram shown below, a short 10uS pulse is supplied to the trigger input to start the ranging, and then the module sends out an 8 cycle burst of ultrasound at 40 kHz and raises its echo. The Echo is a measurable distance object that has pulse width and the range in proportion .The range is calculated using the time interval that is between sending trigger signal and receiving echo signal.

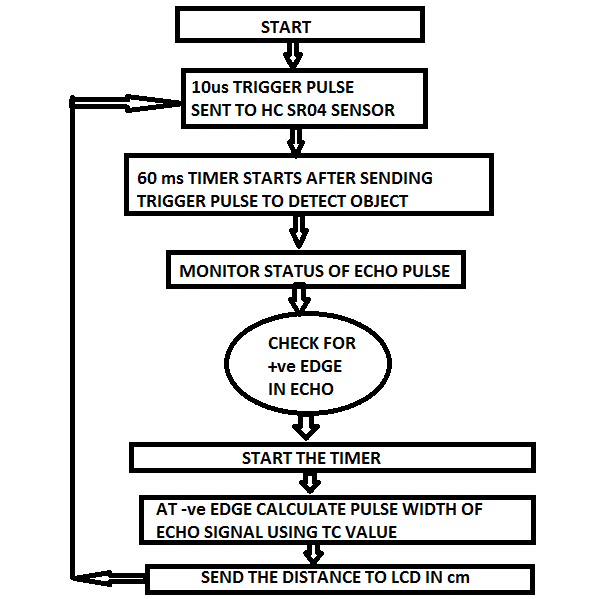
Formula: us / 58 = centimeters or us/ 148 =inch;

OR:

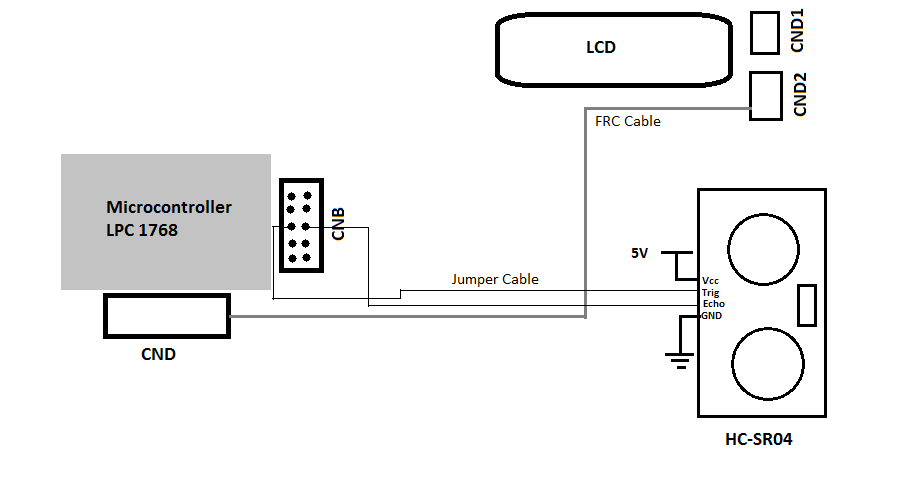
Range = high level time \* velocity (340M/S) / 2



Methodology of Hardware

FloWchart:- 

CIRCUIT Diagram:-



PSEUDO-CODE:-

//TIMER0 IS FOR TRIGGER INPUT

//TIMER1 IS FOR TRIGGER OUTPUT

unsigned char dist[20]; // array for putting distance // value in lcd

unsigned char Msg3[] = {"Dist:"};

unsigned char Msg4[] = {"WELCOME"};

unsigned char Msg5[] = {"RANGE FINDER!"};

//Initializing messages to be displayed on LCD

void TIMER0\_IRQHandler(void) {

{

// TO SEND PULSE FOR 10us IN TRIGGER

}

void TIMER3\_IRQHandler(void)

{

// TO START 60ms TIMER SO THAT IF NO OBJECT IS DETECTED IT STARTS THE PROCESS ALL OVER AGAIN

}

void EINT1\_IRQHandler(void)// TO GET THE LENGTH OF // THE PULSE IN ECHO FOR // DISTANCE CALCULATION

{

//clear EINT1 interr

// SO THAT NOW A POSITIVE EGDE IS ENCOUNTERED THUS 60NS TIMER CAN BE DISABLED

// GETTING WHICH EGDE TRIGGER IS ENCOUNTERED +ve OR -ve

//EINT1 status extract

// POSITIVE EGDE ENCOUNTERED

else

// NEGETIVE EDGE ENCOUNTERED

{

// CAPTURING VALUE OF TC

x=j/58.0; // us/58 OR HighLevelTime\*340/2

temp1 = 0x85;

lcd\_com();

delay\_lcd(800);

sprintf(dist,"%3.2fcm ",x); //CONVERTING FLOAT // INTO STRING

lcd\_puts(&dist[0]); // SHOW DISTANCE IN LCD

delay\_lcd(80000);

}

}

int main(void)

{

lcd\_init();

// LCD display welcome

delay\_lcd(2000000);

clr\_disp(); // clear LCD

temp1 = 0x80;

lcd\_com();

delay\_lcd(800);

// LCD display Range Finder!

delay\_lcd(2000000);

clr\_disp(); // clear LCD

LPC\_PINCON->PINSEL4|=0x00000000; //P2.10 AS TRIGGER

LPC\_PINCON->PINSEL4=(1<<22); // P2.11 AS FUNCTION-1 FOR EXT INTERRUPT

LPC\_GPIO2->FIODIR=0x400;// SETTING P2.10 AS OUTPUT

temp1 = 0x80;

lcd\_com();

delay\_lcd(800);

// LCD display Dist:

// configure external interrupt for positive edge Interrupt

//configure timer0 for 10us trigger pulse

// configuring timer1 for capturing TC value in us

// configuring timer3 for 60ms timer after pulse // is generated from trigger so that if within // 60ms no object is found restart the whole // process

// TIMER0 FOR 10us TRIGGER PULSE

NVIC\_EnableIRQ(EINT1\_IRQn);

NVIC\_EnableIRQ(TIMER0\_IRQn);

NVIC\_EnableIRQ(TIMER3\_IRQn);

// EDGE CONFIGURATION FOR EINT1

// CONFIGURE RISING EDGE FOR EINT1

// staring 10 micro-second timer

while(1); //KEEPS PROGRAM RUNNING

}

CODE

LCD CODE:-

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*-------------------------------------------------------------------------

\* Controller : LPC1768

\* Project : ALS-SDA-AMCTXM3-01

\* Port lines used: Data1 to Data4 - P0.23 to P0.26

\* En - P0.28. RS - P0.27, RW - Ground

\* Connection : CND to CNAB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <lpc17xx.h>

#include "AN\_LCD.h"

unsigned long int temp1=0, temp2=0 ;

//lcd initialization

void lcd\_init()

{

/\* Ports initialized as GPIO \*/

// LPC\_PINCON->PINSEL0 &= 0x00000000; //P0.23 to P0.28

/\* Setting the directions as output \*/

LPC\_GPIO0->FIODIR |= DT\_CTRL;

LPC\_GPIO0->FIODIR |= RS\_CTRL;

LPC\_GPIO0->FIODIR |= EN\_CTRL;

clear\_ports();

delay\_lcd(10000);

temp2 = (0x30<<19);

wr\_cn();

delay\_lcd(25000);

temp2 = (0x30<<19);

wr\_cn();

delay\_lcd(25000);

temp2 = (0x30<<19);

wr\_cn();

delay\_lcd(25000);

temp2 = (0x20<<19);

wr\_cn();

delay\_lcd(25000);

temp1 = 0x28;

lcd\_com();

delay\_lcd(25000);

temp1 = 0x0C;

lcd\_com();

delay\_lcd(20800);

temp1 = 0x06;

lcd\_com();

delay\_lcd(28000);

temp1 = 0x01;

lcd\_com();

delay\_lcd(5000);

temp1 = 0x80;

lcd\_com();

delay\_lcd(800);

return;

}

void lcd\_com(void)

{

temp2 = temp1 & 0xf0;//move data (26-8+1) times : 26 - HN place, 4 - Bits

temp2 = temp2 << 19;//data lines from 23 to 26

wr\_cn();

temp2 = temp1 & 0x0f; //26-4+1

temp2 = temp2 << 23;

wr\_cn();

delay\_lcd(1000);

return;

}

// command nibble o/p routine

void wr\_cn(void) //write command reg

{

clear\_ports();

LPC\_GPIO0->FIOPIN = temp2; // Assign the value to the data lines

LPC\_GPIO0->FIOCLR = RS\_CTRL; // clear bit RS

LPC\_GPIO0->FIOSET = EN\_CTRL; // EN=1

delay\_lcd(25);

LPC\_GPIO0->FIOCLR = EN\_CTRL; // EN =0

return;

}

// data o/p routine which also outputs high nibble first

// and lower nibble next

void lcd\_data(void)

{

temp2 = temp1 & 0xf0;

temp2 = temp2 << 19;

wr\_dn();

temp2= temp1 & 0x0f;

temp2= temp2 << 23;

wr\_dn();

delay\_lcd(1000);

return;

}

// data nibble o/p routine

void wr\_dn(void)

{

clear\_ports();

LPC\_GPIO0->FIOPIN = temp2; // Assign the value to the data lines

LPC\_GPIO0->FIOSET = RS\_CTRL; // set bit RS

LPC\_GPIO0->FIOSET = EN\_CTRL; // EN=1

delay\_lcd(25);

LPC\_GPIO0->FIOCLR = EN\_CTRL; // EN =0

return;

}

void delay\_lcd(unsigned int r1)

{

unsigned int r;

for(r=0; r<r1; r++);

return;

}

void clr\_disp(void)

{

temp1 = 0x01;

lcd\_com();

delay\_lcd(10000);

return;

}

void clear\_ports(void)

{

/\* Clearing the lines at power on \*/

LPC\_GPIO0->FIOCLR = DT\_CTRL; //Clearing data lines

LPC\_GPIO0->FIOCLR = RS\_CTRL; //Clearing RS line

LPC\_GPIO0->FIOCLR = EN\_CTRL; //Clearing Enable line

return;

}

void lcd\_puts(unsigned char \*buf1)

{

unsigned int i=0;

while(buf1[i]!='\0')

{

temp1 = buf1[i];

lcd\_data();

i++;

if(i==16)

{

temp1 = 0xc0;

lcd\_com();

}

}

return;

}

SENSOR-CODE:-

#include<LPC17xx.h>

#include<stdio.h>

#include"AN\_LCD.h"

//TIMER0 IS FOR TRIGGER INPUT

//TIMER1 IS FOR TRIGGER OUTPUT

float x;

unsigned int i,j;

unsigned char dist[20]; // array for putting distance //value in lcd

unsigned char Msg3[] = {"Dist:"};

unsigned char Msg4[] = {"WELCOME"};

unsigned char Msg5[] = {"RANGE FINDER!"};

void TIMER0\_IRQHandler(void) // TO SEND PULSE FOR //10us IN TRIGGER

{

LPC\_TIM0->IR=0x01;

LPC\_GPIO2->FIOCLR=0x00000400;// CLEARING //TRIGGER

LPC\_TIM0->TCR=0x00;

LPC\_TIM3->TCR=0x02;

LPC\_TIM3->TCR=0x01;

}

void TIMER3\_IRQHandler(void)// TO START 60ms TIMER //SO THAT IF NO OBJECT //IS DETECTED IT STARTS //THE PROCESS ALL //OVER AGAIN

{

LPC\_TIM3->IR=0x01;

LPC\_TIM3->TCR=0x00;

LPC\_TIM0->TCR=0x02;

LPC\_TIM0->TCR=0x01;

LPC\_GPIO2->FIOSET=0x00000400;// SETTING TRIGGER

}

void EINT1\_IRQHandler(void)// TO GET THE LENGTH OF //THE PULSE IN ECHO //FOR DISTANCE //CALCULATION

{

LPC\_SC->EXTINT=0x02;//clear EINT1 interr

LPC\_TIM3->TCR=0x00; // SO THAT NOW A POSITIVE //EGDE IS ENCOUNTERED //THUS 60NS TIMER CAN BE //DISABLED

i=LPC\_SC->EXTPOLAR; // GETTING WHICH EGDE //TRIGGER IS ENCOUNTERED //+ve OR -ve

LPC\_TIM1->TCR=0x00;

x=0.0;

i&=0x02;//EINT1 status extract

i>>=1;

if(i==0x01)// POSITIVE EGDE ENCOUNTERED

{

LPC\_SC->EXTPOLAR=0x00;

LPC\_TIM1->TCR=0x02;

LPC\_TIM1->TCR=0x01; //STARTNG TIMER FOR //DISTANCE CALCULATION

}

else// NEGETIVE EDGE ENCOUNTERED

{

LPC\_SC->EXTPOLAR=0x02;

j=LPC\_TIM1->TC; // CAPTURING VALUE OF TC

x=j/58.0; // us/58 OR HighLevelTime\*340/2

temp1 = 0x85;

lcd\_com();

delay\_lcd(800);

sprintf(dist,"%3.2fcm ",x); //CONVERTING FLOAT //INTO STRING

lcd\_puts(&dist[0]); // SHOW DISTANCE IN LCD

delay\_lcd(80000);

}

LPC\_TIM0->TCR=0x02;

LPC\_TIM0->TCR=0x01;

LPC\_GPIO2->FIOSET=0x00000400;// SETTING TRIGGER

}

int main(void)

{

SystemInit();

SystemCoreClockUpdate();

lcd\_init();

temp1 = 0x80;

lcd\_com();

delay\_lcd(800);

lcd\_puts(&Msg4[0]);// to display welcome

delay\_lcd(2000000);

clr\_disp();// clear LCD

temp1 = 0x80;

lcd\_com();

delay\_lcd(800);

lcd\_puts(&Msg5[0]);// to display Range Finder!

delay\_lcd(2000000);

clr\_disp();// clear LCD

LPC\_PINCON->PINSEL4|=0x00000000; //P2.10 AS TRIGGER

LPC\_PINCON->PINSEL4=(1<<22); // P2.11 AS FUNCTION-1 //FOR EXT INTERRUPT

LPC\_GPIO2->FIODIR=0x400;// SETTING P2.10 AS OUTPUT

temp1 = 0x80;

lcd\_com();

delay\_lcd(800);

lcd\_puts(&Msg3[0]);// to display Dist:

// configure external interrupt for positive edge //interrupt

LPC\_GPIO2->FIOPIN=0x00000000;

LPC\_SC->EXTMODE=0x02; // EDGE FOR EINT1

LPC\_SC->EXTPOLAR=0x02; // RISING EDGE FOR EINT1

LPC\_SC->EXTINT=0x0F;

//configure timer0 for 10us trigger pulse

LPC\_TIM0->TCR=0x02;

LPC\_TIM0->CTCR=0x00;

LPC\_TIM0->MR0=29;

LPC\_TIM0->PR=0;

LPC\_TIM0->MCR=0x03;

// configuring timer1 for capturing TC value in us

LPC\_TIM1->TCR=0x02;

LPC\_TIM1->PR=2; // FOR TC IN SECONDS

// configuring timer3 for 60ms timer after pulse is //generated from trigger so that if within 60ms //no object is found restart the whole process

LPC\_TIM3->TCR=0x02;

LPC\_TIM3->MR0=179999;

LPC\_TIM3->EMR=0x30;

LPC\_TIM3->PR=0;

LPC\_TIM3->MCR=0x03;

// TIMER0 FOR 10us TRIGGER PULSE

NVIC\_EnableIRQ(EINT1\_IRQn);

NVIC\_EnableIRQ(TIMER0\_IRQn);

NVIC\_EnableIRQ(TIMER3\_IRQn);

LPC\_SC->EXTMODE=0X02; // EDGE FOR EINT1

LPC\_SC->EXTPOLAR=0X02; // RISING EDGE FOR EINT1

LPC\_GPIO2->FIOCLR = 0X00000400;

LPC\_TIM0->EMR=0x30;

LPC\_TIM3->TCR=0x02;

LPC\_TIM3->TCR=0x01; // staring 10us timer

while(1);

}

Results

**Electric Parameter** (Theoretical)

Working Voltage DC 5 V

Working Current 15mA

Working Frequency 40Hz

Max Range 4m

Min Range 2cm

Measuring Angle 15 degree

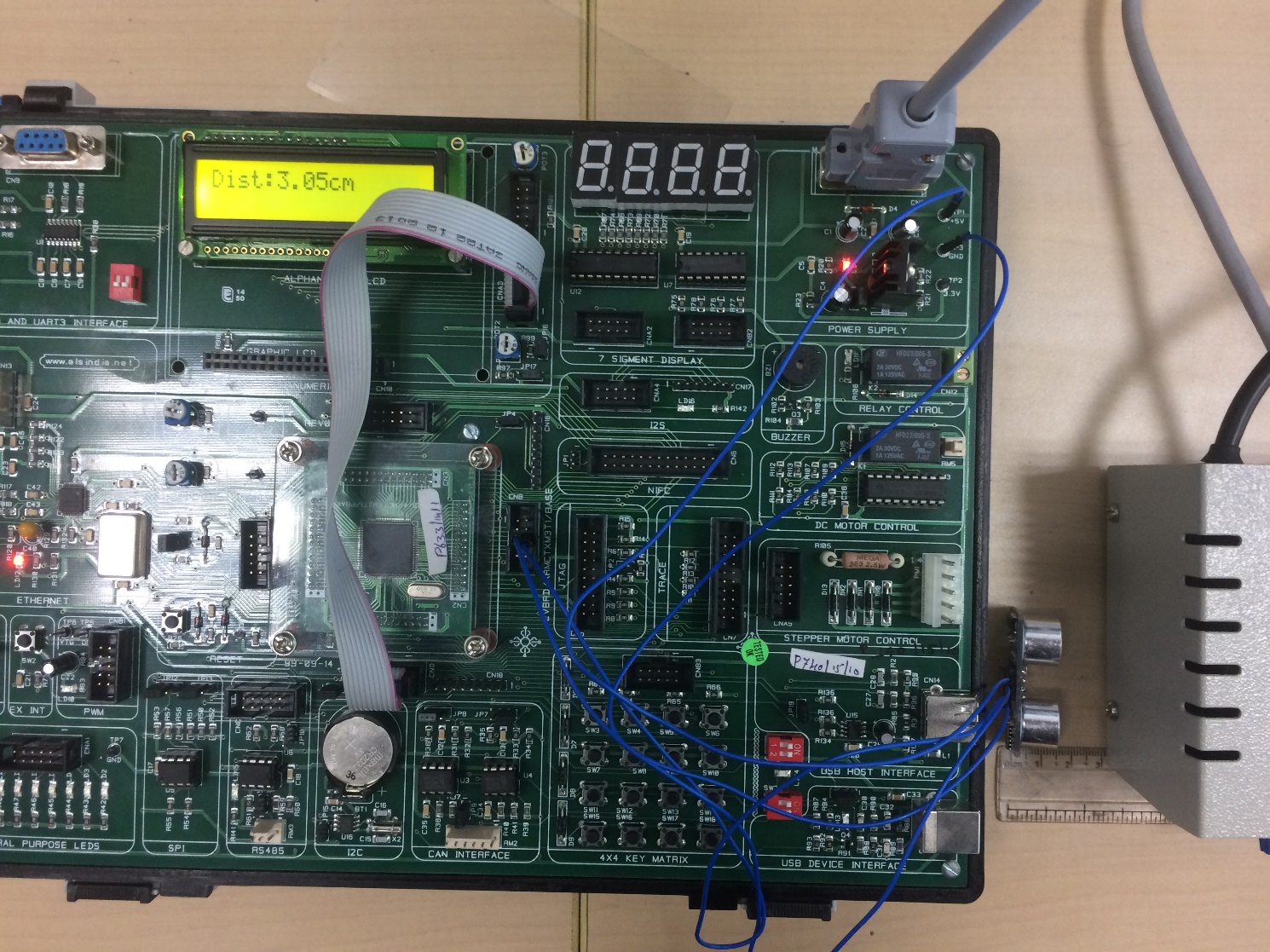
Trigger Input Signal 10uS TTL pulse

Echo Output Signal Input TTL lever signal and the range in proportion

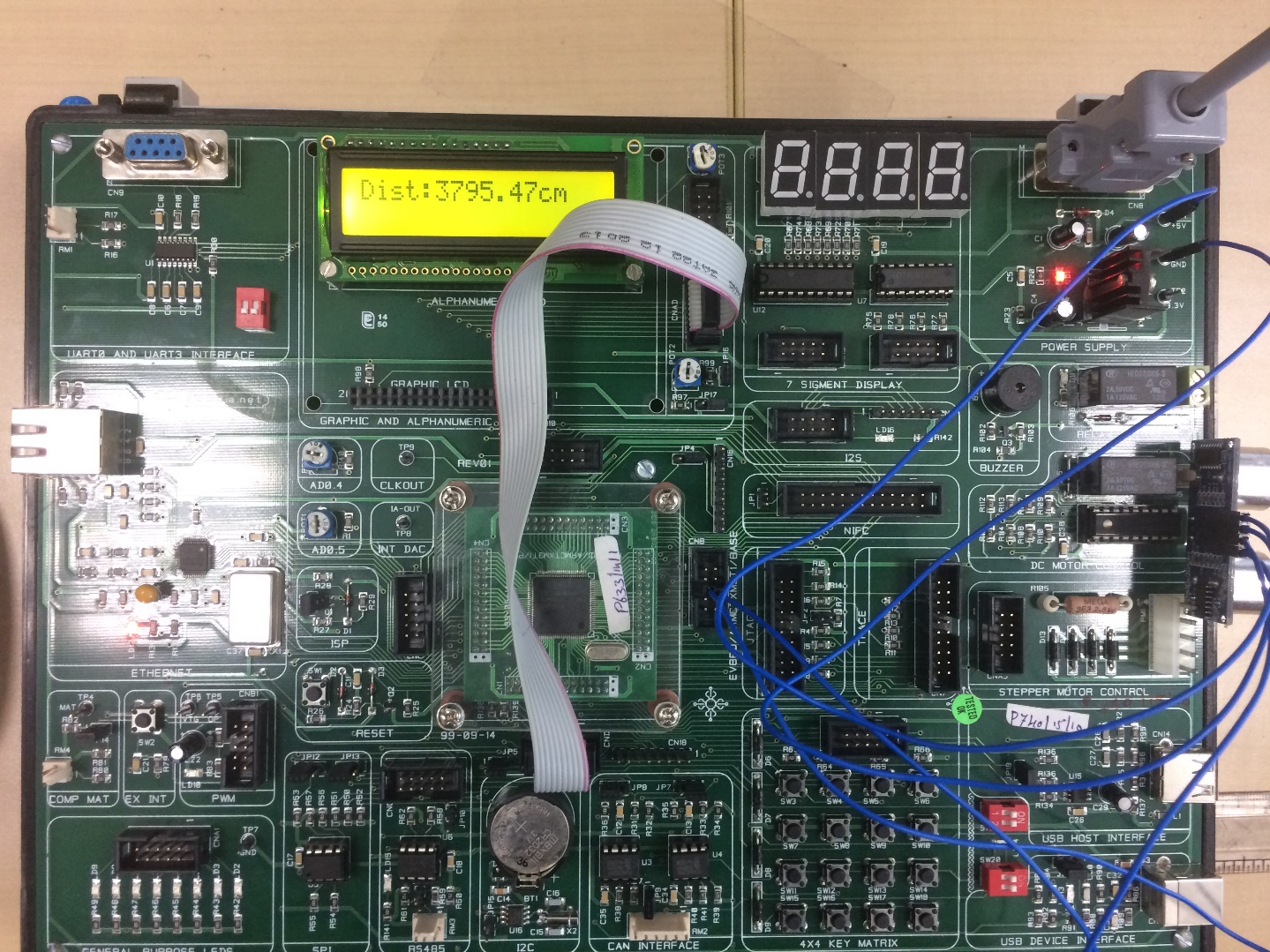
Dimension 45\*20\*15mm

**Practical Parameters:**

Minimum distance measured: 3.05cm



Maximum distance measured: 3795.47cm



Distance measured to check correctness: 15ccm



Degree of correctness = 100\* (Theoretical range – Practical Range)/Theoretical Range %

Degree of correctness =100- 100\*(399.80-372.42)/399.80

Therefore, Degree of correctness = 94.86%

References

<http://venuembedded.weebly.com/blog/ultrasonic-hc-sr04-with-8051>

<https://circuitdigest.com/microcontroller-projects/distance-measurement-using-hc-sr04-avr>

<http://www.gadgetronicx.com/ultrasonic-distance-meter-hcsr04-8051/>

<http://www.ijettjournal.org/volume-9/number-15/IJETT-V9P348.pdf>

<http://www.electronicshub.org/ultrasonic-rangefinder-using-8051/>

<http://arduinobasics.blogspot.in/2012/11/hc-sr04-ultrasonic-sensor.html>

<https://electrosome.com/hc-sr04-ultrasonic-sensor-pic/>

<https://www.ijareeie.com/upload/2014/february/48_Rajan.pdf>

[http://www.armlearning.com/code/nucleo-and-hc-sr04-ultrasonic-sensor-example.php#codesyntax\_3](http://www.armlearning.com/code/nucleo-and-hc-sr04-ultrasonic-sensor-example.php" \l "codesyntax_3)

<https://developer.mbed.org/users/antoniolinux/code/HCSR04/file/86b2086be101/hcsr04.cpp>

<http://www.micropik.com/PDF/HCSR04.pdf>

<http://www.nxp.com/documents/user_manual/UM10360.pdf>

Conclusion

Over the past six months, we have been learning embedded systems and have been implementing programs on the ARM controller using the LPC 1768 Microchip. The programs we implemented were often singular and explored only one aspect of this chip as well as of embedded-c programming. This project has helped us to understand the synchronization of different applications of microcontrollers so as to serve a purpose such as finding the distance of an object a point using an ultrasonic sensor. We also used the setup to calculate the minimum and maximum distances that could be gauged so as to obtain a range for the sensor and to get a degree of correctness by comparing this range with the ideal values provided in the sensor’s user manual.

**RESULT:**

The Minimum distance that could be calculated was found to be 3.05cm

The Maximum distance that could be calculated was found to be 375.47cm

The correctness of the Ultrasonic Sensor was found to be 94.86%