# Embedded Systems Lab [ICT 3111] Project

## Team Details


**Batch No.** : 2
**Team No.** : 2
**Section** : A
**Branch** : Information Technology
**Members** : Ashish Jain - 190911126
Shruti Praveen Jain - 190911132
Y Veera Raghav Reddy - 190911128


## Problem Statement

Write a program to interface a Doppler sensor to LPC1768 and display the distance of an object on a 7-segment display.

## Hardware Components used

- LPC 1768 development board (with 4-digit 7-segment display attached to it)
- HC-SR04 ultrasonic sensor
- Resistors
- Breadboard and jumper wires

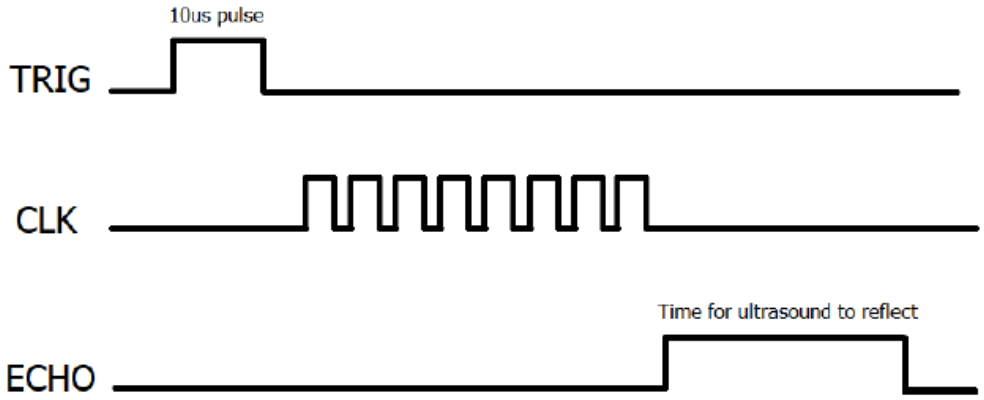
## Concept

A short pulse of 10us is applied to Trigger pin to start distance measurement. After receiving trigger pulse, the HC-SR04 Module sends a burst of 8 ultrasonic pulses at 40Khz. It will then output high on Echo for the amount to time taken for the sound waves to reach back. This time obtained is then used to calculate the distance.

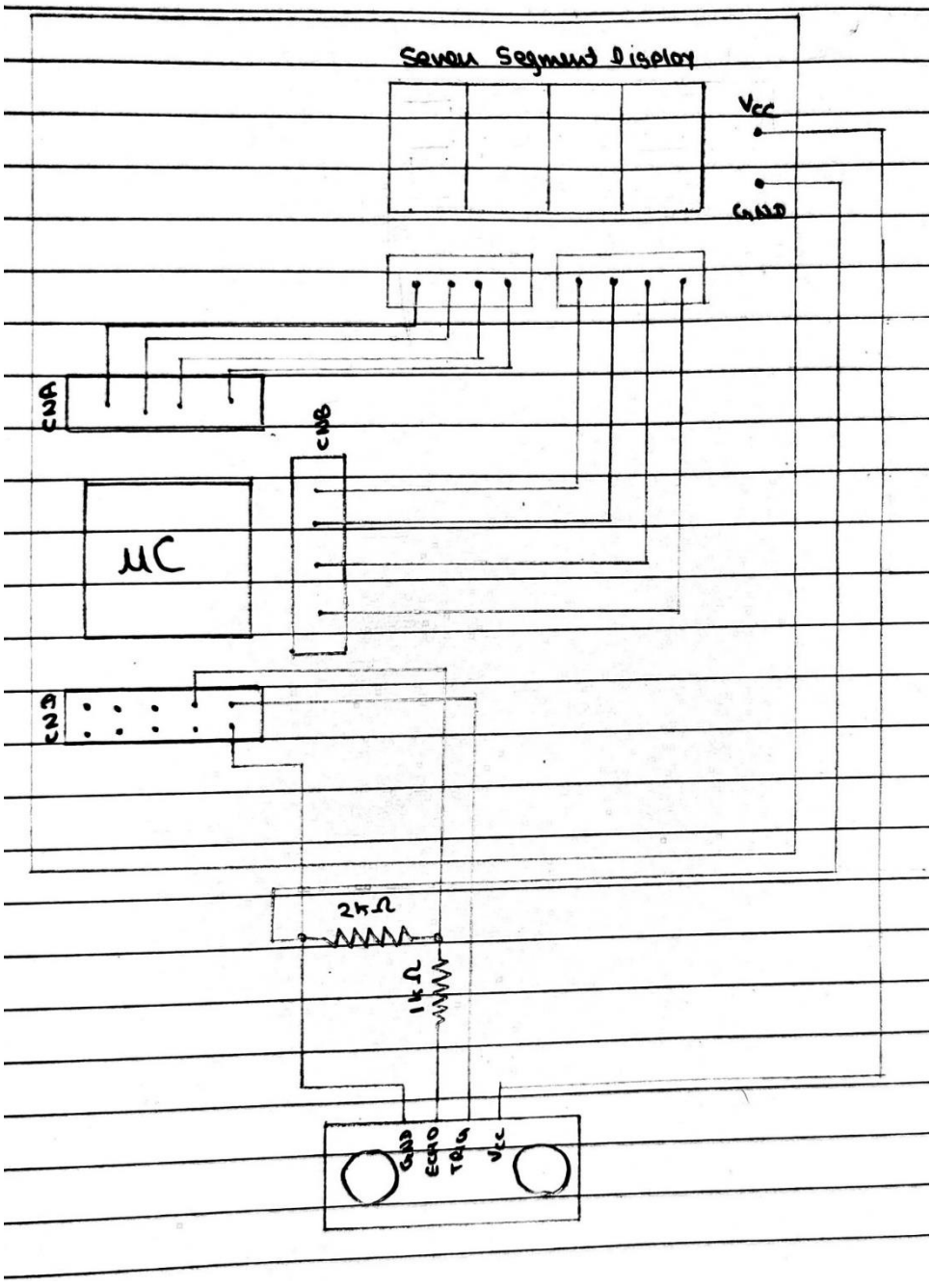
Calculations are as follows. Speed of sound in air,
**Vs = 343 m/s = 0.0343 cm/us**

Distance Travelled = Speed x Time taken
**D' = 343 m/s x T s**
**= 0.0343 cm/us x T us**

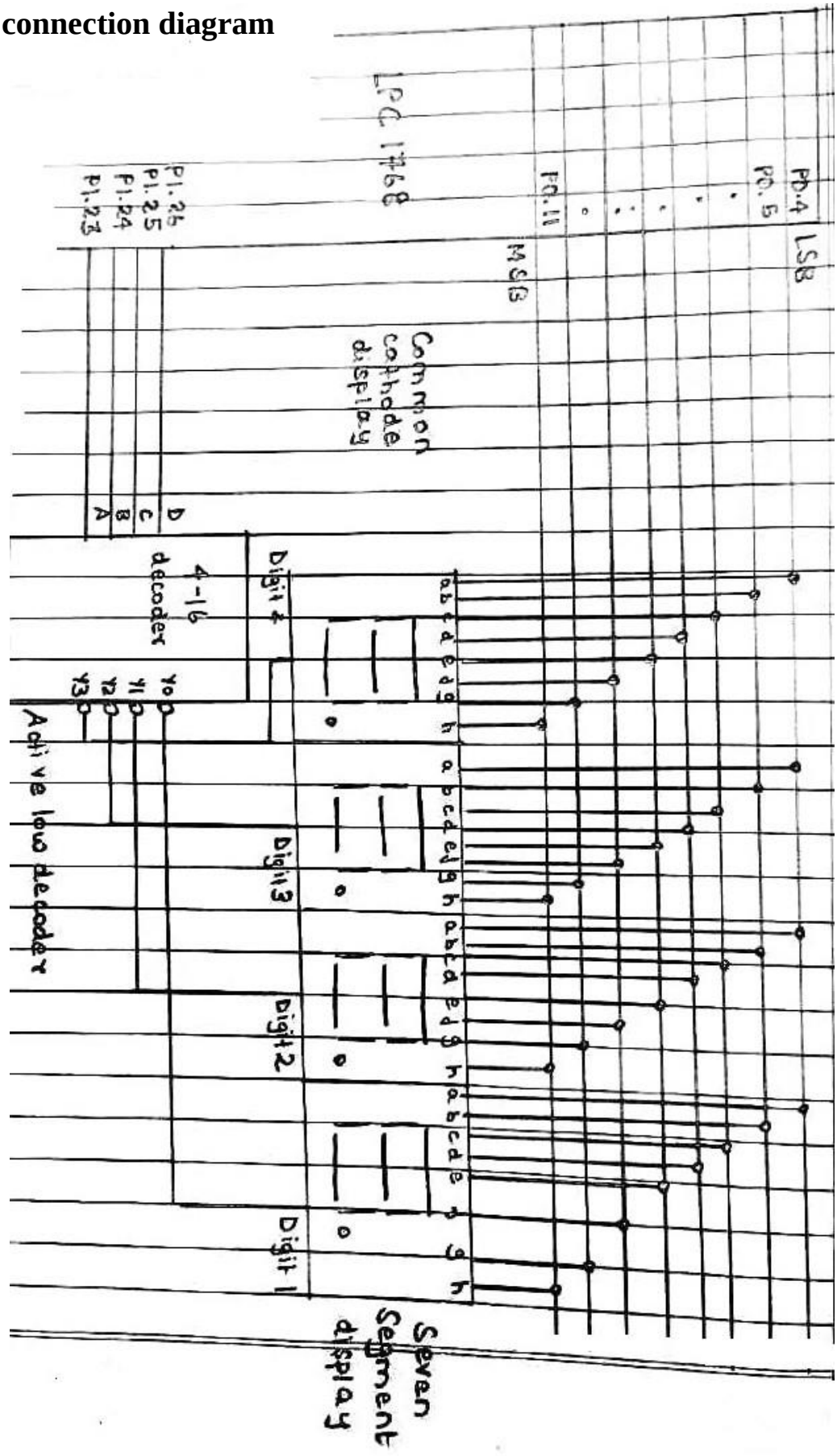
Since the ultrasonic waves travel as an echo back and forth, the total distance travelled is ascertained as
**D = D'/2**
**= (0.0343 x T)/2**
**= 0.01715 x T**




## Development Board Diagram



**Pin connection diagram**



## Code

**Main.c**

//C program to calculate and display the distance

```
//C program to calculate and display the distance
#include <lpc17xx.h>
#include <stdio.h>
#include "doppler_lib.h"

#define TRIG (1<<23) //P0.23
#define ECHO (1<<24) //P0.24

// Stores Hex values of each digit from 0 to F
unsigned char
seven_seg[16]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x77,0x7C,0x3
9,0X5E,0X79,0X71};
//To activate a particular seven segment display using decoder
unsigned int dig_sel[4] = {0<<23, 1<<23, 2<<23, 3<<23};
// will store the digit to be displayed
unsigned char digits[] = {0,0,0,0};
unsigned int i,x;
int echoTime = 0;
float distance = 0;
unsigned long r1;

void timer_init(void);
void display(void);

int main(void)
{
  SystemInit(); //Gets called by Startup code, sets CCLK=100Mhz, PCLK=25Mhz
  SystemCoreClockUpdate();

  initTimer0(); //Init Timer for delay functions

  LPC_GPIO0->FIODIR=0xFF<<4; // set the P0.4 - P0.11 as output
  LPC_GPIO1->FIODIR=0xF<<23; // set P1.23 - P1.26 as output

  LPC_GPIO0->FIODIR |= (TRIG);    //Set P0.23(TRIG) as output
  LPC_GPIO0->FIODIR &= ~(1<<24);  //Set P0.24(ECHO) as input (explicitly)
  LPC_GPIO0->FIOCLR |= (TRIG);    //Set P0.23 LOW initially
  distance=0;
  while(1)
  {
    echoTime = 0; //Resetting echo time
    distance = 0; //Resetting total distance
    x = 100000;   //Resetting delay variable
```

```
    LPC_GPIO0->FIOPIN |= TRIG;
    delayUS(10); //Output 10us HIGH on TRIG pin
    LPC_GPIO0->FIOCLR |= TRIG;

    while(!(LPC_GPIO0->FIOPIN & ECHO)); //Wait for a HIGH on ECHO pin

    startTimer0(); //Start counting

    while(LPC_GPIO0->FIOPIN & ECHO);    //Wait for a LOW on ECHO pin

    echoTime = stopTimer0(); //Stop counting and save value(us) in echoTime

    distance = (0.0343 * echoTime)/2; //Find the distance
    distance *= 100;  //Converting to Centimeters

    for(i = 3;  ;i--)
    {
      digits[i] = (int)distance%10;
      distance/=10;
      if(i==0)
        break;
    }
    display();
    while(x > 0)
      x--;
  }
}


void display(void)
{
  int x=0, y = 3,i;
  //Display 4 segments values one by one
  for(x=3,y=0;y<=3;x--, y++)
  {
    //enable the decoder lines according to the x value
    LPC_GPIO1->FIOPIN=dig_sel[x];
    //for other segments get the 7 segment values of the digits from
seven_seg[]
    r1=(seven_seg[digits[y]]);
    //Put the 7 segment value into data lines(P0.4 to P0.11)
    LPC_GPIO0->FIOPIN=r1<<4;
    //Wait for some time (small delay)
    for(i=0;i<100000;i++);
    //clear the data lines
    LPC_GPIO0->FIOPIN=00<<4;
  }
}
```

```
void timer_init()
{
  LPC_TIM0->CTCR=0X00; //timer mode
  LPC_TIM0->TCR=0X02; //reset TC and PC
  LPC_TIM0->MCR=0X02; //reset the TC and PC on match
  LPC_TIM0->PR=0X02; //TC will increment for every 3 PCLK
  //calculated using formula "MR=(PCLK*DELAY)/PR+1" where the delay is 3s
  LPC_TIM0->MR0=2999999;
  //initially EMC0 is HIGH when there is a match it is configured to become
LOW
  LPC_TIM0->EMR=0X011;
  LPC_TIM0->TCR=0X01; //start the timer
}
```

**doppler.c**

// code that handles the timer

```
#include <lpc17xx.h>
#include "doppler_lib.h"
#include <stdio.h>

void initTimer0(void) //PCLK must be = 25Mhz!
{
    LPC_TIM0->CTCR = 0x0; // To enable timer mode
    LPC_TIM0->PR = PRESCALE; //Increment TC at every 24999+1 clock cycles
    LPC_TIM0->TCR = 0x02; //Reset Timer
}

void startTimer0(void)
{
    LPC_TIM0->TCR = 0x02; //Reset Timer
    LPC_TIM0->TCR = 0x01; //Enable timer
}

unsigned int stopTimer0(void)
{
    LPC_TIM0->TCR = 0x00; //Disable timer
    return LPC_TIM0->TC;
}

void delayUS(unsigned int microseconds) //Using Timer0
{
    LPC_TIM0->TCR = 0x02; //Reset Timer
    LPC_TIM0->TCR = 0x01; //Enable timer
    while(LPC_TIM0->TC < microseconds); //wait until timer counter reaches the
desired delay
    LPC_TIM0->TCR = 0x00; //Disable timer
```

```
}

void delayMS(unsigned int milliseconds) //Using Timer0
{
    delayUS(milliseconds * 1000);
}
```

**doppler_lib.h**

```
#ifndef TEAM2_LPC176X
#define TEAM2_LPC176X

#define PRESCALE (25000-1)

void initTimer0(void);
void startTimer0(void);
unsigned int stopTimer0(void);
void delayUS(unsigned int microseconds);
void delayMS(unsigned int milliseconds);

#endif
```

## Result

Distance is varying but slightly inconsistent with the input distances given.