



**KLE** Technological  
University

Creating Value  
Leveraging Knowledge

## **Review - 1**

### **Team – A5**

USN: 02FE21BEC064 – PRAVEEN MAGADUM  
USN: 02FE21BEC042 – KOMAL MELAVANKI  
USN: 02FE22BEC021 – BHUVAN BUDAVI  
USN: 02FE22BEC019 – BHAKTI BETAGERI

### **Topic:**

**Write a C program to interface LCD and ultrasonic sensor**

Under The Guidance Of:  
Prof. S. M. Hunagund

\_\_\_\_\_  
Guide Signature

- **Code:**

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

#define TRIG_PIN (1 << 10) // Define the pin for the trigger
#define ECHO_PIN (1 << 11) // Define the pin for the echo

unsigned long int temp1=0, temp2=0 ;
unsigned char Msg1[14] = {"Team 5"};
unsigned char Msg2[15] = {"LCD"};
unsigned char buffer[50];

void delay_us(unsigned int us);
float measureDistance(void);

int main(void)
{
    float distance;
    SystemInit();
    lcd_init(); //initialise LCD
    //delay_lcd(3200);

    //temp1 = 0x81; //1st message on LCD 1st line
    //lcd_com();
    //delay_lcd(800);
    //lcd_puts(Msg1);

    //temp1 = 0xC1; //Msg2 on LCD 2nd line
    //lcd_com();
    //delay_lcd(800);
    //lcd_puts(Msg2);
    //delay_lcd(5000);

    while (1)
    {
        distance = measureDistance(); // Measure distance using the ultrasonic sensor
        sprintf(buffer, "Distance: %.2f cm\r\n", distance);
        temp1 = 0x81; //1st message on LCD 1st line
        lcd_com();
        delay_lcd(800);
        lcd_puts(buffer);
        delay_us(1000000); // Delay for a second before the next measurement
    }
}

// Function to measure distance using the ultrasonic sensor
float measureDistance()
{
    uint32_t i;
    float distance;

    // Set TRIG_PIN as output and ECHO_PIN as input
    LPC_GPIO0->FIODIR |= TRIG_PIN;
```

```

LPC_GPIO0->FIODIR &= ~ECHO_PIN;

// Ensure TRIG_PIN is low initially
LPC_GPIO0->FIOCLR |= TRIG_PIN;
delay_us(2);

// Generate 10us pulse on TRIG_PIN to trigger the sensor
LPC_GPIO0->FIOSET |= TRIG_PIN;
delay_us(10);
LPC_GPIO0->FIOCLR |= TRIG_PIN;

// Wait for ECHO_PIN to go high
while (!(LPC_GPIO0->FIOPIN & ECHO_PIN));

// Measure the width of the pulse
i = 0;
while (LPC_GPIO0->FIOPIN & ECHO_PIN)
{
    i++;
    delay_us(1);
}

// Calculate distance using the pulse width
distance = (i * 0.017); // Speed of sound is 340 m/s

return distance;
}

void delay_us(unsigned int us)
{
    us = us * 22; // This is roughly equivalent to 1 us delay
    while (us--);
}

//lcd initialization
void lcd_init()
{
    unsigned char arr1[4]={0x30,0x30,0x30,0x20};
    unsigned char arr2[4]={0x28,0x0C,0x06,0x80};
    unsigned int i;
    /* Ports initialized as GPIO */
    LPC_PINCON->PINSEL3 &= 0xFFFF00FF; //P1.20 to P1.23
    LPC_PINCON->PINSEL7 &= 0XFFF3FFFF; //P3.25
    LPC_PINCON->PINSEL7 &= 0xFFCFFFFFFF; //P3.26
    LPC_PINCON->PINSEL9 &= 0xFCFFFFFFF; //P4.28

    /* Setting the directions as output */
    LPC_GPIO1->FIODIR |= DT_CTRL; // data lines - P1.20 to P1.23
    LPC_GPIO3->FIODIR |= RS_CTRL; // RS - P3.25
    LPC_GPIO3->FIODIR |= RW_CTRL; // RW - P3.26
    LPC_GPIO4->FIODIR |= EN_CTRL; // P4.28

    clear_ports();
    delay_lcd(3200);
}

```

```

        for (i=0;i<4;i++)
        {
            temp2=arr1[i];
            wr_cn();
            delay_lcd(30000);
        }
        for (i=0;i<4;i++)
        {
            temp1=arr2[i];
            lcd_com();
            delay_lcd(800);
        }
        return;
    }

void lcd_com(void)
{
    temp2= temp1 & 0xf0;
    temp2 = temp2 << 16;
    wr_cn();
    temp2 = temp1 & 0x0f;
    temp2 = temp2 << 20;
    //temp2 = temp2 << 4;
    wr_cn();
    delay_lcd(1000);
    return;
}

// command nibble o/p routine
void wr_cn(void)                //write command reg
{
    clear_ports();
    LPC_GPIO1->FIOPIN = temp2;
    LPC_GPIO3->FIOCLR = RW_CTRL;
    LPC_GPIO3->FIOCLR = RS_CTRL;
    LPC_GPIO4->FIOSET = EN_CTRL;
    delay_lcd(25);
    LPC_GPIO4->FIOCLR = EN_CTRL;
    return;
}

void lcd_data(void)
{
    temp2 = temp1 & 0xf0;
    temp2 = temp2<< 16;
    wr_dn();
    temp2= temp1 & 0x0f;
    temp2= temp2 << 20;
    //temp2= temp2 << 4;
    wr_dn();
    delay_lcd(1000);
    return;
}

// data nibble o/p routine
void wr_dn(void)

```

```

{
    clear_ports();
    LPC_GPIO1->FIOPIN = temp2;                // Assign the value to the data lines
    LPC_GPIO3->FIOSET = RS_CTRL;              // set bit RS
    LPC_GPIO3->FIOCLR = RW_CTRL;              // clear bit RW
    LPC_GPIO4->FIOSET = EN_CTRL;              // EN=1
    delay_lcd(25);
    LPC_GPIO4->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_GPIO1->FIOCLR = DT_CTRL; //Clearing data lines
    LPC_GPIO3->FIOCLR = RS_CTRL; //Clearing RS line
    LPC_GPIO3->FIOCLR = RW_CTRL; //Clearing RW line
    LPC_GPIO4->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 = 0xC1;
            lcd_com();
        }
    }
    Return;
}

```

**1**

## OUTPUT

