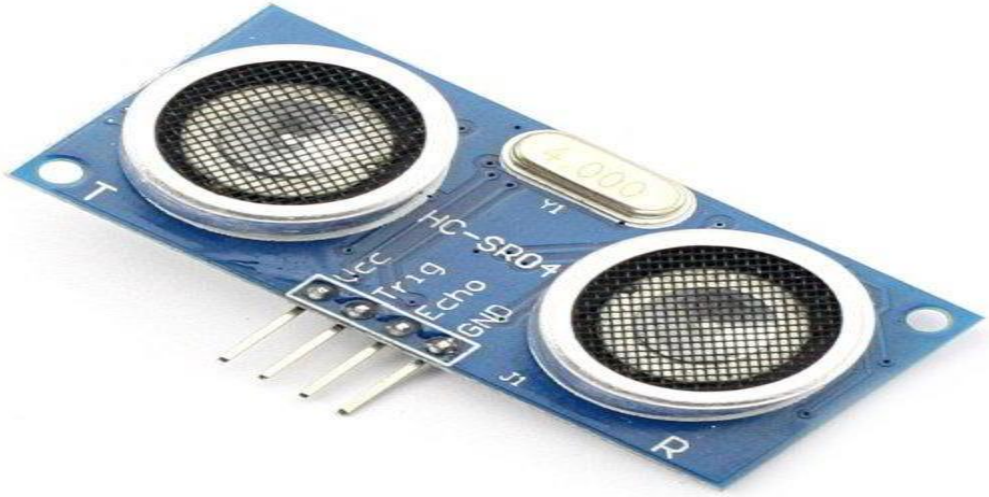
Ultrasonic Module HC-SR04 Interfacing with 8051

Nomula Rishith [21BEE1096], Bhupesh Kumar [21BEE1060], Sreeniketh Dasam [21BEE1337], Reddy Charan Kumar [21BEE1328], Sai Pranav Bhagavatula [21BEE1043]

An ultrasonic sensor range detector using 8051 is a widely used technology for distance measurement and object detection. It operates based on the principles of ultrasonic waves, emitting high-frequency sound waves and then calculating the time taken for the waves to return after hitting an object. This enables accurate and reliable distance measurements, making it applicable in various fields such as robotics, automotive, and industrial automation.

1. INTRODUCTION



Ultrasonic HC-SR04 Module

Ultrasonic Module HC-SR04 works on the principle of SONAR and RADAR systems.

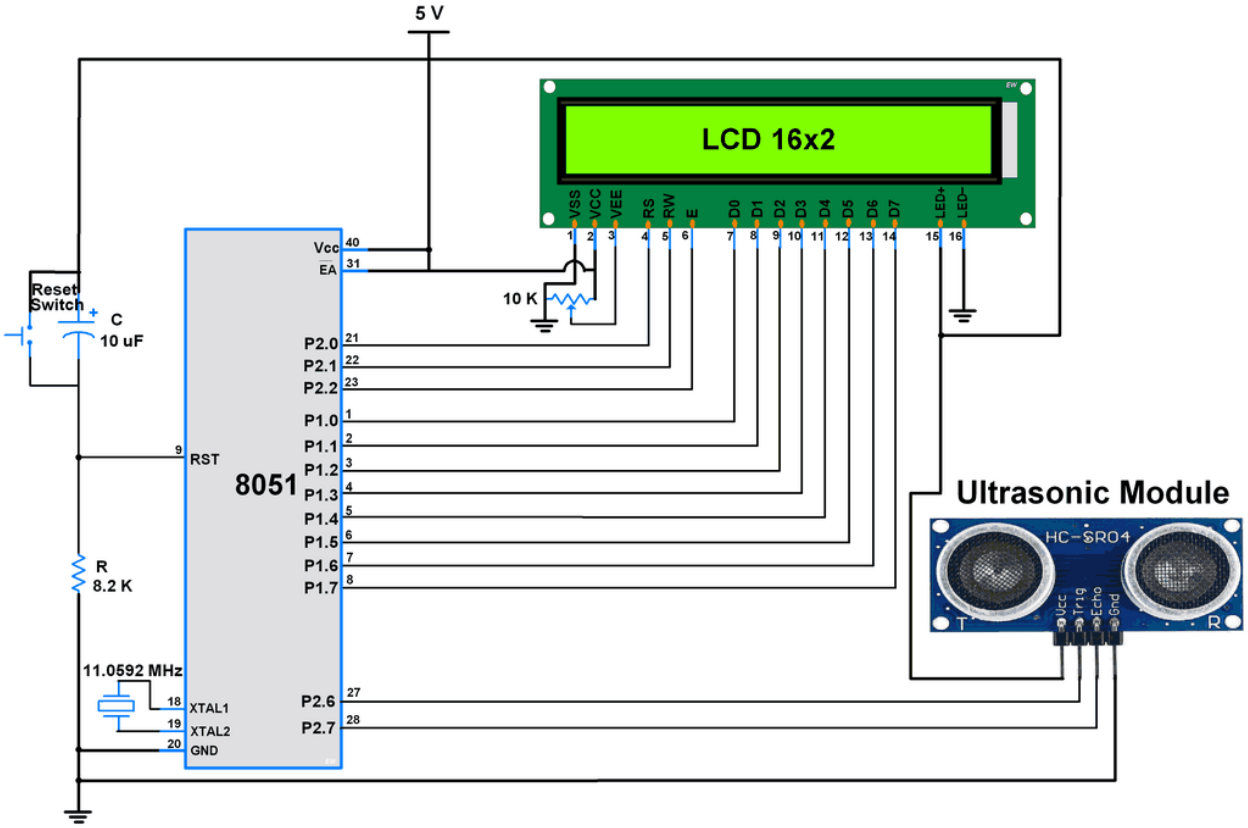
- The HC-SR04 module has an ultrasonic transmitter, receiver, and control circuit on a single board.
- The module has only 4 pins, Vcc, Gnd, Trig, and Echo.
- When a pulse of 10µsec or more is given to the Trig pin, 8 pulses of 40 kHz are generated. After this, the Echo pin is made high by the control circuit in the module.
- The echo pin remains high till it gets the echo signal of the transmitted pulses back.

- The time for which the echo pin remains high, i.e. the width of the Echo pin gives the time taken for generated ultrasonic sound to travel towards the object and return.
- Using this time and the speed of sound in air, we can find the distance of the object using a simple formula for distance using speed and time.

For more information about the ultrasonic module HC-SR04 and how to use it, refer to the topic Ultrasonic Module HC-SR04 in the sensors and modules section.

To measure time, we can use an in-built timer of 8051. To know about 8051 timers and their working, refer to topic 8051 timers in 8051 inside section.

2. INTERFACING DIAGRAM



HC-SR04 Ultrasonic Module Interfacing with 8051

3. OBJECTIVE

In this research endeavor, we propose the development of an application aimed at determining the distance to an object through the integration of an ultrasonic module, specifically the HC-SR04, with the AT89S52 microcontroller. The measured distance will be accurately displayed on a 16x2 LCD screen as part of this system.

4. STEPS OF PROGRAMMING

1. 8051 microcontroller needs to transmit at least 10 us trigger pulses to the HC-SR04 Trig Pin.
2. After getting a trigger pulse, HC-SR04 automatically sends eight 40 kHz sound waves and waits for rising edge output at the Echo pin.
3. When the rising edge capture occurs at the Echo pin which is connected to the input of 8051, start Timer of 8051 and again wait for the falling edge on the Echo pin.
4. As soon as the falling edge is captured at the Echo pin, the microcontroller reads the count of the Timer. This time count is used to calculate the distance to an object.

5. CALCULATION (distance in cm)

$$Distance = \frac{SoundVelocity * Time}{2}$$

Where,

Sound Velocity = 34300 (in cm per second)

Here, the oscillator frequency of AT89S52 (8051) is 11.0592 MHz then the timer frequency of 8051 will be 921.6 kHz. So, the time required to execute 1 instruction is 1.085 us.

So, the timer gets incremented after 1.085 us time elapse.

e.g.

$$= \frac{34300 * TimerCount * 1.085 * 10^{-6}}{2}$$

6. CODE

MAIN CODE

```
/*
        Find distance of an Object by interfacing Ultrasonic HC-SR04 module with 8051(AT89S52)

*/
#include<reg52.h>
#include <stdio.h>
#include <LCD_8_bit.h>
#include <math.h>

#define sound_velocity 34300        /* sound velocity in cm per second */

#define period_in_us pow(10,-6)
#define Clock_period 1.085*period_in_us          /* period for clock cycle of 8051*/

sbit Trigger_pin=P2^6;          /* Trigger pin */
sbit Echo_pin=P2^7;                                                    /* Echo pin */

void Delay_us()
{
                TL0=0xF5;
                TH0=0xFF;
                TR0=1;
                while (TF0==0);
                TR0=0;
                TF0=0;
}

void init_timer(){
        TMOD=0x01;
        /initialize Timer/
        TF0=0;
        TR0 = 0;
}

void send_trigger_pulse(){
          Trigger_pin= 1;          /* pull trigger pin HIGH */
   Delay_us();              /* provide 10uS Delay*/
   Trigger_pin = 0;         /* pull trigger pin LOW*/
}

void main()
{
        float distance_measurement, value;
        unsigned char distance_in_cm[10];
        LCD_Init();
        /* Initialize 16x2 LCD */
        LCD_String_xy(1,1,"Distance");
        init_timer();                                                          /*
Initialize Timer*/

while(1)
{
                send_trigger_pulse();                          /* send trigger pulse of 10us */
```

```
                    while(!Echo_pin);         /* Waiting for Echo */
   TR0 = 1;              /* Timer Starts */
   while(Echo_pin && !TF0);   /* Waiting for Echo goes LOW */
   TR0 = 0;              /* Stop the timer */

                    /* calculate distance using timer */
                    value = Clock_period * sound_velocity;
                    distance_measurement = (TL0|(TH0<<8));
                                    /* read timer register for time count */
                    distance_measurement = (distance_measurement*value)/2.0;  /* find distance(in cm) */

                    sprintf(distance_in_cm, "%.2f", distance_measurement);
                    LCD_String_xy(2,1,distance_in_cm);
                                            /* show distance on 16x2 LCD */
                    LCD_String("  cm  ");

                    delay(100);
}
}
```

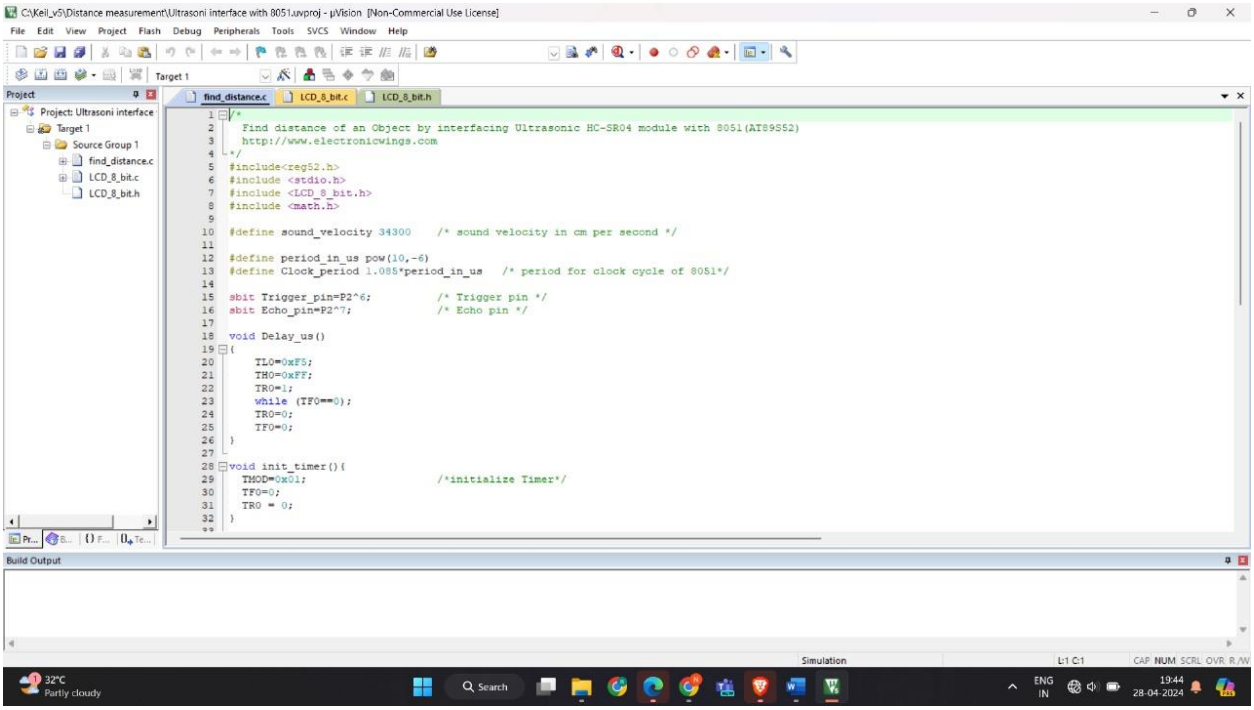
LCD INTERFACING CODE

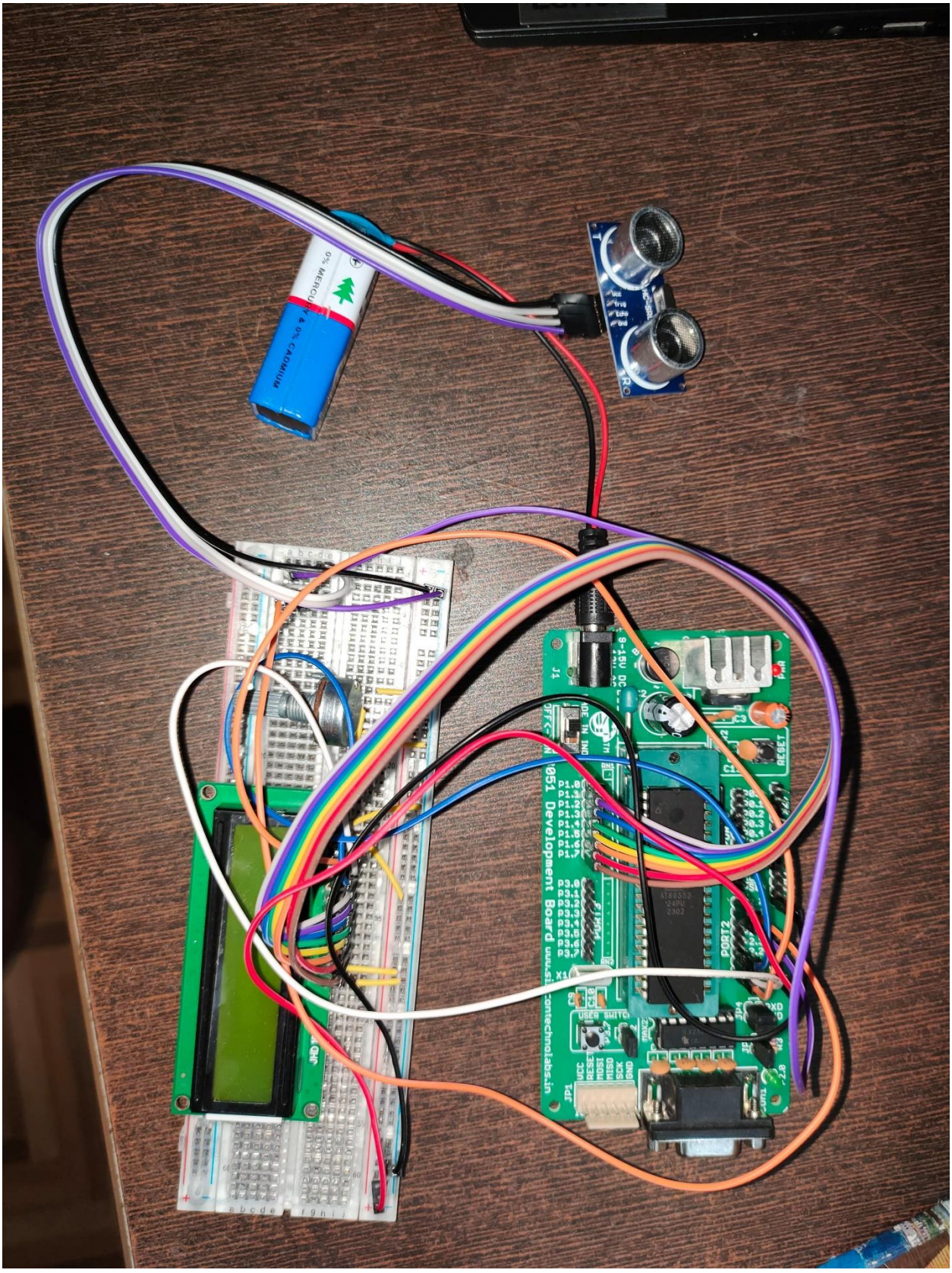
```
/*
 * LCD_8_bit.c
 *
 * http://www.electronicwings.com
 */

#include <reg52.h>
#include <intrins.h>
#include "LCD_8_BIT.h"

#define LCD_Data P1

sbit RS=P2^0;
sbit RW=P2^1;
sbit EN=P2^2;

void delay(unsigned int k)
{
        int i,j;
        for (i=0;i<k;i++)
                for (j=0;j<112;j++);
}

void LCD_Command(char Command)
{
        LCD_Data = Command;
        RS=0;
        RW=0;
        EN=1;
        delay(1);
        EN=0;
        delay(3);
```

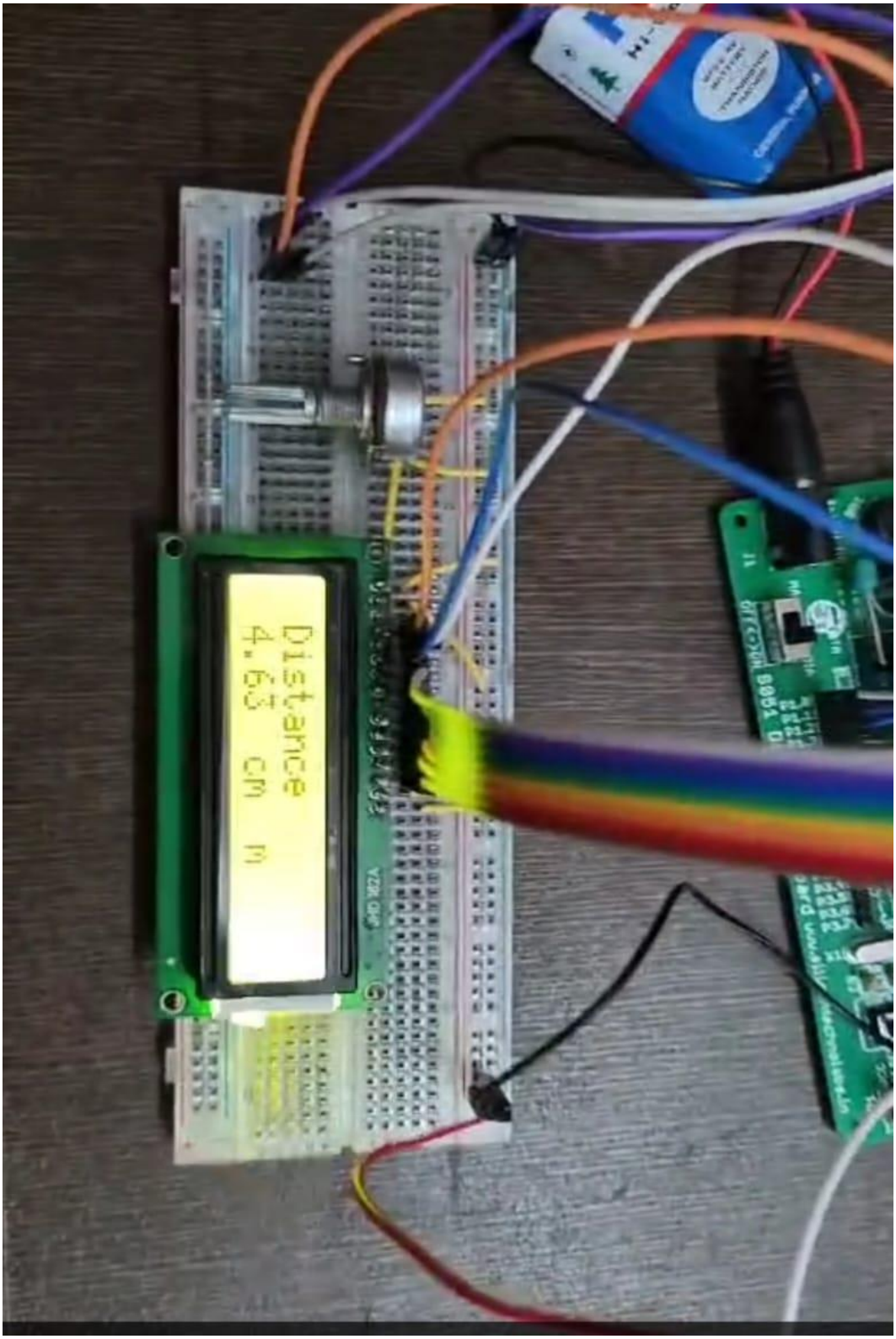
```
}
void LCD_Char(char Data)
{
        LCD_Data = Data;
        RS=1;
        RW=0;
        EN=1;
        delay(1);
        EN=0;
        delay(3);
}

void LCD_String(unsigned char *str)
{
        int i;
        for(i=0;str[i]!=0;i++)                                  /* Send each char
of string till the NULL */
        {
                LCD_Char (str[i]);                                      /* Call
LCD data write */
        }
}

void LCD_String_xy (unsigned char row, unsigned char pos, unsigned char str)    / Send string to LCD
function */
{
        if (row == 1)
                LCD_Command((pos & 0x0F)|0x80);                 /* Command of first row
and required position<16 */
        else if (row == 2)
                LCD_Command((pos & 0x0F)|0xC0);                         /* Command of
Second row and required position<16 */
        LCD_String(str);                                                /* Call LCD string
function */
}

void LCD_Init()
{
        delay(50);
        LCD_Command(0x38);
        LCD_Command(0x0C);
        LCD_Command(0x01);
        LCD_Command(0x06);
        LCD_Command(0x80);
}
```

```
/*
 * LCD_8_bit.h
 */

#ifndef _LCD_8_BIT_H_
#define _LCD_8_BIT_H_

#include <reg52.h>

void delay(unsigned int);
void LCD_Command(char Command);
void LCD_Char(char Data);
void LCD_String(unsigned char *ptr);
void LCD_String_xy(unsigned char row,unsigned char column, unsigned char *);
void LCD_Init();

#endif
```

7. OUTPUT IMAGES




CADMIUM
9-15V DC
RESET
8051 Development Board
USER SWITCH
GND


Distance
4.63 cm