1. Project Title

Distance Measurement and Display using 8051, Ultrasonic Sensor, and LCD (with Arduino UNO as Programmer)

---

2. Objective

To build a system using the 8051 (AT89S52) microcontroller that:

- Measures distance using the HC-SR04 ultrasonic sensor.
- Displays the measured distance on an I2C LCD screen.
- Uses Arduino UNO as an ISP programmer to flash code into the 8051.

---

3. Components Required

- AT89S52 (8051) Microcontroller
- HC-SR04 Ultrasonic Sensor
- 16x2 I2C LCD with Potentiometer
- Arduino UNO
- Jumper Wires
- Breadboard

---

4. Connections Overview

A. Ultrasonic Sensor (HC-SR04) to 8051

Sensor Pin	8051 Pin	Description
VCC	5V	Power supply
TRIG	P3.2	Trigger signal
ECHO	P3.3	Echo signal
GND	GND	Ground

B. LCD with Potentiometer to 8051

LCD Pin	8051 Pin	Description
GND	GND	Common ground
VCC	5V	Power supply
SDA	P1.0	I2C Data Line
SCL	P1.1	I2C Clock Line
LED+	5V	LCD Backlight Power

C. Arduino UNO to 8051 (for Programming using ISP)

Arduino Pin	8051 Pin	Function
Pin 10	RESET	Reset signal
Pin 11	MOSI	Master Out Slave In
Pin 12	MISO	Master In Slave Out
Pin 13	SCK	Serial Clock
GND	GND	Ground
VCC	5V	Power supply

5. Software Used

- **Keil uVision** for writing and compiling C code for 8051.
- **AVRDUDE** and **Arduino UNO** for flashing the hex file to 8051.

6. Working Principle

1. **Distance Measurement:**
    - The HC-SR04 ultrasonic sensor sends out ultrasonic pulses via the **TRIG** pin.
    - The reflected signal is captured by the **ECHO** pin.
    - Time taken is calculated and converted to distance using: Distance (cm) = (Time × Speed of Sound) / 2

2. **Display:**
    - The measured distance is sent via I2C to the 16x2 LCD.
    - The LCD displays the real-time distance in cm.
3. **Programming the 8051:**
    - The compiled hex file is transferred using **Arduino UNO as ISP** and **AVRDUDE**.

---

7. Codes

Embedded C code

```
#include<reg51.h>

#define LCD_ADDR 0x4E // I2C LCD address for PCF8574 (0x27 << 1)

sbit SDA = P1^0; sbit
SCL = P1^1; sbit trig =
P3^2; sbit echo =
P3^3;

// ---------- Delay Functions ---------- void
delay_us(unsigned int t) {   while(t--);
}

void delay_ms(unsigned int t) {
unsigned int i, j;   for(i = 0; i < t;
i++)     for(j = 0; j < 1275; j++);
}

// ---------- I2C Functions ---------- void
i2c_start() {
  SDA = 1; SCL = 1; delay_us(5);
  SDA = 0; delay_us(5);
  SCL = 0;
}

void i2c_stop() {
  SDA = 0; SCL = 1; delay_us(5);
  SDA = 1; delay_us(5);
}

bit i2c_write(unsigned char dat) { unsigned char i;   bit ack;   for(i = 0; i < 8; i++) {
    SDA = (dat & 0x80);
    SCL = 1; delay_us(5);
SCL = 0; delay_us(5);     dat
<<= 1;
  }
  SDA = 1;   SCL = 1;
delay_us(5);   ack =
SDA;   SCL = 0;   return
ack;
}

// ---------- LCD I2C Functions ---------- void
lcd_i2c_cmd(unsigned char cmd) {   unsigned
char upper = cmd & 0xF0;   unsigned char
lower = (cmd << 4) & 0xF0;

  i2c_start();
  i2c_write(LCD_ADDR);
i2c_write(upper | 0x0C);
i2c_write(upper | 0x08);
i2c_write(lower | 0x0C);
i2c_write(lower | 0x08);
i2c_stop();   delay_us(50);

}


void lcd_i2c_data(unsigned char dat) {
unsigned char upper = dat & 0xF0;   unsigned
char lower = (dat << 4) & 0xF0;


  i2c_start();
  i2c_write(LCD_ADDR);
i2c_write(upper | 0x0D);
i2c_write(upper | 0x09);
i2c_write(lower | 0x0D);
i2c_write(lower | 0x09);
i2c_stop();   delay_us(50);
}


void lcd_i2c_init() {   delay_ms(50);
lcd_i2c_cmd(0x33);
lcd_i2c_cmd(0x32);
lcd_i2c_cmd(0x28);
lcd_i2c_cmd(0x0C);
```

```
lcd_i2c_cmd(0x06);
lcd_i2c_cmd(0x01);   delay_ms(5);
}

void lcd_i2c_print(char *str) {
  while(*str)
    lcd_i2c_data(*str++);
}

// ---------- Ultrasonic Sensor Functions ----------
void send_trigger() {   trig = 0;   delay_us(2);   trig
= 1;
  delay_us(15);
  trig = 0;
}

unsigned int get_distance() {   unsigned
int time;

  send_trigger();

  // Wait for echo to go HIGH   while(!echo);

  // Start Timer1 (16-bit mode)
  TMOD &= 0x0F;      // Clear Timer1 bits
  TMOD |= 0x10;      // Timer1 in Mode 1
  TH1 = 0;
  TL1 = 0;
  TR1 = 1;          // Start Timer1

  while(echo);      // Wait until echo goes LOW

  TR1 = 0;          // Stop Timer1

  time = (TH1 << 8) | TL1;

  // Convert to distance in cm (based on 11.0592 MHz clock and 1 machine cycle = 1.085 us)
return (time * 1.085) / 58;
}

// ---------- Main ----------
void main() {   unsigned
int dist;

  lcd_i2c_init();   lcd_i2c_cmd(0x80);
lcd_i2c_print("Distance:");

  while(1) {     dist =
get_distance();

    lcd_i2c_cmd(0xC0);
lcd_i2c_data((dist/100) + '0');
lcd_i2c_data(((dist/10)%10) + '0');
lcd_i2c_data((dist%10) + '0');
lcd_i2c_print(" cm    ");     delay_ms(200);

  }
```

```
}
```

Arduino Code

File>Examples>ArduinoISP
Then Flash into Arduino UNO board after selecting Board type and COM Port.

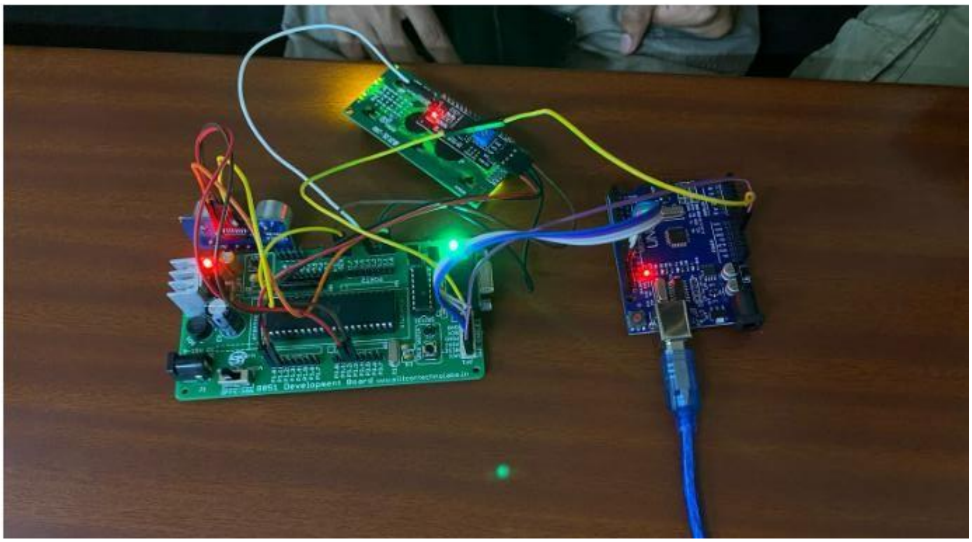
---

8. Steps to Execute the Project

1. **Hardware Setup:**
   - Connect the **Ultrasonic Sensor**, **LCD with Potentiometer**, and **Arduino UNO** to the **8051 (AT89S52)** microcontroller as per the circuit diagram.
2. **Write Embedded C Code:**
   - Use **Keil uVision** to write the embedded C code for interfacing the ultrasonic sensor and LCD. o The code should include I2C LCD display logic and distance calculation from the ultrasonic sensor.
3. **Generate HEX File:**
   - Compile the code in Keil to generate the corresponding .hex file for the 8051 microcontroller.
4. **Flash Arduino UNO with ArduinoISP:**
   - Open **Arduino IDE**. o Go to **File > Examples > 11.ArduinoISP > ArduinoISP**.
   - Upload this sketch to the Arduino UNO to turn it into an ISP programmer.
5. **Burn HEX to 8051 Using AVRDUDE:**
   - Connect the Arduino UNO to the 8051 circuit as per the SPI/ISP pin configuration.
   - Use **AVRDUDE** command-line tool to flash the hex file onto the AT89S52 via Arduino.
     (Make sure to configure the right COM port and parameters in the command)
6. **Verify Output on LCD:**
   - Once flashing is complete, power the circuit.
   - The LCD should display the measured **distance from the ultrasonic sensor** in centimeters.

---

9. Connections



10. Output