Experiment No: 1

Experiment Name: Smoke Detection with Arduino

Objectives:

- Develop a system using an Arduino Uno and a smoke sensor that detects the presence of smoke and triggers an alarm.
- Understand the operating principles and characteristics of the chosen sensor.

Materials:

- Arduino Uno or compatible board
- MQ-2 gas sensor
- Breadboard
- Jumper wires
- Buzzer (optional)

Circuit Diagram:

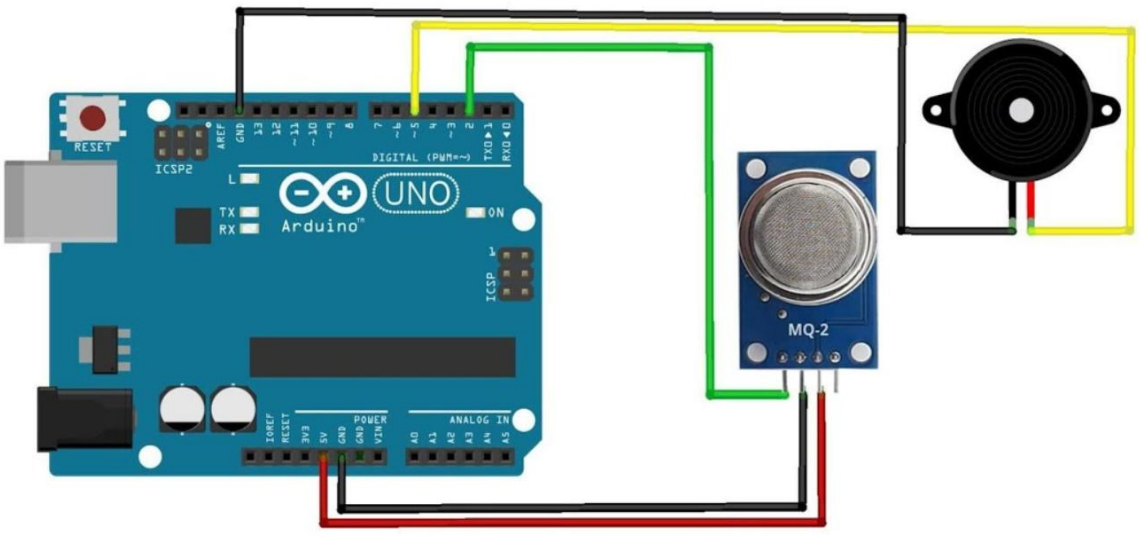

Fig: Smoke Detector Arduino Circuit

Procedure:

1. **Circuit Assembly:**

   - Connect the positive terminal of the MQ-2 to Vin on the Arduino.
   - Connect the negative terminal of the MQ-2 to GND on the Arduino.
   - Connect the analog output of the MQ-2 (AO) to an analog pin on the Arduino (e.g., A0).
   - Connect the buzzer to another digital pin

2. **Testing the Detector:**

   - Power up your Arduino.
   - Cover the MQ-2 sensor with your finger or blow smoke towards it.
   - Observe the buzzer. It should tone when smoke is detected.

3. **Analysis and Discussion:**

   - Discuss the principle of operation of the MQ-2 sensor.
   - Analyze the sensitivity and limitations of the system.
   - Explore potential improvements and applications of your smoke detector.

Code:

```
int buzzer = 10;
int smokeA0 = A5;

// Your threshold value. You might need to change it.
int sensorThres = 400;

void setup() {
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
}

void loop() {
  int analogSensor = analogRead(smokeA0);
```

```
  Serial.print("Pin A0: ");
  Serial.println(analogSensor);
  // Checks if it has reached the threshold value
  if (analogSensor > sensorThres)
  {
    tone(buzzer, 1000, 200);
  }
  else
  {
    noTone(buzzer);
  }
  delay(100);
}
```

Experiment No: 2

Experiment Name: Rain Detection Alarm System with Arduino

Objectives:

- Develop a system using an Arduino Uno and a rain sensor that detects rainfall and triggers an alarm.

Materials:

- Arduino Uno
- Rain sensor module (e.g., YL-83)
- Breadboard
- Jumper wires
- Buzzer

Circuit Diagram:

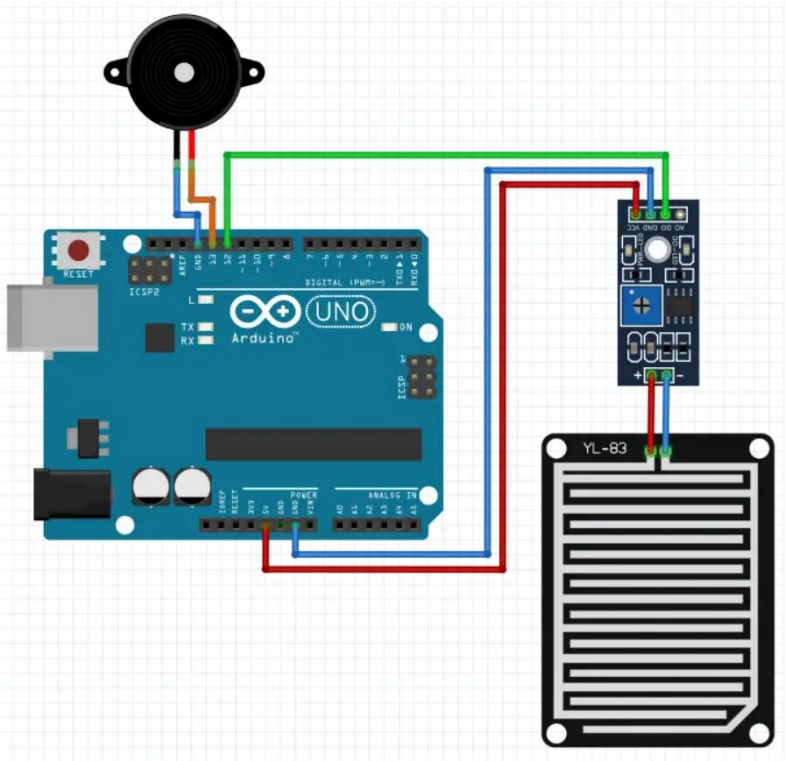

Fig: Rain Detector Arduino Circuit

Procedure:

1. **Circuit Assembly:**

    - Connect the VCC pin of the rain sensor to the 5V pin of the Arduino.
    - Connect the GND pin of the rain sensor to the GND pin of the Arduino.
    - Connect the DO/A0 (digital output) pin of the rain sensor to any digital pin of the Arduino (e.g., pin 12).
    - Connect the positive leg of the buzzer to any digital pin of the Arduino (e.g., pin 5).
    - Connect the negative leg of the buzzer to GND through a resistor (e.g., 220Ω).
    - Connect the battery or power supply to the Arduino's Vin and GND pins.

2. **Explanation:**

The code works the same way as with the Arduino Uno:

- The rain sensor outputs a LOW signal when it detects water (rain).
- The code continuously reads the sensor's digital output pin.
- If the sensor detects rain and it wasn't raining before, the is Raining flag is set to true and the buzzer is turned on.
- If the sensor detects no rain and it was raining before, the isRaining flag is set to false and the buzzer is turned off.

Code:

```
#define rain A0
#define buzzer 5
int value;
int set=10;
void setup() {
Serial.begin(9600);
pinMode(buzzer,OUTPUT);
pinMode(rain,INPUT);
}
void loop() {
```

```
value = analogRead(rain);
Serial.println("Rain Detector is ON");
Serial.println(value);
value = map(value,0,1023,225,0);
Serial.println(value);
if(value>=set){
Serial.println("Rain has came");
digitalWrite(buzzer,HIGH);
}
else{
digitalWrite(buzzer,LOW);
}
delay(200);
}
```

Experiment No: 3

Experiment Name: Sound Sensor Detection using Arduino Uno

Objective:

- To develop a system using an Arduino Uno and a sound sensor that can detect and respond to sound levels in the environment.

Materials:

- Arduino Uno or compatible board
- Sound sensor (e.g., KY-038 or microphone)
- Breadboard
- Jumper wires
- LEDs

Circuit Diagram:

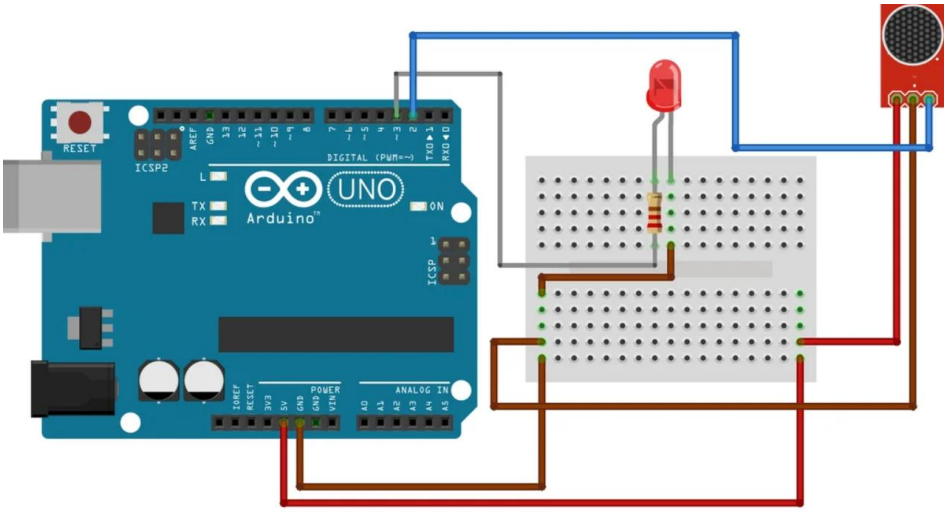

Fig: Sound Detector Arduino Circuit

Procedure:

1. **Circuit Assembly:**
    - A0 (on sound sensor) to Analog pin 2 on Arduino board.
    - Vcc or + (on sound sensor) to 5v on Arduino board.
    - GND (on sound sensor) to GND on Arduino board.
    - LED short leg to GND on Arduino Board on opposite side of wire connections. Long leg to digital pin 13 on the Arduino Board.
    - Connect the Arduino to the power supply.
2. **Testing and Analysis:**
    - Power up your Arduino.
    - Observe the LED(s).
    - Clap your hands, whistle, or make other sounds near the sensor.
    - Analyze how the sensor value changes with different sound intensities.
    - Adjust the thresholds in the code to experiment with sensitivity.
3. **Discussion and Extensions:**
    - Discuss the principle of operation of your chosen sound sensor.
    - Explain how the code converts the sensor reading into sound detection.
    - Explore potential improvements and applications of your sound sensor project.

Code:

```
const int ledpin = 13;
const int soundpin = A0;
const int threshold = 550;

void setup() {
  Serial.begin(9600);
  pinMode(ledpin, OUTPUT);
  pinMode(soundpin, INPUT);
}

void loop() {
  int soundsens = analogRead(soundpin);

  Serial.println(soundsens); // Print the sensor value to serial monitor
```

```
  if (soundsens >= threshold) {
    digitalWrite(ledpin, HIGH);
    delay(2000);
  } else {
    digitalWrite(ledpin, LOW);
  }
}
```

Experiment No: 4

Experiment Name: Distance Measuring Sensor using Arduino Uno

Objectives:

- To develop a system using an Arduino Uno and a distance measuring sensor that can accurately and reliably measure the distance to objects within a specified range.
- Understand the operating principles and characteristics of the chosen sensor.

Materials:

- Arduino Uno board
- HC-SR04 Ultrasonic Sensor
- Breadboard
- Jumper wires
- LCD Display
- Computer with Arduino IDE software installed

Circuit Diagram:

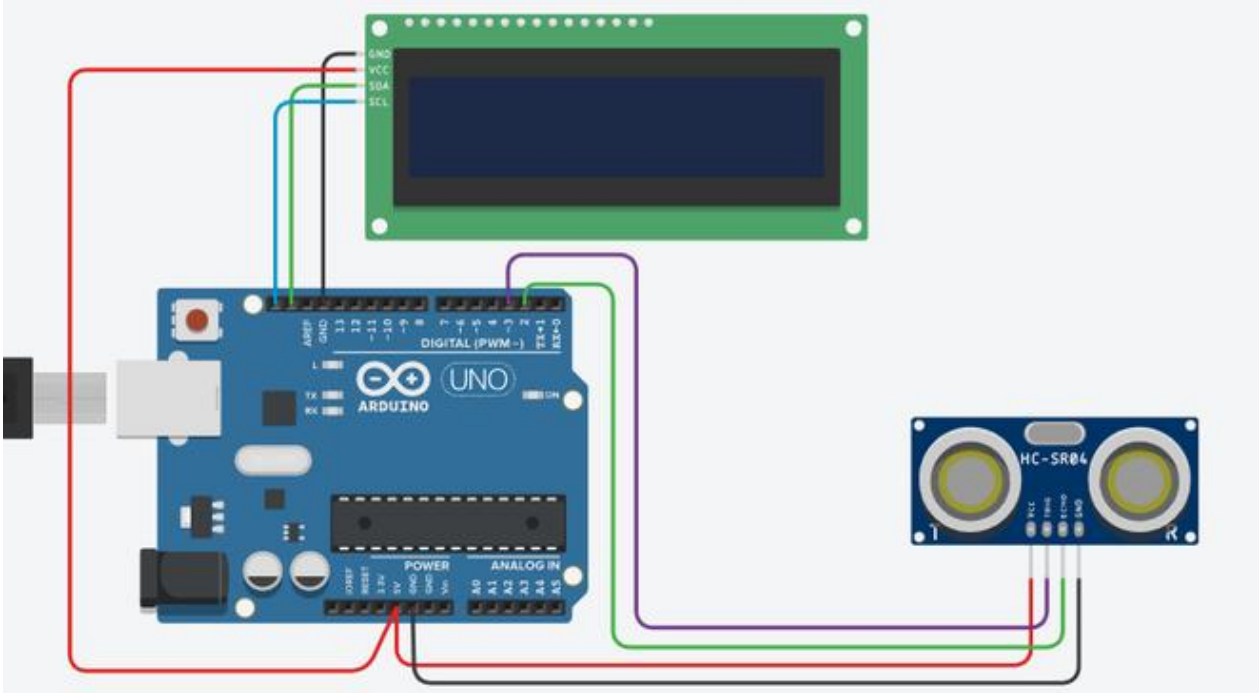

Fig: Ultrasonic Distance Measurement Arduino Circuit

Procedure:

1. **Connect the sensor to the Arduino:**
    - Connect the VCC (5V) pin of the sensor to the 5V pin of the Arduino.
    - Connect the GND pin of the sensor to the GND pin of the Arduino.
    - Connect the Trig pin of the sensor to a digital pin on the Arduino (e.g., pin 9).
    - Connect the Echo pin of the sensor to another digital pin on the Arduino (e.g., pin 10).
    - Connect the VCC pin of the LCD to the 5V pin of the Arduino
    - Connect the GND pin of the LCD to the GND pin of the Arduino.
    - Connect the SDA pin of the LCD to the SDA pin of the Arduino (usually A4 on Uno).
    - Connect the SCL pin of the LCD to the SCL pin of the Arduino (usually A5 on Uno).

2. **Run the experiment:**
    - Open the Serial Monitor in the Arduino IDE.
    - Point the sensor at an object and observe the readings.
    - The LCD Monitor will display the distance to the object in centimeters.

3. **Analysis and Discussion:**

- How accurate are the distance measurements? Compare them to a ruler or other measuring tool.
- What are the limitations of this sensor?

Code:

```
#include <LiquidCrystal_I2C.h>

#define trigPin 9
#define echoPin 10

LiquidCrystal_I2C lcd(0x27, 16, 2); // Set the LCD I2C address (adjust if needed)

void setup()

{
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
```

```
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
}
void loop()

{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  float distance = duration * 0.034 / 2;
  // Display on serial monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  // Display on LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Distance: ");
  lcd.print(distance);
  lcd.print(" cm");
  delay(100);
}
```

Experiment No: 5

Experiment Name: Temperature and Humidity Sensor DHT22 with Arduino Uno

Objectives:

- Understand the working principle of the DHT22 sensor.
- Interface the DHT22 sensor with the Arduino Uno.
- Write Arduino code to read sensor data, calculate temperature and humidity, and display them on the serial monitor. (Optional: Display on an LCD screen)

Materials:

- Arduino Uno
- DHT22 temperature and humidity sensor
- Breadboard
- Jumper wires
- 16x2 I2C LCD display with I2C interface adapter

Circuit Diagram:

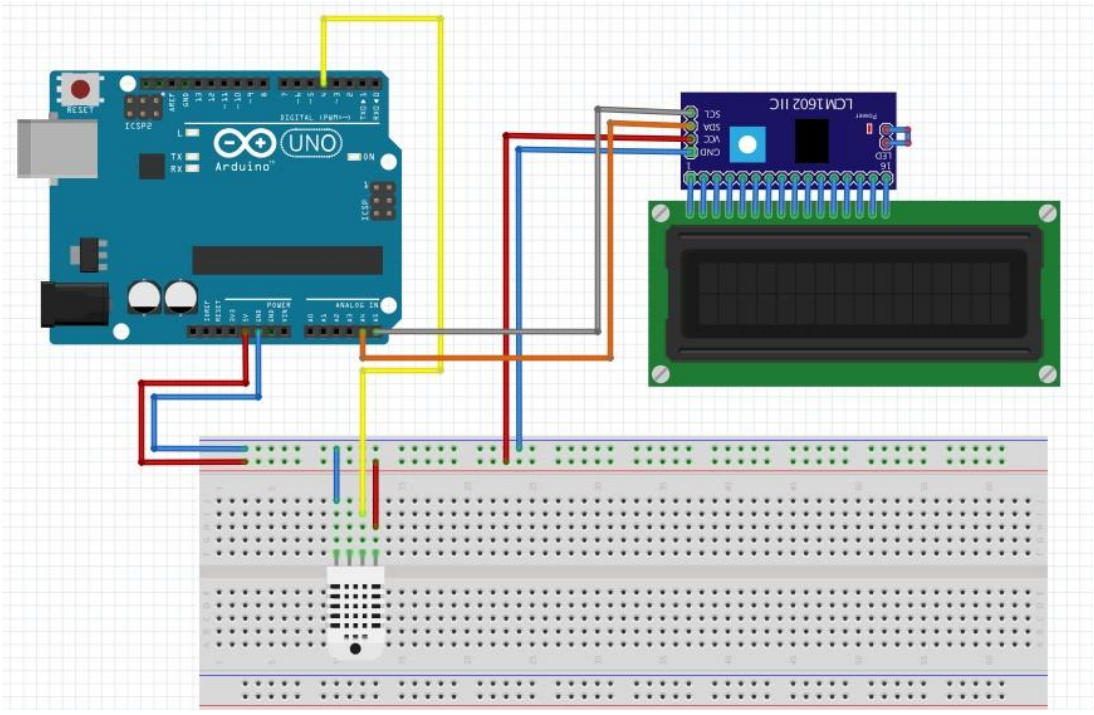

Fig: Arduino DTH22 Humidity Temperature with LCD I2C 16x2 Display Circuit

Procedure:

1. DHT22 Sensor:
    - Connect the VCC pin of the DHT22 to the Arduino's 5V pin.
    - Connect the GND pin of the DHT22 to the Arduino's GND pin.
    - Connect the DATA pin of the DHT22 to Arduino pin 4 (can be changed in the code).
2. I2C LCD Display:
    - Connect the VCC pin of the I2C LCD display to the Arduino's 5V pin.
    - Connect the GND pin of the I2C LCD display to the Arduino's GND pin.
    - Connect the SDA pin of the I2C LCD display to the Arduino's A4 pin (SCL) and the SCL pin of the I2C LCD display to the Arduino's A5 pin (SDA).
3. Testing and Analysis:
    - Open the serial monitor in the Arduino IDE.
    - Observe the temperature and humidity values being printed at regular intervals.

Code:

```
//Libraries
#include <DHT.h>
//I2C LCD:
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display,Pls check your lcd.

//Constants
#define DHTPIN 7     // what pin we're connected to
#define DHTTYPE DHT22   // DHT 22
DHT dht(DHTPIN, DHTTYPE); //// Initialize DHT sensor for normal 16mhz Arduino

//Variables
//int chk;
int h;  //Stores humidity value
int t; //Stores temperature value

void setup()
```

```
{
   Serial.begin(9600);
   Serial.println("Temperature and Humidity Sensor Test");
   dht.begin();
   lcd.init(); //initialize the lcd
   lcd.backlight(); //open the backlight
}

void loop()
{
   //Read data and store it to variables h (humidity) and t (temperature)
   // Reading temperature or humidity takes about 250 milliseconds!
   h = dht.readHumidity();
   t = dht.readTemperature();

   //Print temp and humidity values to serial monitor
   Serial.print("Humidity: ");
   Serial.print(h);
   Serial.print(" %, Temp: ");
   Serial.print(t);
   Serial.println(" ° Celsius");

// set the cursor to (0,0):
// print from 0 to 9:

   lcd.setCursor(0, 0);
   lcd.println(" Now Temperature ");

   lcd.setCursor(0, 1);
   lcd.print("T:");
   lcd.print(t);
   lcd.print("C");

   lcd.setCursor(8, 1);
   lcd.print("H:");
   lcd.print(h);
   lcd.print("%");

  delay(1000); //Delay 1 sec.
}
```

Experiment No: 6

Experiment Name: Soil Moisture Sensor with Arduino Uno

Aim:

- Understand the working principle of a Soil Moisture Sensor.
- Interface the Soil Moisture Sensor with the Arduino Uno.
- Write Arduino code to read sensor data, convert it to moisture percentage, and display it on the serial monitor or an LCD screen.

Materials:

- Arduino Uno
- Soil Moisture Sensor (YL-38 or FC-33)
- Breadboard
- Jumper wires
- LCD screen (optional)

Circuit Diagram:

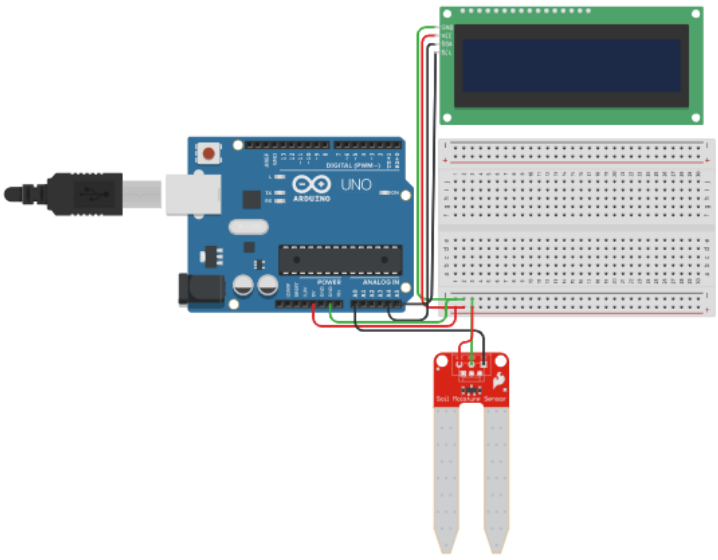

Fig: Arduino Soil Moisture with LCD I2C Display Circuit

Procedure:

1. Soil Moisture Sensor:

    - Connect the VCC pin of the sensor to the Arduino's 5V pin (or external power supply, if needed).
    - Connect the GND pin of the sensor to the Arduino's GND pin.
    - Connect the Analog output pin of the sensor to an analog pin on the Arduino (e.g., A0).

2. Connect the I2C LCD display:
    - Connect the VCC pin to the Arduino's 5V pin.
    - Connect the GND pin to the Arduino's GND pin.
    - Connect the SDA pin to the Arduino's A4 pin (SDA).
    - Connect the SCL pin to the Arduino's A5 pin (SCL).

Code:

```
#include <LiquidCrystal_I2C.h> // Include the LiquidCrystal_I2C library

// Define LCD parameters (adjust based on your specific display)
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Set the LCD I2C address, columns, and rows

int sensorPin = A0;  // Analog pin connected to the sensor
float voltage;
float moisture;

void setup() {
  Serial.begin(9600);
  lcd.begin();         // Initialize the LCD
  lcd.backlight();      // Turn on the LCD backlight
}

void loop() {
  voltage = analogRead(sensorPin);  // Read analog voltage from the sensor
  // Convert voltage to moisture percentage (adjust the formula if needed)
  moisture = (voltage * 100.0) / 1023.0;
```

```
  // Display moisture on LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Soil Moisture:");
  lcd.setCursor(0, 1);
  lcd.print(moisture);
  lcd.print("%");

  Serial.print("Moisture: ");
  Serial.print(moisture);
  Serial.println("%");

  delay(1000); // Delay between readings
}
```