**Live Soil Moisture Monitoring System**

**✅ Project Title:**

**Soil Moisture Live Monitoring System with Status Display and Alerts**

**📌 Objective:**

To continuously monitor the **soil moisture level** using a sensor, display real-time readings on a **16x2 I2C LCD**, and provide **visual (LED)** and **audio (buzzer)** alerts based on moisture conditions: LOW, MODERATE, or HIGH.

**🧾 Components Required**

**(URL:-** [**https://app.cirkitdesigner.com/project/9bcdb34c-3fc7-4a0a-9fb2-fcc3bce55f2d**](https://app.cirkitdesigner.com/project/9bcdb34c-3fc7-4a0a-9fb2-fcc3bce55f2d)**)**

| **Component Name** | **Pin No** | **Destination Component** | **Pin No** | **Special Remark** |
| --- | --- | --- | --- | --- |
| Arduino Uno | A0 | Soil Moisture Sensor | Analog Output | Reads analog moisture level |
| Arduino Uno | 2 | Red LED | + (Anode) | Lights up on LOW moisture |
| Arduino Uno | 3 | Blue LED | + (Anode) | Lights up on MODERATE moisture |
| Arduino Uno | 4 | Green LED | + (Anode) | Lights up on HIGH moisture |
| Arduino Uno | 5 | Buzzer | + | Beeps for MODERATE and HIGH states |
| Arduino Uno (I2C pins) | A4 (SDA), A5 (SCL) | I2C LCD Display (16x2) | SDA, SCL | Shows live moisture level and status |
| All components | GND, VCC | Arduino GND, 5V | GND, 5V | Common ground and power supply |

**🔧 Wiring Diagram Overview**

* **Soil Moisture Sensor**:
  + VCC → 5V
  + GND → GND
  + AO → A0
* **LEDs** (via current-limiting resistors, ~220Ω recommended):
  + Red → Pin 2
  + Blue → Pin 3
  + Green → Pin 4
* **Buzzer**:
  + Positive → Pin 5
  + Negative → GND
* **LCD (I2C 16x2)**:
  + SDA → A4
  + SCL → A5
  + VCC → 5V
  + GND → GND

**💡 Working Principle**

1. The **soil moisture sensor** gives analog values between 0 (wet) and 1023 (dry).
2. The values are **mapped to a percentage (0–100%)** for display.
3. Based on this percentage:
   * **<30%** → Status: LOW → Red LED ON (no buzzer)
   * **30%–60%** → Status: MODERATE → Blue LED + short buzzer beep
   * **>60%** → Status: HIGH → Green LED + short buzzer beep
4. The **LCD alternates** between showing the **moisture level** and **status** every second.
5. **Serial Monitor** also shows real-time values for debugging.

**💾 Required Arduino Libraries:**

You’ll need this library:

* **LiquidCrystal\_I2C**

To install:

1. Open Arduino IDE.
2. Go to **Sketch → Include Library → Manage Libraries…**
3. Search for LiquidCrystal\_I2C
4. Install **by Frank de Brabander** (or compatible).

**🧾 Arduino Code**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#define SOIL\_SENSOR A0

#define RED\_LED 2

#define BLUE\_LED 3

#define GREEN\_LED 4

#define BUZZER 5

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Use 0x3F if 0x27 doesn't work

void setup() {

Serial.begin(9600);

lcd.init();

lcd.backlight();

pinMode(RED\_LED, OUTPUT);

pinMode(BLUE\_LED, OUTPUT);

pinMode(GREEN\_LED, OUTPUT);

pinMode(BUZZER, OUTPUT);

lcd.setCursor(0, 0);

lcd.print("Soil Moisture:");

}

void loop() {

int sensorValue = analogRead(SOIL\_SENSOR);

int moisturePercent = map(sensorValue, 1023, 0, 0, 100); // Dry = 1023, Wet = 0

Serial.print("Moisture: ");

Serial.print(moisturePercent);

Serial.println("%");

// Turn off all LEDs and buzzer

digitalWrite(RED\_LED, LOW);

digitalWrite(BLUE\_LED, LOW);

digitalWrite(GREEN\_LED, LOW);

digitalWrite(BUZZER, LOW);

// Step 1: Display Moisture Level

lcd.setCursor(0, 1);

lcd.print("Level: ");

lcd.print(moisturePercent);

lcd.print("% ");

delay(1000);

// Step 2: Display Status and trigger LED/Buzzer

if (moisturePercent < 30) {

digitalWrite(RED\_LED, HIGH);

lcd.setCursor(0, 1);

lcd.print("Status: LOW ");

delay(1000); // Only LED, no buzzer

}

else if (moisturePercent >= 30 && moisturePercent <= 60) {

digitalWrite(BLUE\_LED, HIGH);

lcd.setCursor(0, 1);

lcd.print("Status: MODERATE");

shortBeep();

delay(1000);

}

else {

digitalWrite(GREEN\_LED, HIGH);

lcd.setCursor(0, 1);

lcd.print("Status: HIGH ");

shortBeep();

delay(1000);

}

}

// Function to make short beep for MODERATE and HIGH

void shortBeep() {

digitalWrite(BUZZER, HIGH);

delay(200);

digitalWrite(BUZZER, LOW);

}

**📊 Expected Output Behavior**

| **Moisture (%)** | **LCD Status** | **LED On** | **Buzzer** |
| --- | --- | --- | --- |
| < 30% | Status: LOW | Red | OFF |
| 30%–60% | Status: MODERATE | Blue | Short Beep |
| > 60% | Status: HIGH | Green | Short Beep |

**✅ Tips for Testing**

* Remove sensor from moist soil to simulate LOW level.
* Insert into wet soil for HIGH.
* Slightly damp soil gives MODERATE.
* You can test on Serial Monitor too.

**📌 Optional Enhancements**

* Use an **SD card module** to log data.
* Add **Bluetooth/Wi-Fi (e.g., ESP32)** for remote monitoring.
* Implement **automated irrigation** on LOW level.

Let me know if you want this as a **PDF**, **Word**, or **Excel** file — or if you'd like me to convert this into a **presentation slide format**.