Smart Irrigation System

Manisha Kaushal(220623)
Table No. 08
EE381: EC Project

1. Problem Statement and Motivation

Watering plants manually often leads to irregularities, overwatering, or underwatering. This can affect plant health, especially when users are away or forget to water. The aim is to automate irrigation using soil moisture sensing to conserve water and ensure timely watering.

2. Existing Solutions and Limitations

Existing Solutions:

- Drip irrigation with timers
- Smart garden kits
- IoT-based automatic irrigation systems

Shortcomings:

- High cost
- Internet dependence
- Limited DIY customization

Why Our Approach is Unique:

- Low-cost, offline solution
- Simple and replicable with Arduino Nano
- Can be upgraded with IoT later

3. Timeline

- Week 1: Circuit testing with soil moisture sensor and relay
- Week 2: LCD integration and pump activation
- Week 3: Final prototype testing and troubleshooting

4. Circuit Diagram

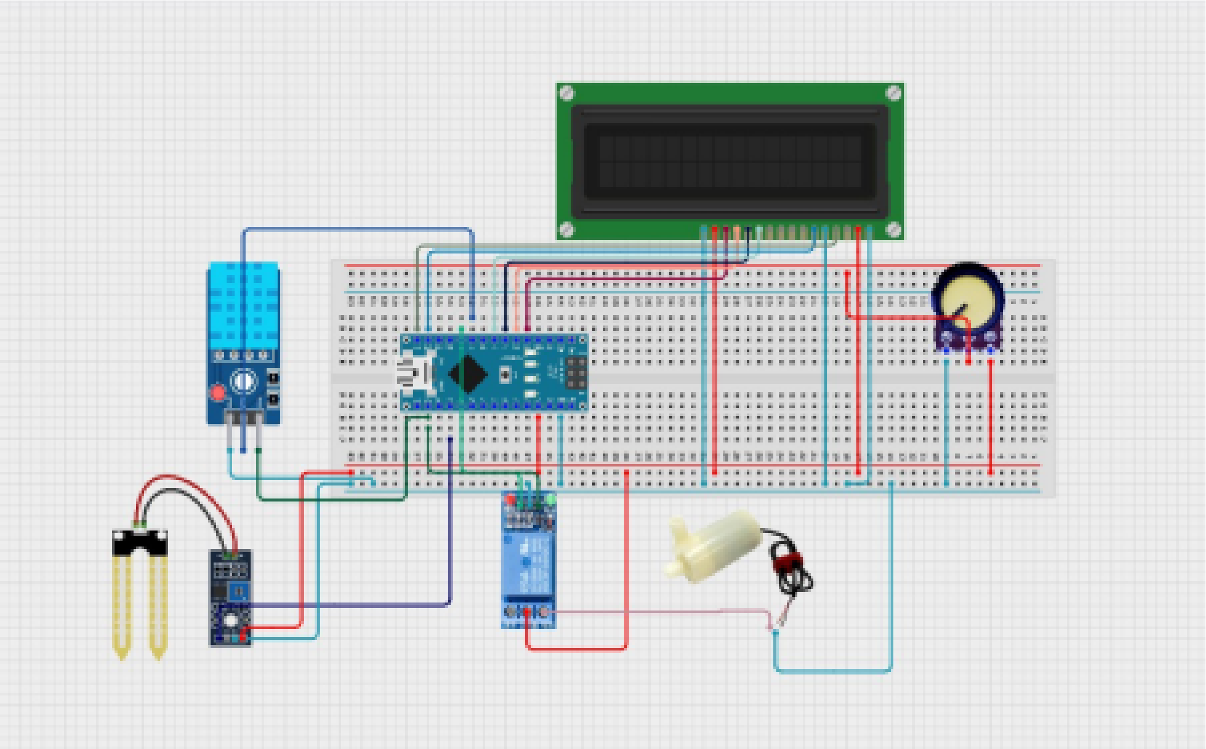

Figure 1: Complete circuit diagram of smart irrigation setup

5. Implementation Details

5.1 System Overview

The Arduino Nano reads the soil moisture level through an analog sensor. If the value is below a certain threshold, the relay turns ON the pump. A 16x2 LCD shows the current moisture level, temperature, humidity, and pump status.

5.2 Arduino Code Summary

- Reads analog soil moisture value
- Uses DHT11 sensor to display temp/humidity
- Controls a relay module to switch ON/OFF pump
- Displays info on 16x2 LCD via parallel communication

5.3 IoT Expansion

- NodeMCU ESP8266 for remote monitoring via Blynk
- Smartphone ON/OFF pump switch and moisture display

Appendix: Arduino Code

```
/************** IoT-Enabled Smart Irrigation System **************/
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal.h>
#include <DHT.h>

// ---------------- WiFi + Blynk Setup ----------------
char auth[] = "X7f8dK9L3mQ2wT1zV6pB5aN0cR";
char ssid[] = "SmartFarm_IoT_WiFi";
char pass[] = "Manisha484#";


// ---------------- Pin Definitions ----------------
#define SOIL_PIN A0
#define RELAY_PIN D1
#define DHTPIN D2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int threshold = 410;  // Soil moisture threshold
int pumpStatus = 0;   // 0 = OFF, 1 = ON

BlynkTimer timer;

// ---------------- Pump Control Function ----------------
void controlPump() {
  int moisture = analogRead(SOIL_PIN);
  float temp = dht.readTemperature();
  float humid = dht.readHumidity();
  // Send data to Blynk App
  Blynk.virtualWrite(V1, moisture);
  Blynk.virtualWrite(V2, temp);
  Blynk.virtualWrite(V3, humid);
  Blynk.virtualWrite(V4, pumpStatus);

  // LCD Display
  lcd.setCursor(0,0);
  lcd.print("M:"); lcd.print(moisture);
  lcd.print(" T:"); lcd.print(temp,0); lcd.print((char)223); lcd.print("C");

  lcd.setCursor(0,1);
  lcd.print("H:"); lcd.print(humid,0); lcd.print("% P:");

  if (moisture > threshold && pumpStatus == 0) {
    digitalWrite(RELAY_PIN, HIGH);  // Pump ON
    pumpStatus = 1;
    Blynk.notify("Pump Turned ON - Soil Dry");
  } else if (moisture <= threshold && pumpStatus == 1) {
    digitalWrite(RELAY_PIN, LOW);   // Pump OFF
    pumpStatus = 0;
    Blynk.notify("Pump Turned OFF - Soil Wet");
  }

  lcd.print(pumpStatus ? "ON " : "OFF");
}

// ---------------- Manual Control from App ----------------
BLYNK_WRITE(V5) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    digitalWrite(RELAY_PIN, HIGH);
    pumpStatus = 1;
  } else {
    digitalWrite(RELAY_PIN, LOW);
    pumpStatus = 0;
  }
}

// ---------------- Setup ----------------
void setup() {
  Serial.begin(9600);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);
  lcd.begin(16,2);
  lcd.print("Smart Irrigation");
  dht.begin();

  Blynk.begin(auth, ssid, pass);
  timer.setInterval(2000L, controlPump); // every 2 sec
}

// ---------------- Loop ----------------
void loop() {
  Blynk.run();
  timer.run();
}
```

```
}
```

Listing 1: Final Arduino Code

6. Results

- Successfully monitored soil moisture and displayed info on LCD
- Pump activated correctly based on soil dryness
- All components worked stably with Arduino Nano

7. Future Improvements

- Replace Arduino Nano with NodeMCU for IoT access
- Add solar power and battery for field deployments
- Implement mobile alerts and automatic daily watering schedule

8. Project Prototype Image

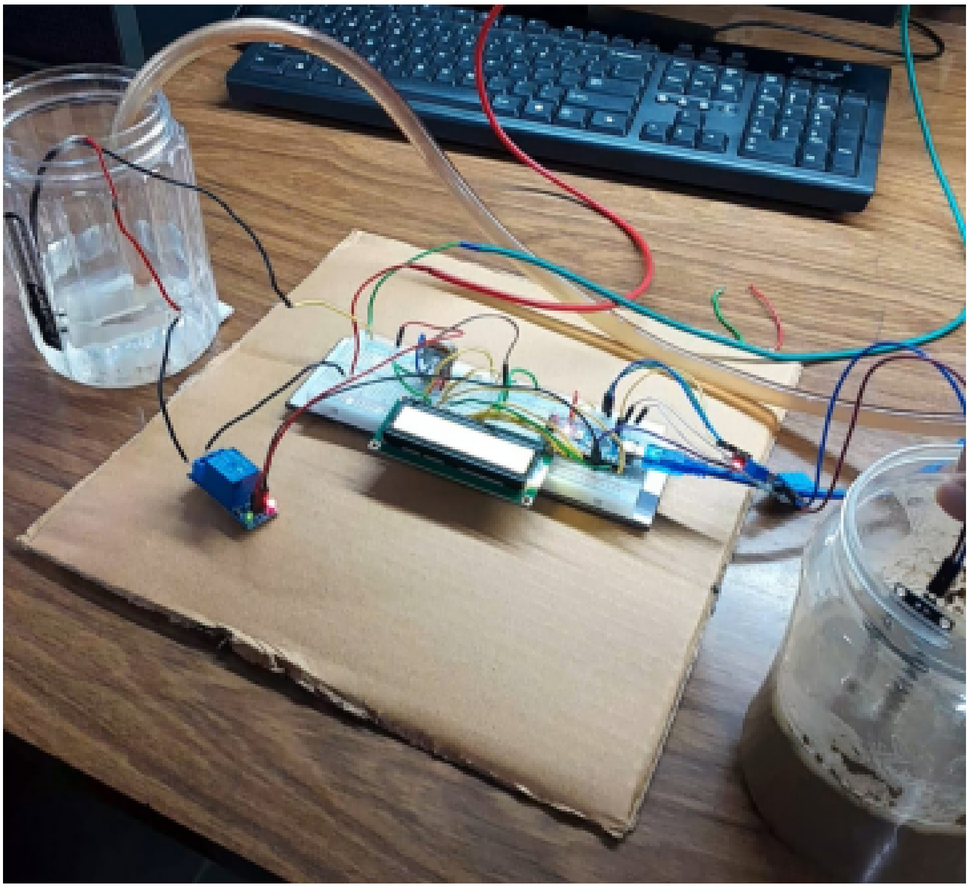

Figure 2: Final working prototype of smart irrigation system

9. Conclusion

The Smart Plant Watering System offers an efficient, low-cost alternative for home or garden automation. It is simple to deploy and can be expanded with IoT capabilities. This solution promotes water conservation while ensuring better care for plants.