I.O.T. PROJECT

SMART IRRIGATION SYSTEM

B.Tech 2nd Year Students :

RITESH (2301EE60)

Smart Irrigation System Using IoT – Detailed Project Report

1. Objective

The objective of this project is to develop a smart irrigation system that leverages IoT and embedded technologies to provide an automated water management solution for agricultural fields, gardens, and home plants. By using sensors to detect real-time soil moisture, temperature, and humidity, this system helps optimize water usage and reduce human effort.

2. Motivation

Manual irrigation systems often result in overwatering or underwatering due to guesswork or unawareness of the soil condition. Water scarcity is a global concern, especially in agriculture. This motivated us to create a system that not only automates watering based on actual soil conditions but also logs data to the cloud for future analytics. The goal is to make farming more efficient, sustainable, and technologically advanced.

3. Importance

This project is essential for modern agriculture, where resource conservation and productivity are critical. Smart irrigation ensures minimal water wastage, enhances crop yield, reduces the need for manual supervision, and supports precision farming techniques. It is scalable and can be implemented in rural farms, urban gardens, and greenhouses alike.

4. Key Features

- Automated soil moisture-based water control
- Temperature and humidity monitoring
- OLED display for real-time feedback
- Data logging on ThingSpeak (IoT cloud)
- Low cost and beginner friendly
- Energy efficient and suitable for remote environments

5. Components Used

- NodeMCU ESP8266 Microcontroller
- Soil Moisture Sensor
- DHT11 Temperature and Humidity Sensor
- SSD1306 OLED Display (0.96”)
- Relay Module
- Mini Water Pump (5V)

- Breadboard and Jumper Wires
- USB Cable and Power Supply

6. Software and Tools Used

- Arduino IDE for writing and uploading the program to NodeMCU
- ThingSpeak Cloud for IoT data monitoring and visualization
- Libraries used: ESP8266WiFi, Wire, Adafruit_GFX, Adafruit_SSD1306, DHT
- Windows OS and USB communication drivers for NodeMCU

7. Working Principle

The system reads data from a soil moisture sensor, and if the value indicates dryness, the NodeMCU triggers a relay which starts the water pump. Simultaneously, the DHT11 sensor collects environmental data which, along with moisture readings and pump status, are displayed on an OLED screen and sent to ThingSpeak. This ensures both real-time decision-making and remote monitoring capability.

8. Applications

- Smart farming and agriculture
- Automated irrigation in gardens and parks
- Greenhouses and controlled agricultural environments
- Remote village farms and urban terrace farming
- Research labs and educational demonstrations

9. Arduino Code

```
// Smart Irrigation System with DHT11, OLED (SPI), Soil Moisture Sensor, and Relay
// Developed for NodeMCU ESP8266

#include <ESP8266WiFi.h>
#include <DHT.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <SPI.h>

// === OLED CONFIG ===
#define SCREEN_WIDTH 128
```

```
#define SCREEN_HEIGHT 64
#define OLED_MOSI  D3  // Data
#define OLED_CLK   D4  // Clock
#define OLED_DC    D8  // Data/Command
#define OLED_CS    D6  // Chip Select
#define OLED_RESET -1  // Not connected
Adafruit_SSD1306 display(SCREEN_WIDTH,
SCREEN_HEIGHT, &SPI, OLED_DC, OLED_RESET,
OLED_CS);

// === DHT Sensor ===
#define DHTPIN D2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// === Soil Moisture Sensor ===
const int soilPin = A0;
const int AirValue = 790;     // Calibrated dry value
const int WaterValue = 390;   // Calibrated wet value
```

```
int soilMoisturePercent = 0;

// === Relay ===
#define relayPin 5 // Pump control (active LOW relay)

// === WiFi & ThingSpeak ===
const char* ssid = "ABC";
const char* pass = "12345566";
String apiKey = "Y61FOA06BK8SOBWS";
const char* server = "api.thingspeak.com";
WiFiClient client;

void setup() {
  Serial.begin(115200);
  dht.begin();
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH); // Turn OFF pump
(active LOW)
```

```
  // Setup OLED Display
  if (!display.begin(SSD1306_SWITCHCAPVCC)) {
    Serial.println("OLED failed to initialize");
  } else {
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 0);
    display.println("OLED Ready");
    display.display();
  }

  // Connect to WiFi
  WiFi.begin(ssid, pass);
  Serial.print("Connecting to WiFi");
  unsigned long startAttemptTime = millis();
  while (WiFi.status() != WL_CONNECTED && millis() -
startAttemptTime < 10000) {
    delay(500);
```

```
    Serial.print(".");
  }
  if (WiFi.status() == WL_CONNECTED){
    Serial.println("\nWiFi Connected");
  }
  else {
    Serial.println("WiFi Connection failed");
  }
  Serial.println("\nWiFi Connected");
}

void loop() {
 // digitalWrite(5,1);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("DHT Read Failed");
    return;
  }
```

```
  // Read soil sensor and convert to percentage
  int soilVal = analogRead(soilPin);
  soilMoisturePercent = map(soilVal, AirValue,
WaterValue, 0, 100);
  soilMoisturePercent = constrain(soilMoisturePercent,
0, 100);

  // Relay logic (Active LOW)
  if (soilMoisturePercent <= 70) {
   digitalWrite(relayPin, LOW); // Turn ON pump
    Serial.println("Pump is ON");
  } else {
   digitalWrite(relayPin, HIGH); // Turn OFF pump
    Serial.println("Pump is OFF");
  }

  // Serial Debug
  Serial.print("Soil Moisture: ");
Serial.print(soilMoisturePercent); Serial.println("%");
```

```
  Serial.print("Humidity: "); Serial.println(h);
  Serial.print("Temperature: "); Serial.println(t);

  // Display on OLED
  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0, 0);
  display.print("Soil RH: ");
display.print(soilMoisturePercent);
display.println("%");
  display.print("Humidity: "); display.print(h);
display.println("%");
  display.print("Temp: "); display.print(t);
display.println(" C");
  display.print("Pump: ");
display.println(digitalRead(relayPin) == LOW ? "ON" :
"OFF");
  display.display();

  // ThingSpeak Update
  if (client.connect(server, 80)) {
```

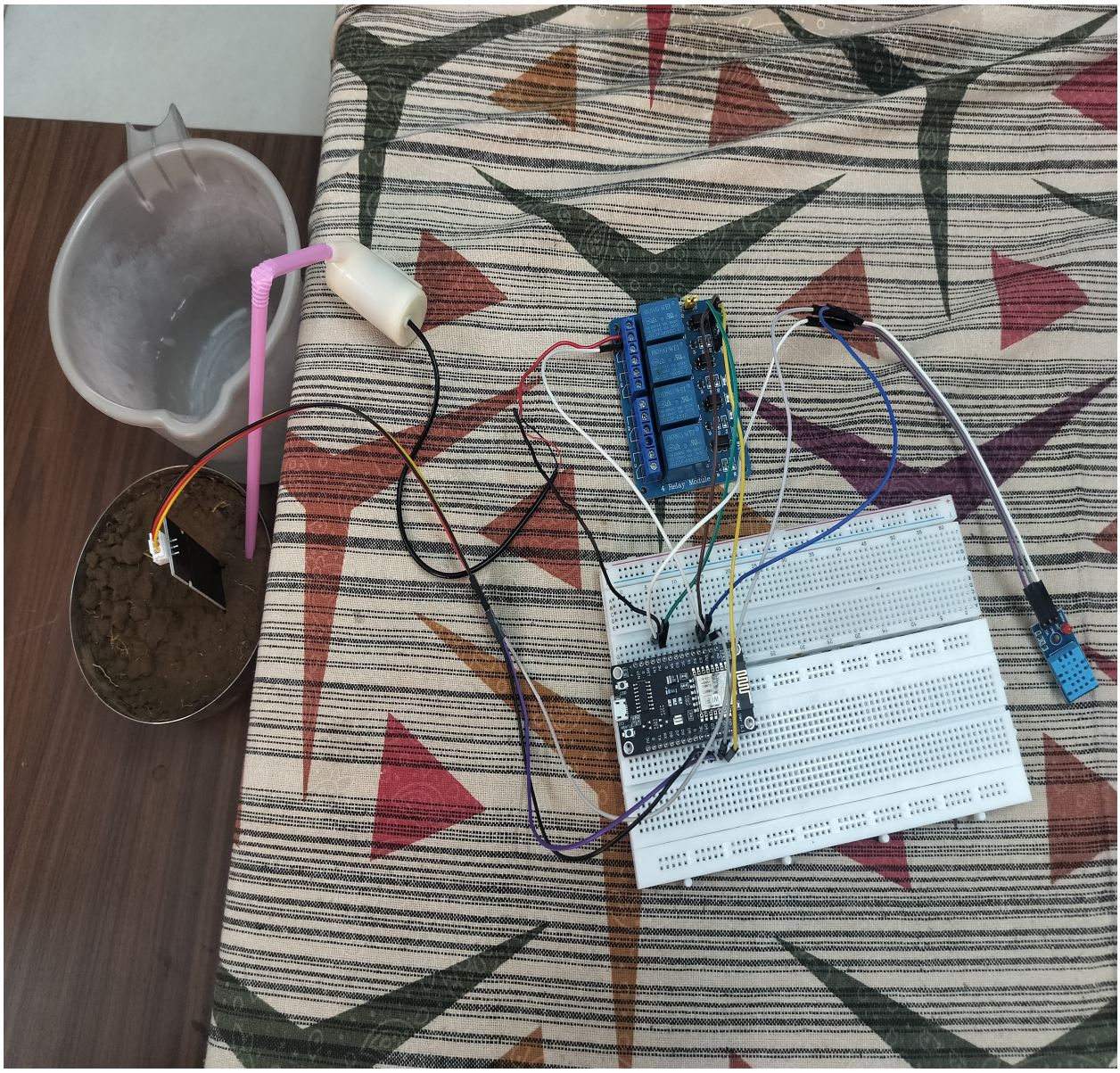
```
    String postStr = apiKey;
    postStr += "&field1=" + String(soilMoisturePercent);
    postStr += "&field2=" + String(h);
    postStr += "&field3=" + String(t);
    postStr += "&field4=" + String(digitalRead(relayPin) == LOW ? 1 : 0);
    postStr += "\r\n\r\n";


    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);
```

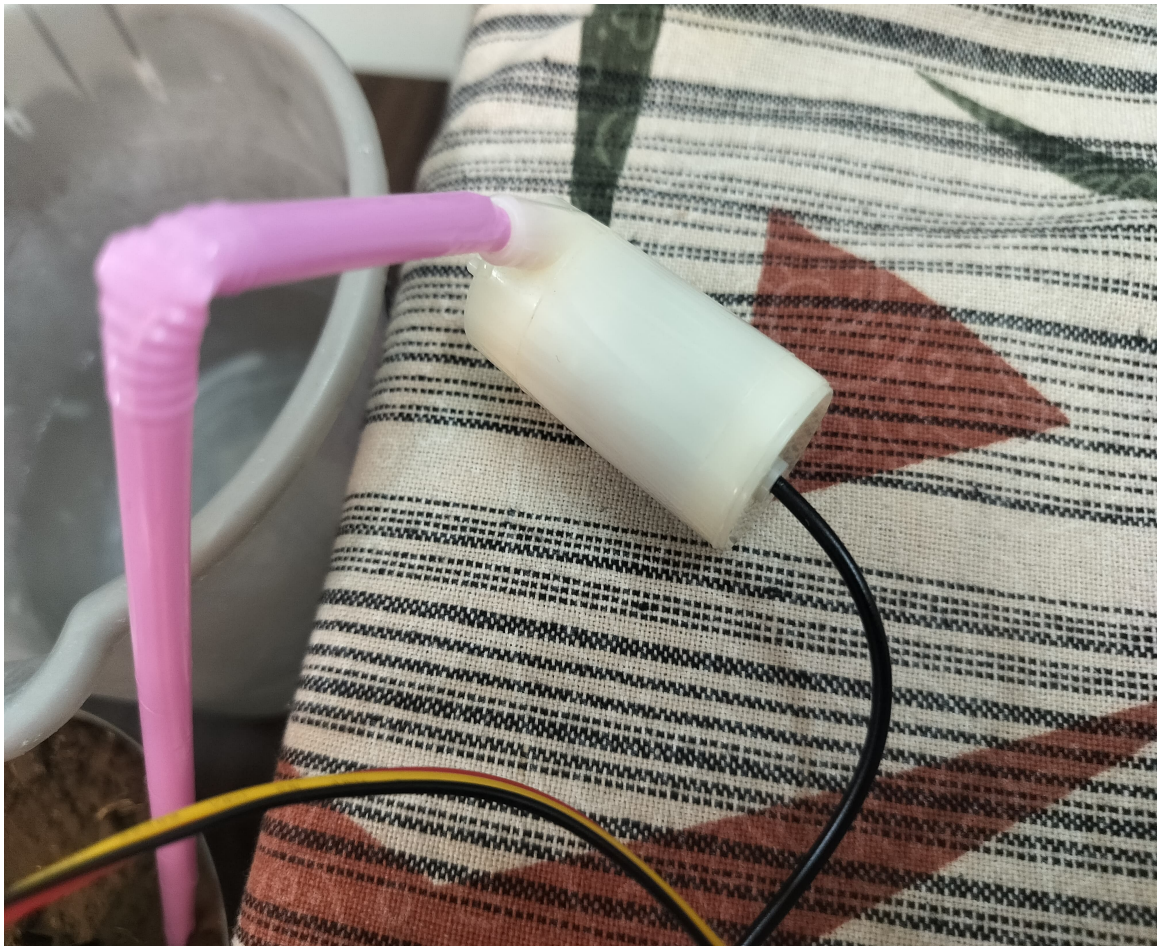
```
    client.stop();
    Serial.println("Data sent to ThingSpeak");
  }

  delay(5000); // 5 second delay
}
```

11. PROJECT IMAGES :

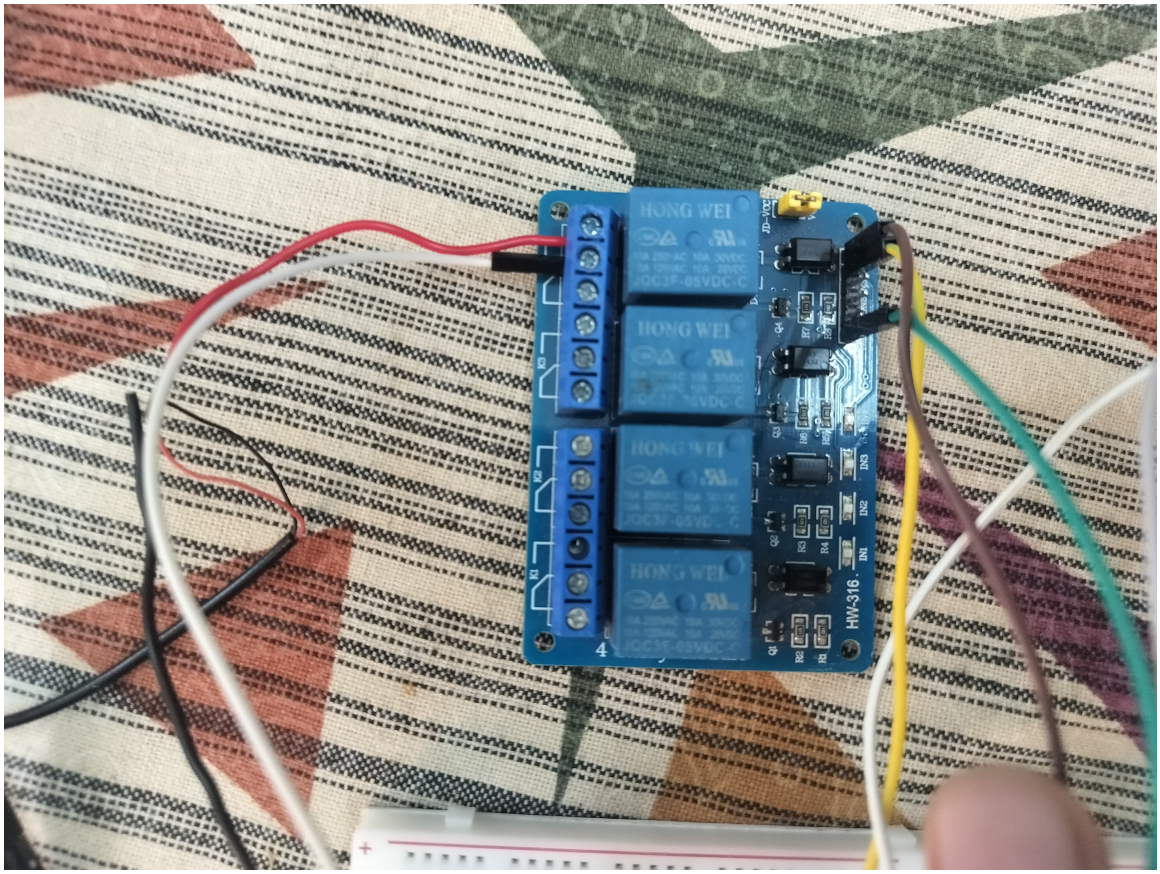


FULL PROJECT IMAGE

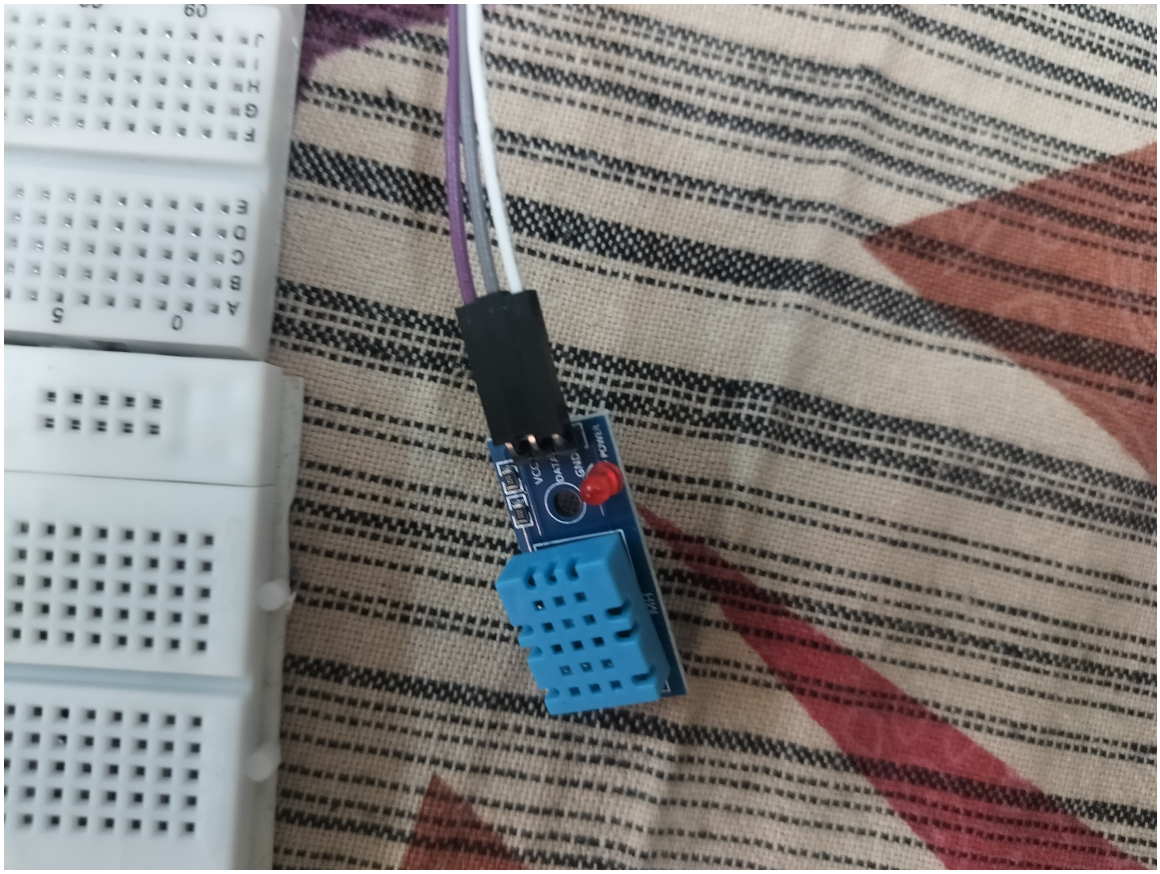


5 VOLT DC PUMP

CAPACITIVE SOIL MOISTURE SENSOR



RELAY MODULE



DHT11 SENSOR

11. Conclusion

This project provides a practical implementation of smart agriculture using affordable electronics and open-source tools. It empowers farmers and enthusiasts to automate irrigation systems with minimal resources and contributes to sustainable farming practices. The integration with IoT platforms also lays the foundation for future expansion into machine learning-based crop prediction and weather integration.