AIR UNIVERISTY, ISLAMABAD
Department of Mechatronics Engineering

	AIR UNIVERSITY
	DEPARTMENT OF MECHATRONICS ENGINEERING
	Project Report

ICT Project
Garden IoT

Introduction:

Global water scarcity is a growing concern, and traditional irrigation methods are often inefficient, relying on manual labor or static timers. This project introduces a "Smart" alternative that uses soil moisture data to drive irrigation decisions. The integration of IoT allows for remote monitoring, data logging, and precision agriculture at an affordable cost.
We will make a smart irrigation system based on ESP32 microcontroller.

Limitations of Traditional Gardening:

1. **Water Inefficiency:** Over-watering leads to resource waste and environmental damage.
2. **Plant Mortality:** Under-watering or inconsistent schedules lead to crop and plant loss.
3. **Manual Dependency:** Traditional gardening requires physical presence and constant monitoring

System Objectives:

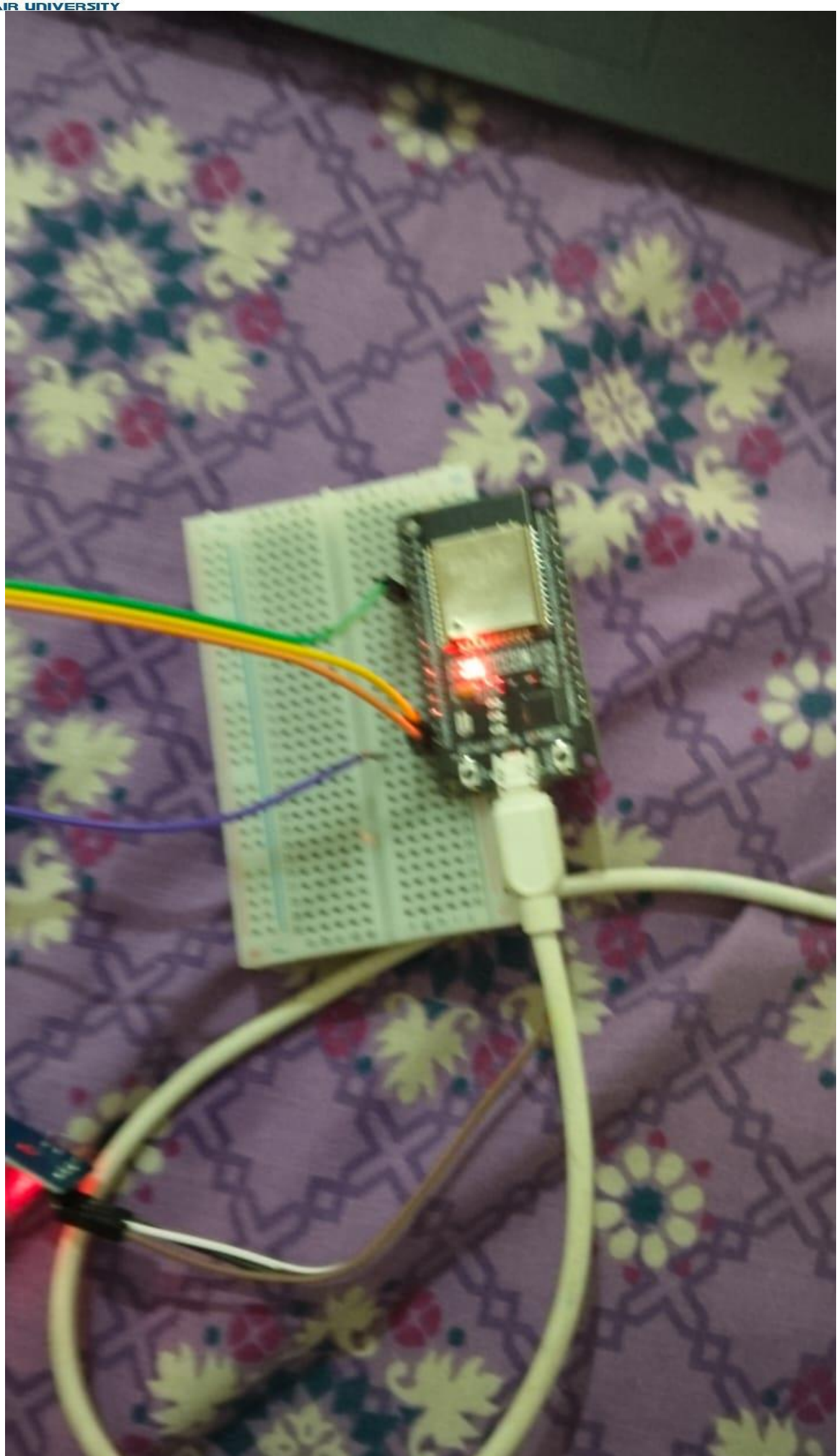
1. To automate the irrigation process based on real-time soil moisture levels.
2. To provide a wireless interface for monitoring environmental conditions.
3. To reduce water consumption by delivering water only when necessary
4. To create a scalable and low-cost hardware architecture.

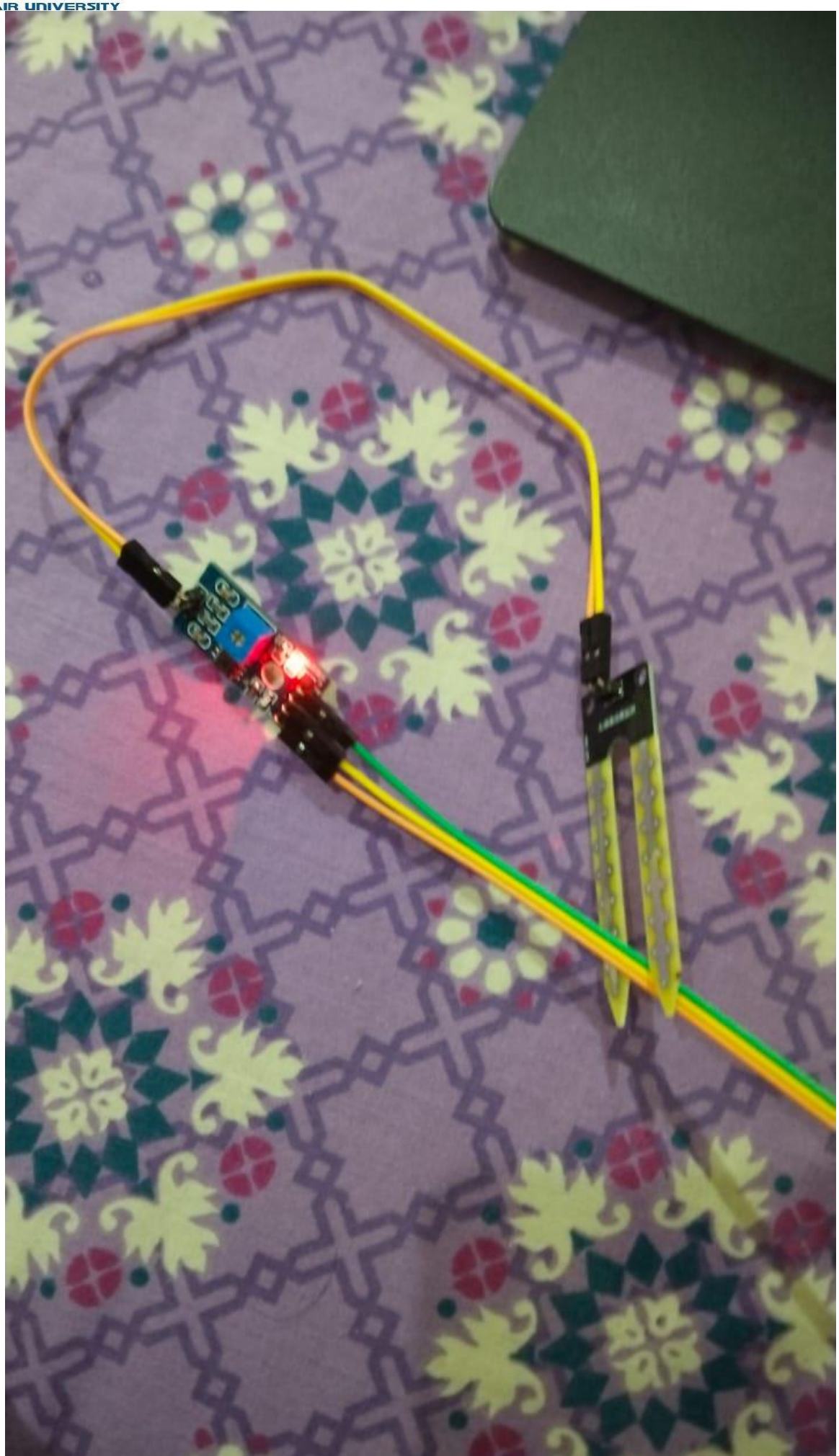
System Architecture and Components:

Hardware Components

1. **ESP32 Microcontroller:** The "brain" of the system, featuring integrated Wi-Fi and Bluetooth.
2. **Capacitive Soil Moisture Sensor:** Measures the water content in the soil.
3. **DHT11 Sensor:** Monitors temperature and humidity.

4. **Relay Module:** Acts as a switch to control the high-voltage water pump
5. **Submersible Pump:** Delivers water to the plants.





Software Components:

1. Arduino IDE
2. Blynk IoT

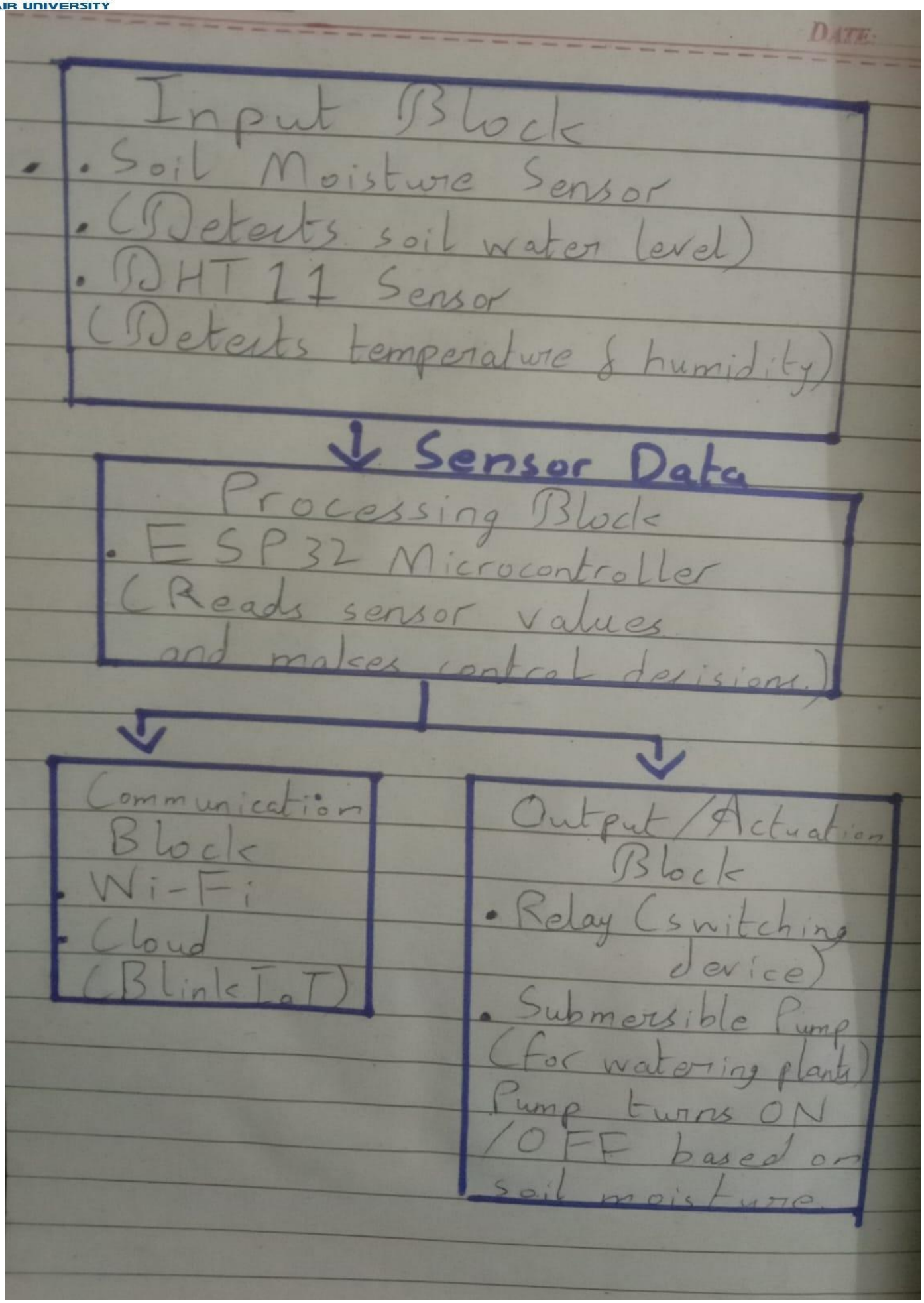
Working:

- **Data Acquisition:** The ESP32 wakes from deep sleep and reads analog values from the soil moisture sensor.

- **Logic Processing:** If the moisture level falls below a pre-defined threshold (e.g., 30%), the ESP32 triggers the relay.

- **Actuation:** The relay turns on the water pump for a set duration.

- **Data Transmission:** The system sends the sensor readings and pump status to a cloud dashboard via Wi-Fi.


DATE:
Input Block
• Soil Moisture Sensor
(Detects soil water level)
• DHT 11 Sensor
(Detects temperature & humidity)
Sensor Data
Processing Block
• ESP32 Microcontroller
(Reads sensor values
and makes control decisions)
Communication Block
• Wi-Fi
• Cloud
(Blink IoT)
Output / Actuation Block
• Relay (switching device)
• Submersible Pump
(for watering plants)
Pump turns ON / OFF based on soil moisture

Code(s):

```
#define BLYNK_TEMPLATE_ID "TMPL6q5AHgkT1"

#define BLYNK_TEMPLATE_NAME "Quickstart Template"

#define BLYNK_AUTH_TOKEN "4ZzV3CvG7WskQuwnCDFvBg37aDI4duXn"


#define BLYNK_PRINT Serial


#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <DHT.h>


// Pin definitions

#define relay 5        // GPIO5 for relay (pump)

#define DHTPIN 18      // GPIO18 for DHT11

#define DHTTYPE DHT11

#define moisturePin 34 // GPIO34 for soil moisture


// ✅ WiFi credentials (UPDATED)

char auth[] = "4ZzV3CvG7WskQuwnCDFvBg37aDI4duXn";

char ssid[] = "Abdul Hadi";

char pass[] = "1234567890";


DHT dht(DHTPIN, DHTTYPE);


void setup() {

  Serial.begin(115200);
```

```
  pinMode(relay, OUTPUT);
  digitalWrite(relay, LOW); // Pump OFF initially

  dht.begin();

  // Connect to Blynk
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  Serial.println("✅ Garden IoT MANUAL Mode Started");
}

void loop() {
  Blynk.run();

  // Read DHT11
  float h = dht.readHumidity();
  float t = dht.readTemperature();     // Celsius
  float f = dht.readTemperature(true); // Fahrenheit

  if (!isnan(h) && !isnan(t)) {
    Blynk.virtualWrite(V7, h); // Humidity
    Blynk.virtualWrite(V8, t); // Temp C
    Blynk.virtualWrite(V3, f); // Temp F

    Serial.print("Humidity: ");
    Serial.print(h);
```

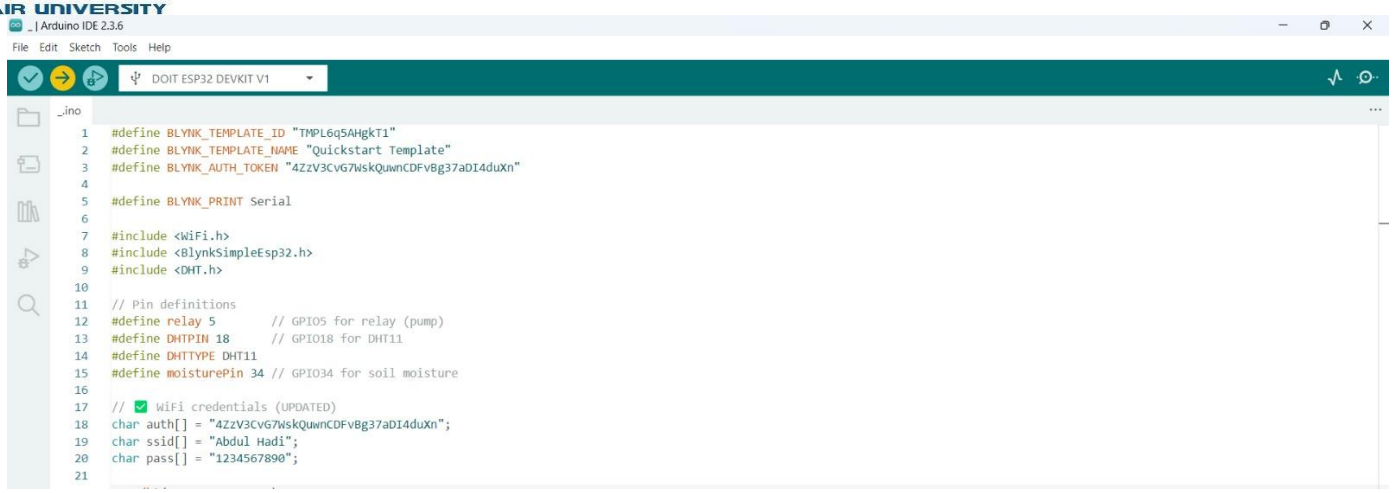
```
    Serial.print("% | Temp: ");
    Serial.print(t);
    Serial.println("°C");
  } else {
    Serial.println("❌ DHT read failed");
  }

  // Read soil moisture (DISPLAY ONLY)
  int sensorValue = analogRead(moisturePin);
  Blynk.virtualWrite(V2, sensorValue);
  Serial.print("Moisture: ");
  Serial.println(sensorValue);

  delay(2000);
}

// ✅ MANUAL pump control ONLY
BLYNK_WRITE(V1) {
  int value = param.asInt(); // 1 = ON, 0 = OFF
  digitalWrite(relay, value);

  Serial.print("🖐 Manual Pump: ");
  Serial.println(value ? "ON" : "OFF");
}
```



Conclusion:

The ESP32 Smart Garden project successfully demonstrates the power of IoT in solving everyday environmental challenges. By combining low-cost sensors with intelligent logic, we can transition from "blind" irrigation to "precision" irrigation, paving the way for smarter, more sustainable cities.