AUTOMATED IRRIGATION CONTROL SYSTEM BASED ON ENVIRONMENT

Team Members:

BALAJI N M - 312323106028

HAFEEZUR RAHIMAN - 312323106052

DILLAN ROMAULD VITUS - 312323106044

ASHWIN SUDHAKARAN - 312323106026

AIM

To design and implement an automated irrigation system using an ESP8266 microcontroller that monitors soil moisture levels in real time and controls a water pump accordingly. The system seeks to ensure efficient water usage, reduce wastage, and provide a low-cost, sustainable solution for smart agriculture and gardening applications.

Tools/Hardware Required

ESP8266
Relay Module
Water Pump
Soil Moisture Sensor
LEDs
Resistor

THEORY

ESP8266 – It is a low-cost Wi-Fi microchip with a full TCP/IP stack and microcontroller capability. It is the central processing unit in this system, responsible for reading data from the turbidity sensor, controlling the LED, and displaying information on the OLED screen. Its built-in Wi-Fi makes it ideal for projects that require internet connectivity, though this particular setup uses it for local processing.

Soil Moisture Sensor - The soil moisture sensor is a transducer that measures the volumetric water content in the soil. It works by passing a small electrical current through the soil between its two probes and measuring the resistance. Wet soil has lower resistance (higher conductivity), while dry soil has higher resistance. The sensor outputs an analog voltage that the ESP8266's ADC (Analog-to-Digital Converter) can read and convert into a digital value to represent the moisture level.

Relay Module - A relay module is an electrically operated switch that allows a low-power microcontroller (like the ESP8266) to control a high-power device (like the water pump). It provides electrical isolation between the two circuits, protecting the sensitive ESP8266 from the higher voltage and current needed by the pump. The ESP8266 sends a low-voltage signal to the relay, which in turn switches on or off the separate, high-voltage circuit connected to the pump.

Water Pump - The water pump is the physical actuator that performs the watering. It's an electromechanical device that, when powered, moves water from a reservoir to the plant. It is controlled by the relay module, turning on only when the soil moisture sensor detects that the soil is dry and the ESP8266 commands it to.

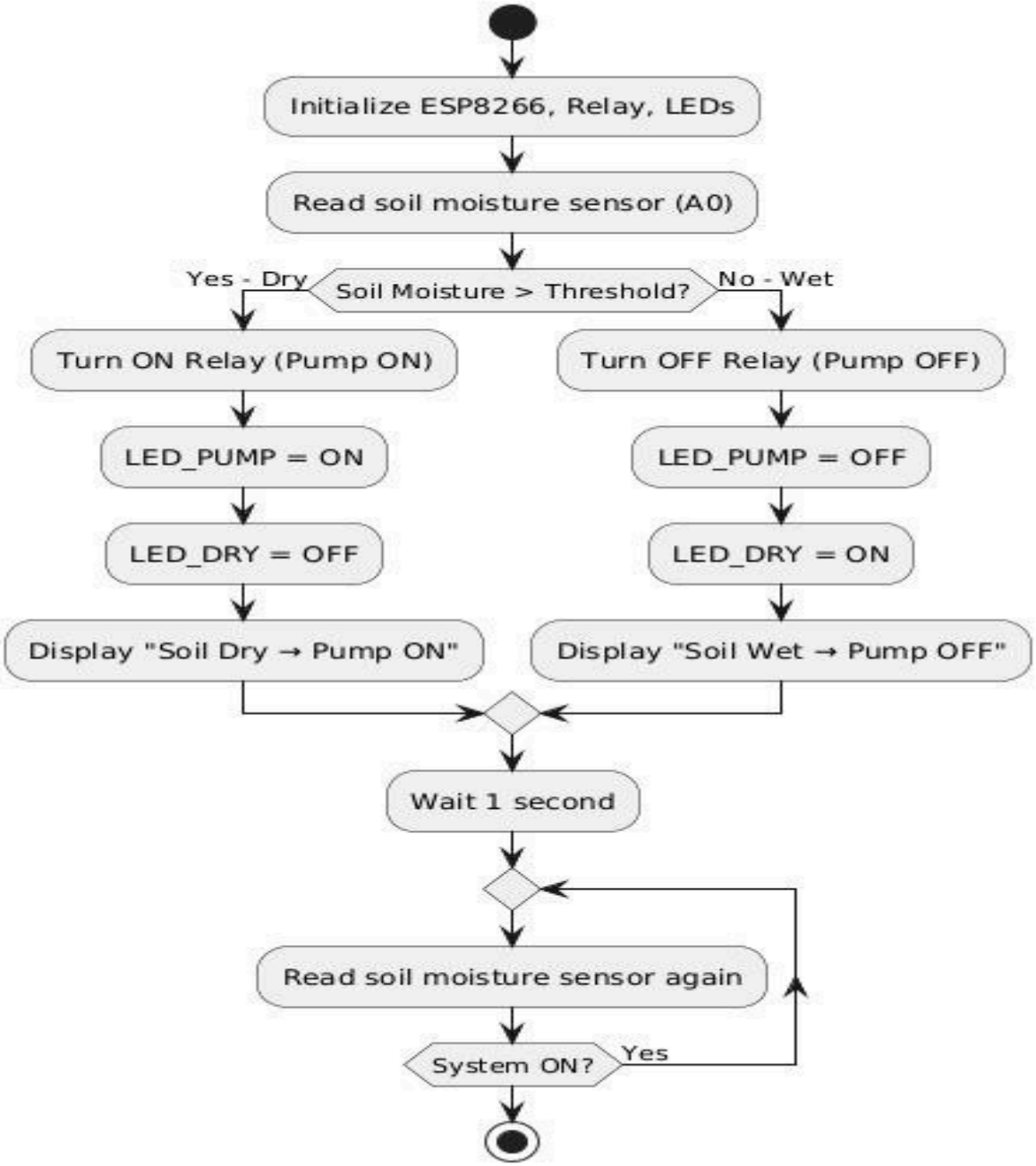
LEDs and Resistors - An LED (Light-Emitting Diode) is a semiconductor that emits light when an electric current passes through it. In this system, LEDs serve as simple visual indicators to show the system's status (e.g., a green LED for "soil is moist" and a red LED for "watering in progress"). A resistor is a passive component used to limit the current flowing to the LED,

preventing it from burning out. The resistor's value is calculated using Ohm's Law to ensure the LED operates at its correct and safe current level.

PIN TABLE

Components	ESP8266 3.3V GND A0
Soil Moisture VCC	3.3V GND D1(GPI05)
Soil Moisture GND	Power Supply
Soil Moisture A0	Relay NO
Relay Module VCC	Power GND
Relay Module GND	D2(GPI04)
Relay Module IN	GND via 10k ohm resistor
Relay COM	D3(GPIO0) GND via 10k
Water Pump(+)	ohm resistor
Water Pump(-)	
Yellow LED Anode	
Yellow LED Cathode	
Red LED Anode	
Red LED Cathode	

FLOWCHART

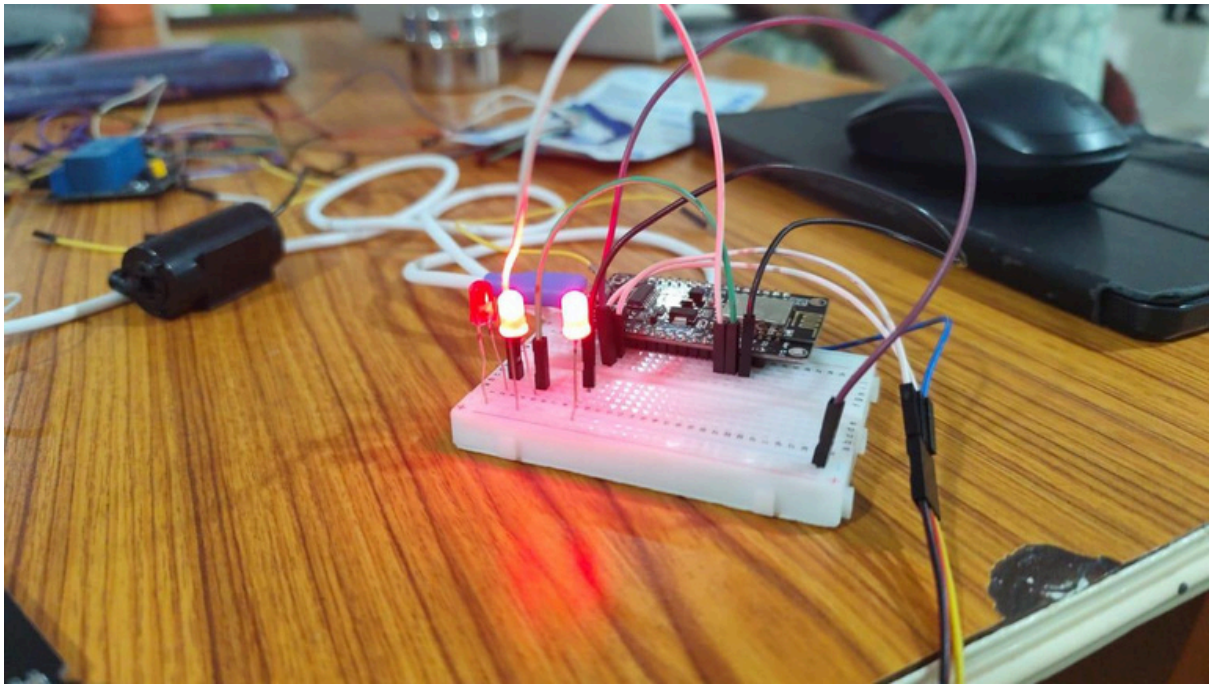

Initialize ESP8266, Relay, LEDs
Read soil moisture sensor (A0)
Soil Moisture > Threshold?
Yes - Dry
No - Wet
Turn ON Relay (Pump ON)
Turn OFF Relay (Pump OFF)
LED_PUMP = ON
LED_PUMP = OFF
LED_DRY = OFF
LED_DRY = ON
Display "Soil Dry → Pump ON"
Display "Soil Wet → Pump OFF"
Wait 1 second
Read soil moisture sensor again
System ON?
Yes

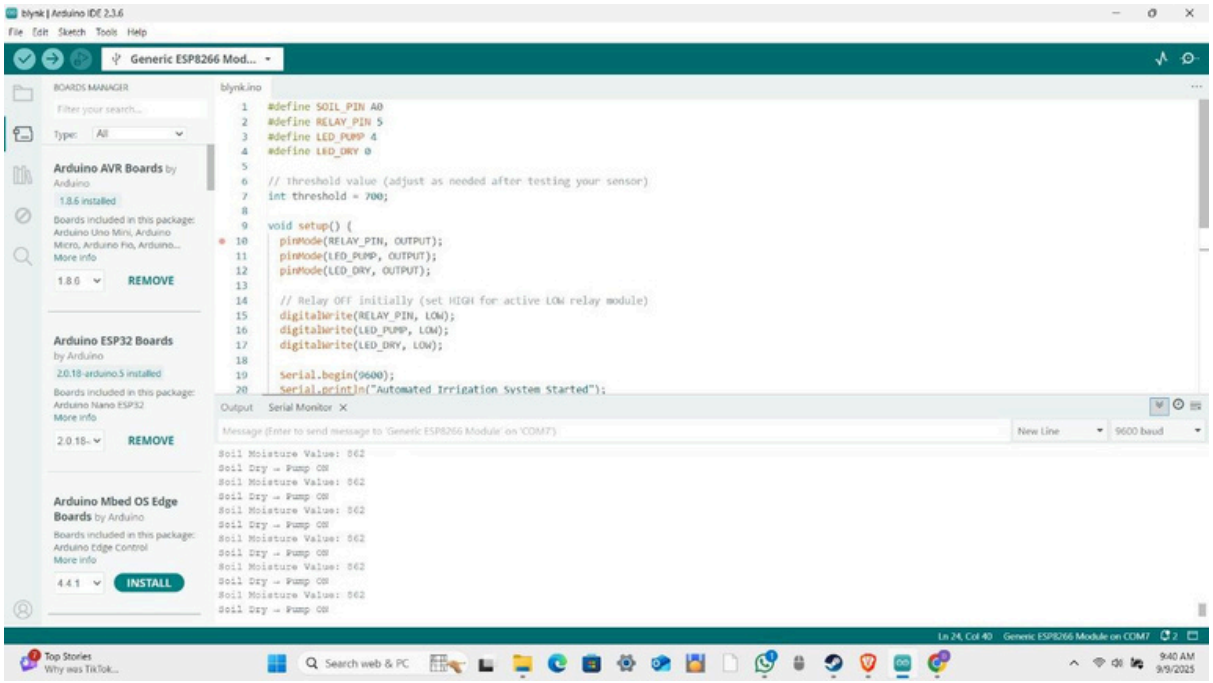
CODE

```
#define SOIL_PIN A0 #define RELAY_PIN 5 #define LED_PUMP 4
#define LED_DRY 0 // Threshold value (adjust as needed after
testing your sensor) int threshold = 700; void setup() {
 pinMode(RELAY_PIN, OUTPUT);
 pinMode(LED_PUMP, OUTPUT);
 pinMode(LED_DRY, OUTPUT);
 // Relay OFF initially (set HIGH for active LOW relay module)
 digitalWrite(RELAY_PIN, LOW);
 digitalWrite(LED_PUMP, LOW);
 digitalWrite(LED_DRY, LOW);
 Serial.begin(9600);
 Serial.println("Automated Irrigation System Started");
}
void loop() {
 int soilValue = analogRead(SOIL_PIN);
 Serial.print("Soil Moisture Value: ");
 Serial.println(soilValue);
 // Dry soil (value > threshold) → Pump ON
 if (soilValue > threshold) {
  digitalWrite(RELAY_PIN, HIGH); // Relay ON (active LOW)
  digitalWrite(LED_PUMP, HIGH);
```

```
 digitalWrite(LED_DRY, LOW);
   Serial.println("Soil Dry → Pump ON");
 } else {
   // Wet soil (value ≤ threshold) → Pump OFF
   digitalWrite(RELAY_PIN, LOW); // Relay OFF
   digitalWrite(LED_PUMP, LOW);
   digitalWrite(LED_DRY, HIGH);
   Serial.println("Soil Wet → Pump OFF");
 }
 delay(1000);
}
```

DEMONSTRATION





EXECUTION

Initialization (Setup Stage): The Arduino board initializes three output pins: RELAY_PIN → controls the water pump through a relay module. LED_PUMP → indicates the pump status. LED_DRY → indicates soil dryness. Initially, the pump is OFF (relay set to HIGH for active LOW module). Both LEDs are turned OFF. Serial communication is started at 9600 baud rate to display sensor readings and system
status.

Reading Soil Moisture (Loop Stage):

The soil moisture sensor connected to pin A0 is read using analogRead().

The sensor returns a value in the range 0 to 1023.

Lower values indicate wet soil.

Higher values indicate dry soil.

The sensor reading is printed on the Serial Monitor.