IOT BASED AUTOMATIC PLANT WATERING SYSTEM

Abstract: This project proposes an IoT-based solution for automatic watering of plants using Arduino Uno, soil moisture sensor, relay module, and a mini water pump. The system continuously monitors the soil moisture level and automatically activates the water pump when the soil becomes dry, ensuring optimal hydration for plants. By automating the irrigation process, the system effectively reduces human effort and prevents water wastage. The project demonstrates how simple, low-cost components can be integrated to create a smart and sustainable solution for everyday plant care, making it suitable for both home gardens and small-scale farming applications.

Objective: The main objective of this project is to automate plant irrigation using IoT technology to maintain ideal soil moisture without manual monitoring. It aims to enhance convenience for users who may not be available to water plants regularly, especially in urban households or small agricultural setups. The system is designed to provide a reliable, efficient, and eco friendly method of watering, ensuring consistent plant growth and resource optimization.

Significance: This system is significant because it highlights the practical benefits of IoT in sustainable agriculture and smart living. By minimizing water wastage and manual labor, it supports environmental conservation and encourages the adoption of smart farming practices even at a small scale. The project also serves as a foundation for more advanced applications such as remote monitoring, data-driven irrigation, and large-scale smart farming systems — demonstrating the growing importance of automation in agriculture and daily life.

Table of Contents:-

List of Figures, Tables, Abbreviations:

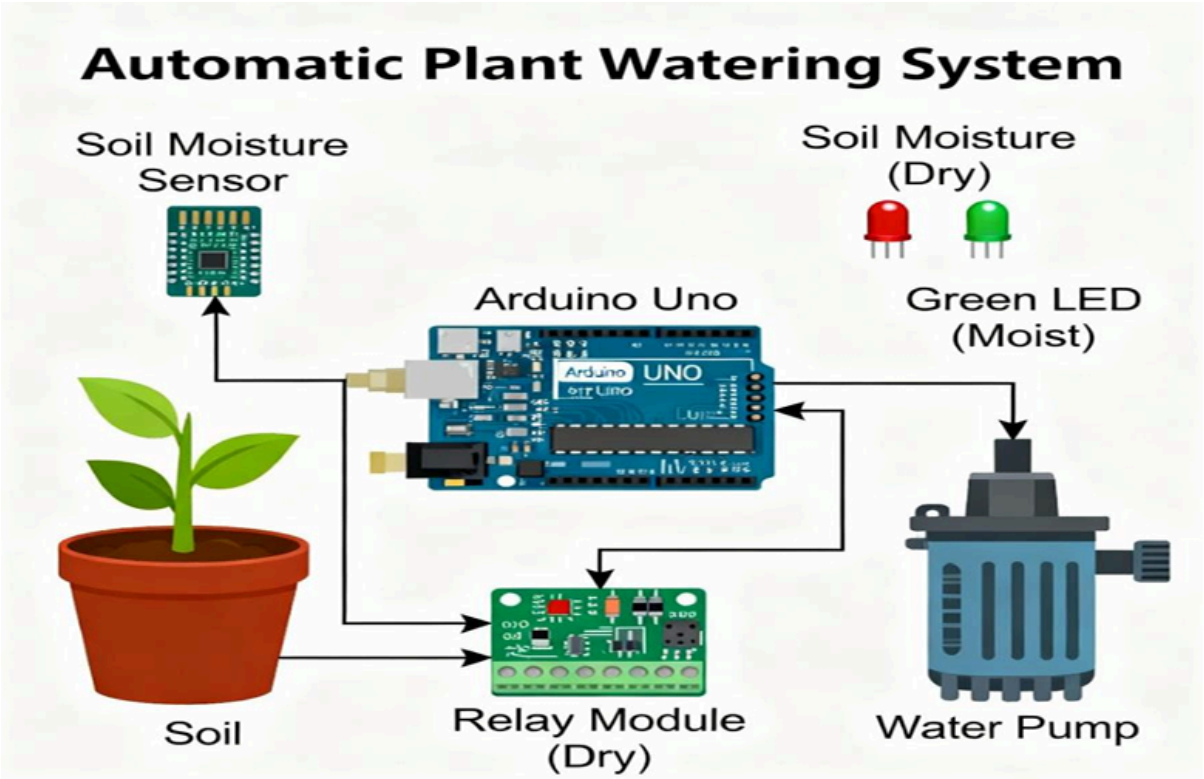
Figure: Tinkercad Circuit Diagram
Table: Component List
Abbreviations: LED (Light Emitting Diode), IDE (Integrated Development Environment), IoT (Internet of Things), VCC (Voltage Common Collector)

1)Introduction:- IoT enables smart automation across industries; here, it resolves the challenge of manually watering plants, especially when owners are absent. The problem addressed is inconsistent irrigation causing plant stress. The objective is reliable, automated plant watering with simple setup. Scope: home and small-scale use; limitation: suitable mainly for small gardens or pots.

2) Literature Review / Existing Systems:- Existing solutions use timers or basic moisture sensors without full automation. Comparatively, this system integrates real-time sensing and decision-making using microcontrollers (Arduino Uno), which improves accuracy. Research trends favour IoT enabled smart farming for resource efficiency.

3) System Design:-

Block Diagram:



Architecture:- Central Arduino receives sensor data, actuates relay/pump based on readings.

Hardware Requirements:
1.Arduino Uno
2.Soil Moisture Sensor
3.Relay Module
4. 5V Mini Pump
5. Jumper wires
6.Breadboard
7.Power Source.

Software Requirements:
1.Arduino IDE for programming
2.Tinkercad Circuits for simulation.

4) Methodology:-

Working Principle:

1. Project Initialization
- Open Tinkercad Circuits.
- Create a New Circuit workspace.
- This will be the base where all components are placed and wired.

2. Adding Components
From the components panel, search for "Arduino Uno" and drag-drop it onto the workspace.
Next, search for "Soil Moisture Sensor" and add it.
Use the Zoom controls (+ / –) in Tinkercad to adjust the view.
Then, search for "Breadboard" and place it on the workspace.
On this breadboard, add two LEDs (LED1 and LED2) — typically used to indicate system states (e.g., "Dry = Red ON, Moist = Green ON").

3. Wiring the Circuit

Soil Moisture Sensor:
- VCC → 5V on Arduino
- GND → GND on Arduino
- AO (Analog Output) → A0 on Arduino

LEDs on Breadboard:
- Connect each LED in series with a resistor (220Ω or 330Ω).
- One LED (Red) connected to digital pin D7 for "Dry Soil" indication.
- One LED (Green) connected to digital pin D8 for "Moist Soil" indication.

Optional Pump Simulation:

- Instead of using a real pump in Tinkercad, you can use an LED or DC motor to represent the pump.
- Controlled via a transistor or relay module connected to another digital pin (e.g., D9).

4. Programming Logic (Arduino Sketch Flow)

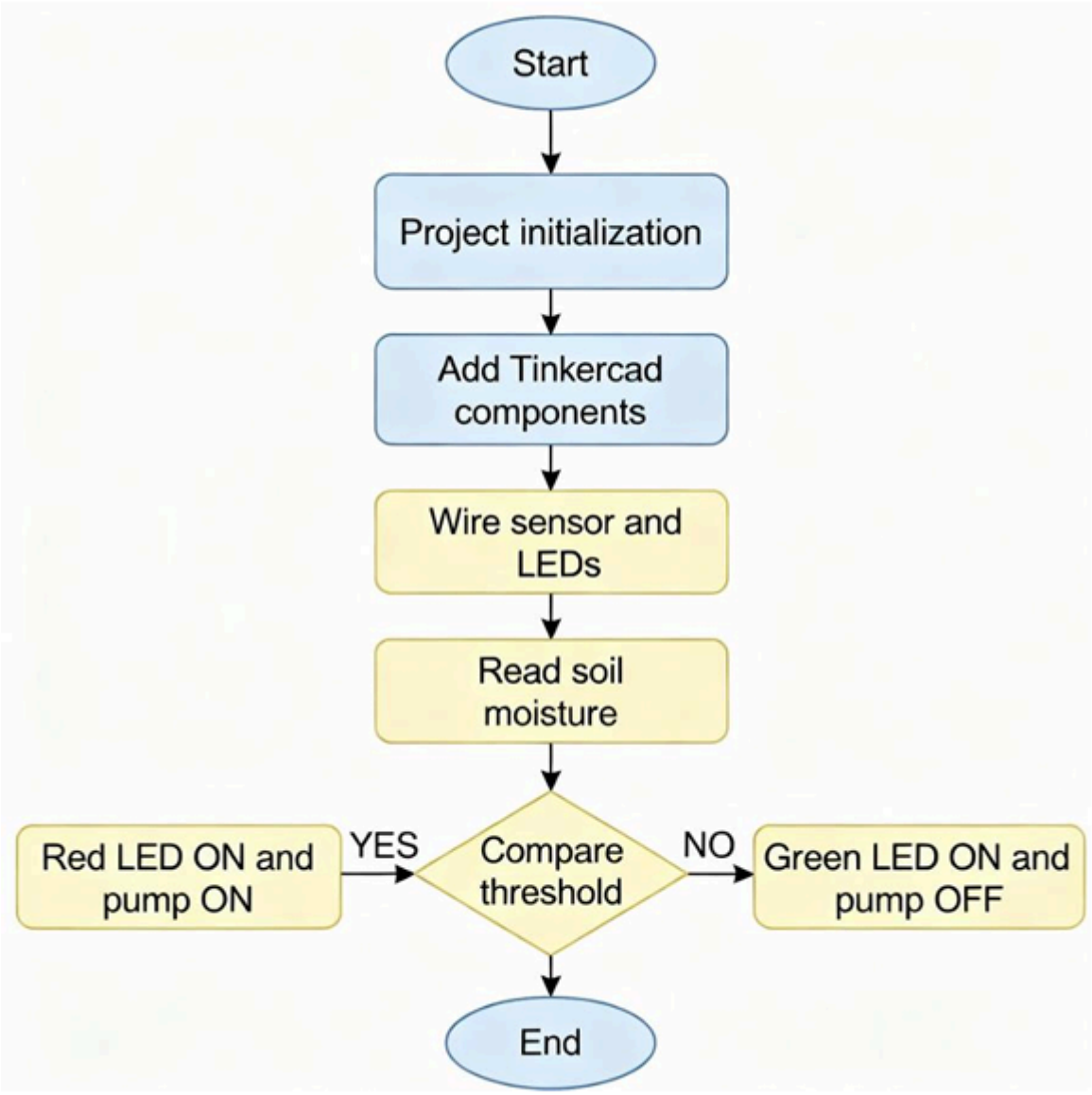
1. Setup Phase:

- Initialize serial communication for monitoring.
- Define pin modes for sensor input (A0) and LEDs (D7, D8).

2. Loop Phase:

- Read the analog value from the soil moisture sensor (0–1023).
- Compare the value to a threshold (decides dry vs moist soil).
- If value < threshold (soil is dry): Turn ON Red LED.
- Activate Pump (or simulated motor/LED).
- Else (soil is moist):
- Turn ON Green LED.
- Turn OFF Pump.

Flowchart / Algorithm:



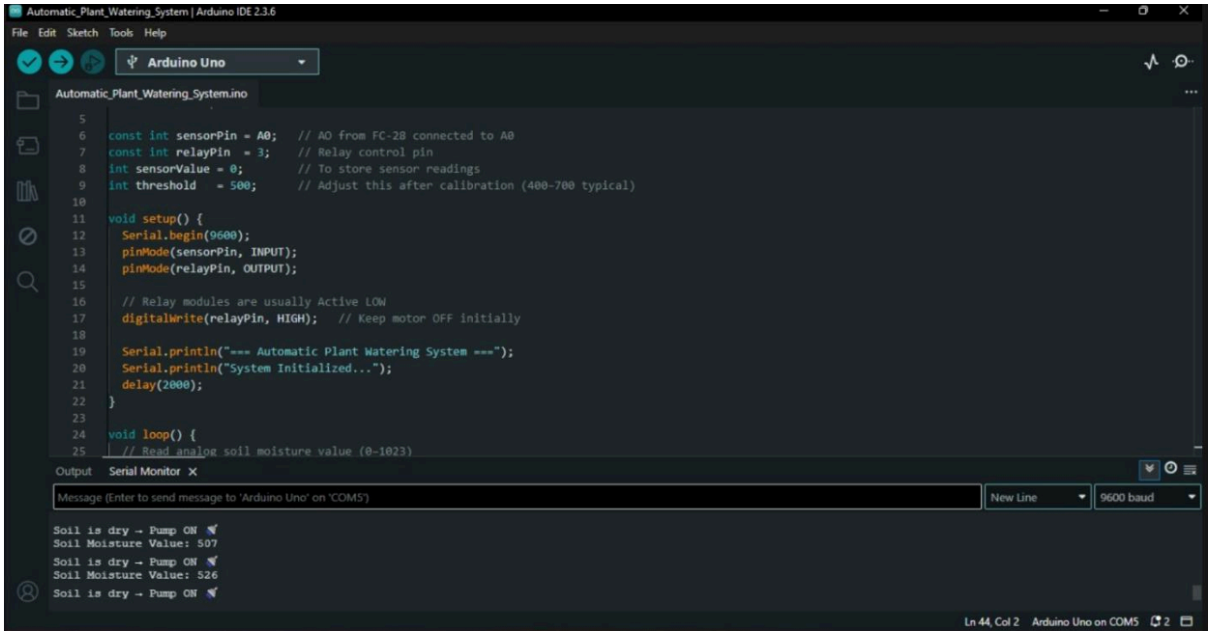
5) Implementation:-

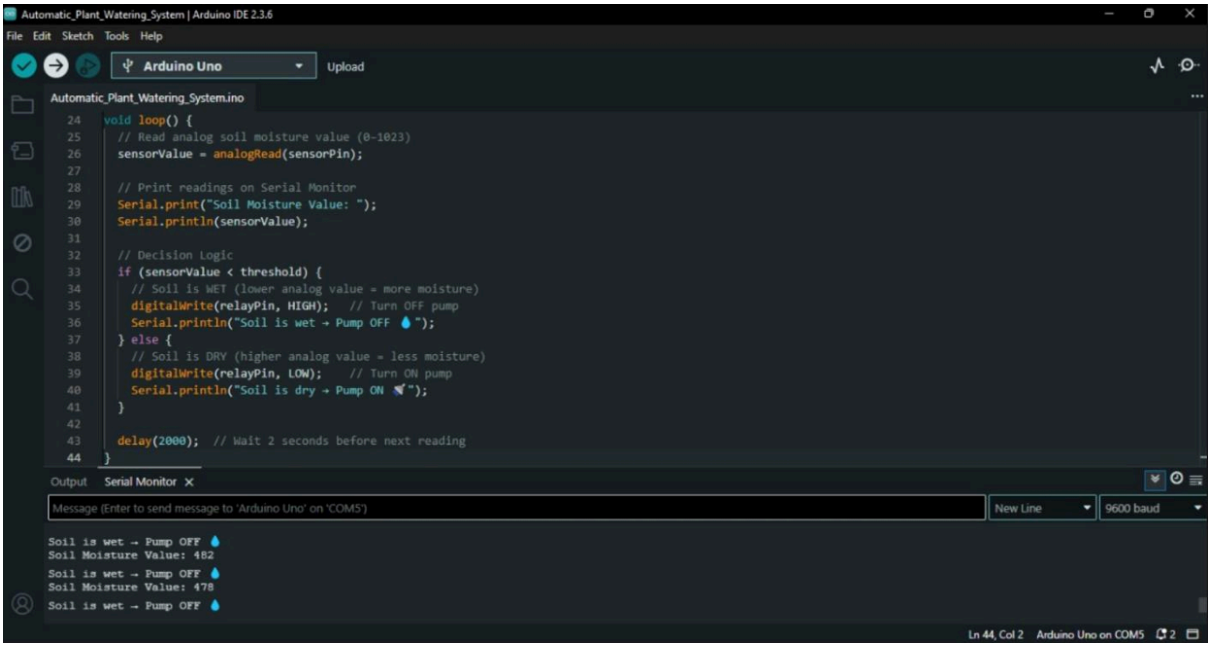
Hardware Setup: Circuit includes Arduino, soil moisture sensor, relay, pump, LEDs for dry/moist indication.

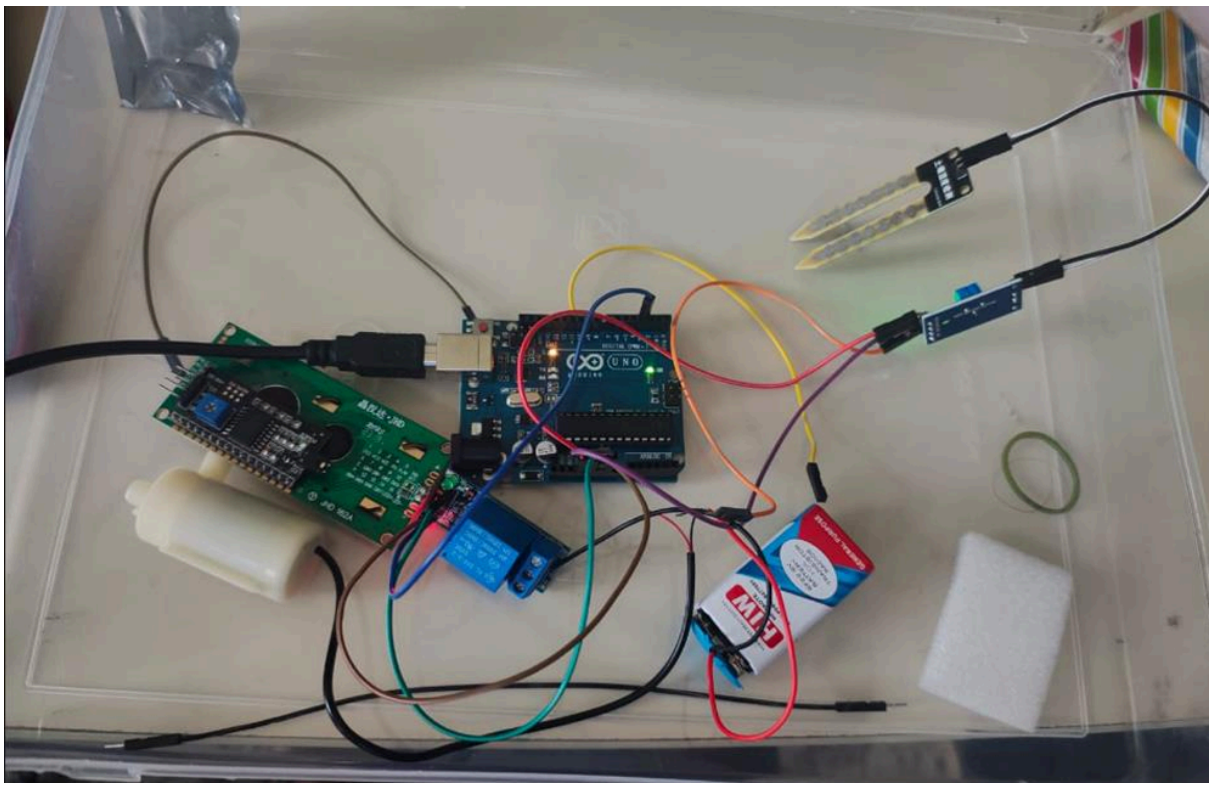
Software Coding & Integration: Arduino sketch sets pin modes, reads sensor, controls pump and LEDs, serial output for monitoring.

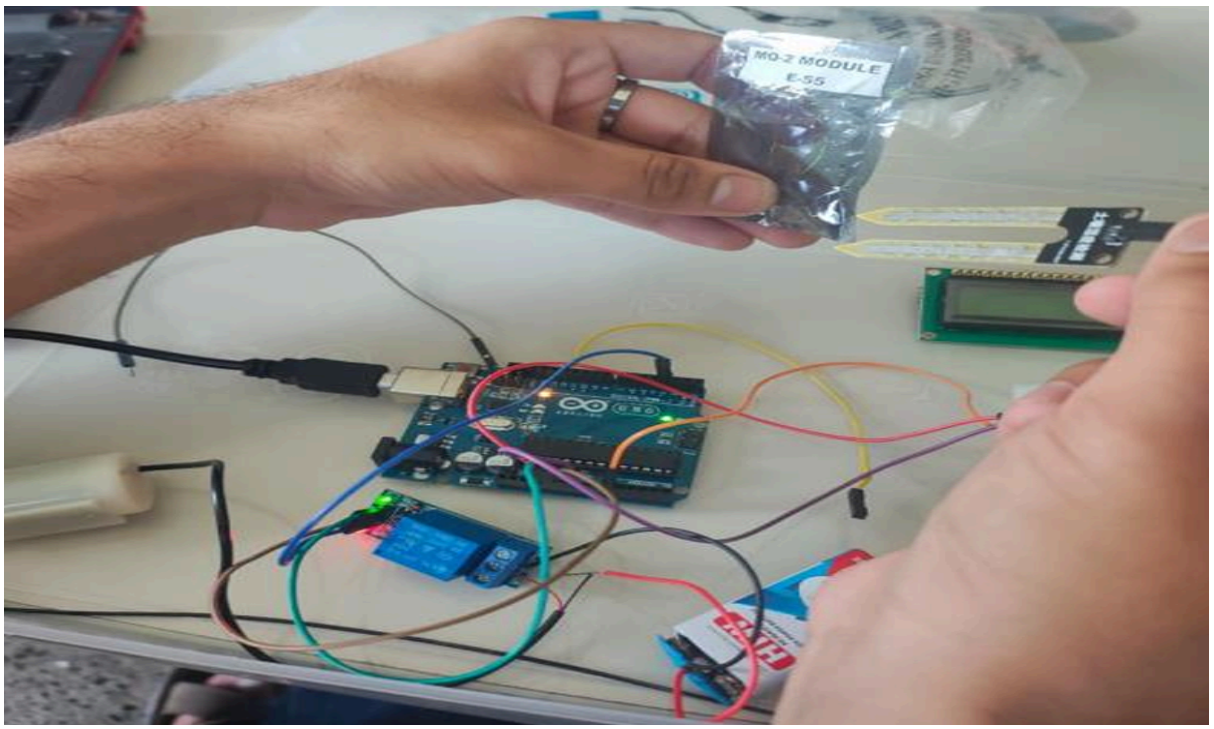
Communication Protocols: Within circuit

6) Results:-


Automatic_Plant_Watering_System | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Arduino Uno
Automatic_Plant_Watering_System.ino
const int sensorPin = A0;   // AO from FC-28 connected to A0
const int relayPin = 3;   // Relay control pin
int sensorValue = 0;   // To store sensor readings
int threshold   = 500;   // Adjust this after calibration (400-700 typical)
void setup() {
Serial.begin(9600);
pinMode(sensorPin, INPUT);
pinMode(relayPin, OUTPUT);
// Relay modules are usually Active LOW
digitalWrite(relayPin, HIGH);   // Keep motor OFF initially
Serial.println("=== Automatic Plant Watering System ===");
Serial.println("System Initialized...");
delay(2000);
}
void loop() {
// Read analog soil moisture value (0-1023)
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM5')
New Line
9600 baud
Soil is dry → Pump ON
Soil Moisture Value: 507
Soil is dry → Pump ON
Soil Moisture Value: 526
Soil is dry → Pump ON
Ln 44, Col 2 Arduino Uno on COM5


Automatic_Plant_Watering_System | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Arduino Uno
Upload
Automatic_Plant_Watering_System.ino
void loop() {
// Read analog soil moisture value (0-1023)
sensorValue = analogRead(sensorPin);
// Print readings on Serial Monitor
Serial.print("Soil Moisture Value: ");
Serial.println(sensorValue);
// Decision Logic
if (sensorValue < threshold) {
// Soil is WET (lower analog value = more moisture)
digitalWrite(relayPin, HIGH);   // Turn OFF pump
Serial.println("Soil is wet → Pump OFF");
} else {
// Soil is DRY (higher analog value = less moisture)
digitalWrite(relayPin, LOW);   // Turn ON pump
Serial.println("Soil is dry → Pump ON");
}
delay(2000); // Wait 2 seconds before next reading
}
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM5')
New Line
9600 baud
Soil is wet → Pump OFF
Soil Moisture Value: 482
Soil is wet → Pump OFF
Soil Moisture Value: 478
Soil is wet → Pump OFF
Ln 44, Col 2 Arduino Uno on COM5

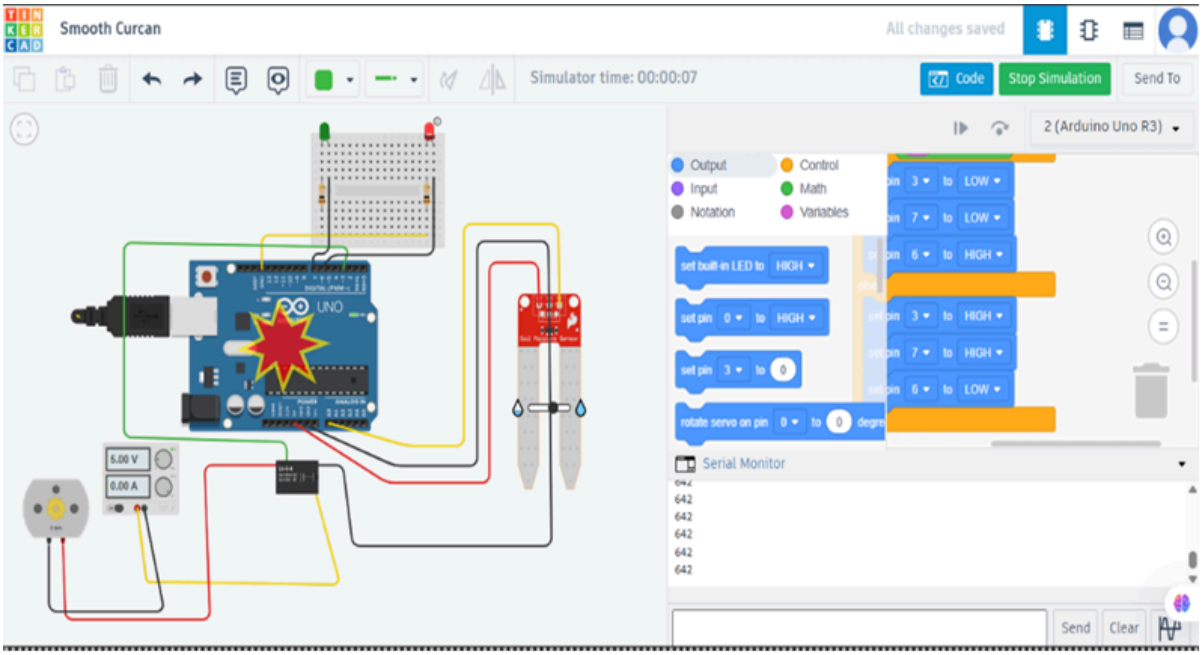



MQ-2 MODULE
E-55

Test Cases & Outputs:

- Run the simulation in Tinkercad.
- Adjust the soil moisture sensor slider to simulate dry and wet soil.
- Observe LEDs and pump behavior changing accordingly.

Tinkercad:-



CODE:-

```
// C++ code

int soil = 0;

void setup()
{
pinMode(A0, INPUT);
Serial.begin(9600);
pinMode(3, OUTPUT);
pinMode(7, OUTPUT);
pinMode(6, OUTPUT);
}
```

```
void loop()
{
soil = analogRead(A0);
Serial.println(soil);
if (soil >= 500)
{
digitalWrite(3, LOW);
digitalWrite(7, LOW);
digitalWrite(6, HIGH);
}
else
{
digitalWrite(3, HIGH);
digitalWrite(7, HIGH);
digitalWrite(6, LOW);
}
delay(10); // Delay a little bit to improve simulation performance }
```

Performance Analysis:

a) System reliably responds to moisture changes.
b) Robust for small-scale deployment.

7) Applications:-

Home gardening automation
Small-scale greenhouse irrigation
Urban agriculture
School/college planting projects

8) Advantages:-

a) Simple
b) Cost-effective
c) Minimal maintenance
d) Saves water
e) Scalable.

9) Conclusion:-

The Smart Plant Watering System successfully demonstrates the application of IoT and automation in sustainable plant care. By continuously monitoring the soil moisture level and controlling the water supply through a microcontroller, the system minimizes manual effort and prevents both over- and under-watering of plants. The use of low-cost components such as Arduino Uno, a soil moisture sensor, and a relay-controlled water pump makes the setup simple, affordable, and efficient for household and small-scale agricultural use. Overall, the project achieves its goal of conserving water, reducing human intervention, and ensuring healthier plant growth through an automated irrigation approach.