MSA University
Faculty of Engineering
Mechatronics Systems Engineering Program

MSE 252 (Group No. 3)

Project Title: Smart Irrigation System

by:

Cadet: Khaled Diaa Eldin 247181

Cadet: Joudy Ahmed 245133

Cadet: Abdulrahman Khairy 249915

Cadet: Seifeldin Eslam Abbas 249437

Cadet: Ali Abdelelah Ali 244093

Under Supervision of:

Assoc. Prof. Dr. Mohamed Atef Ismail Kamel

Eng. Ziena Mahmoud, B.Sc.

Eng. Mariam Amro, B.Sc.

Abstract

Our project “Smart Irrigation System" aims at optimizing irrigation water as well as improving environmental conditions for plants through various environmental sensors. The system uses an ultrasonic sensor to detect water levels in the irrigation tank, ensuring that the water level does not go below a level that would impair the functionality of the irrigation pump. A temperature sensor helps identify environmental factors to determine suitable plants as well as exchange information related to adaptations to environmental changes. A soil moisture sensor determines water levels in the soil, ensuring that irrigation only takes place when necessary. A rain sensor also helps detect rainfall, which allows it to modify environmental covering for plants, ensuring that those that are sensitive to rain are shielded while those that love water remain exposed. The system has a microcontroller ESP32 that oversees data acquisition, decision-making, and wireless communication. The development of a graphical user interface (GUI) is also part of this project that will enable the visualization of real-time data from the sensors, among other functions.

Table of Contents

Chapter 1. Introduction and Literature Review

1.1 Overview

Smart irrigation systems uses Internet of Things (IoT) technologies, such as wireless communication, networked controllers, and sensors, to maintain optimum growth conditionsand real-time monitoring to supply the plants or crops with its requirements. By utilising data-driven decision-making for manual or fixed-schedule watering, these systems seek to improve plant health and minimise water waste. For instance, a systematic review discovered that irrigation control is increasingly impedding embedded systems that uses IoT, Machine Learning (ML), or Deep Learning (DL). Furthermore, when compared to traditional methods, one study found that using an IoT-based smart irrigation system increased crop productivity by 35% and reduced water consumption by almost 50%.

1.2 Historical Background

Irrigation systems were once independent. Initially, irrigation relied on timed control devices or daily schedules, with a person supervising the watering of the plants daily at specific times. Watering occurred at predetermined intervals, regardless of soil type or environmental conditions. With the development of sensor systems over time, irrigation became more responsive and easier thanks to timers and soil moisture sensors, making watering optional based on soil conditions at specific times. More recently, it has become possible to develop prototypes and practical agricultural systems thanks to the growth of IoT platforms, wireless communications, and low-cost microcontrollers such as Arduino/ESP.

1.3 Recent Trends

To increase irrigation accuracy and dependability, recent research highlights the use of several environmental sensors (soil moisture, ambient temperature/humidity, rain detection, and water-tank level) and their combinations. While many systems track soil moisture, fewer fully integrate features like communication technologies and a wider environmental context, according to a survey of IoT work in smart farm irrigation. Artificial intelligence (AI) and computer vision are increasingly being used for plant-health monitoring and irrigation. For instance, an experiment was made to show how to analyze soil color and determine irrigation needs under various lighting and soil conditions using a vision-based system installed on a Raspberry Pi. Another analysis of AI and IoT in precision irrigation pointed out that while new developments (such as edge computing, 5G, blockchain, and solar-powered systems) are being investigated, access and scalability issues still exist.

1.4 Project Objective

The main objective of this project is to design and implement a smart irrigation system that has specific purpose such as:

1- To design an automated irrigation system without the need of the human follow up that basically based on real-time environmental data.
2- To monitor soil moisture, temperature, humidity, rainfall and water level using specific sensors.
3- To reduce any type of wastage.
4- To implement closed-loop control system that responds dynamically to changing the environmental conditions.
5- To develop a low-cost, energy-efficient and scalable system suitable for agricultural applications.

Chapter 2. Proposed System Design

2.1 Mechanical System Design

The mechanical design of a Smart Irrigation System is based on designing mechanical systems involving water storage, water supply, and water flow control mechanisms. The design must aid water conservation, and protection of electronic components.

1- Water storage unit

Figure 2 – water storage unit

water storage container is used to represent the irrigation water source. The tank provides a stable supply of water for the system, maintaining consistent operation of the pump. The mechanical design of the tank focuses on preventing leakage, structural stability, and easy integration with the pumping unit and sensors

2- Pumping system

Figure 3- pumping system

A low-voltage DC water pump is employed to transfer water from the storage tank to the irrigation outlet. The pump has a mechanical mounting that eliminates any vibrations and misalignments while the pump operates. The pump to be used must have the right flow rate and head to promote the efficiency of water transfer while conserving energy.

3- Flow control mechanism

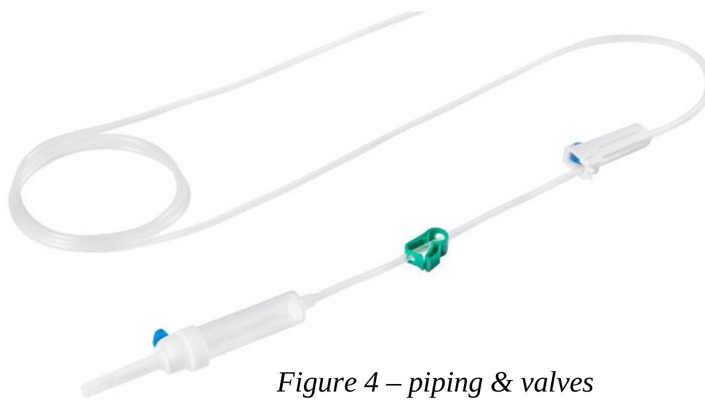

Figure 4 – piping & valves

Water is conveyed from the pump to the irrigation area by a network of flexible tubing, with silicone pipes. In this design, flexible tubing is laid down so that the pressure loss can be minimized, and there are no sharp elbows to create friction and impede smooth discharge. The system is designed such that every future upgrade could be integrated into it, from other advanced techniques to drip irrigation. A solenoid valve is installed within this system that regulates water delivery. This allows for an automated valve operation based on the microcontroller input and control, which enables the scheduling of irrigation with much accuracy. The combined piping and flow control system reduces water losses and further enhances the irrigation system efficiency.

4- Mechanical Housing and Structural Support

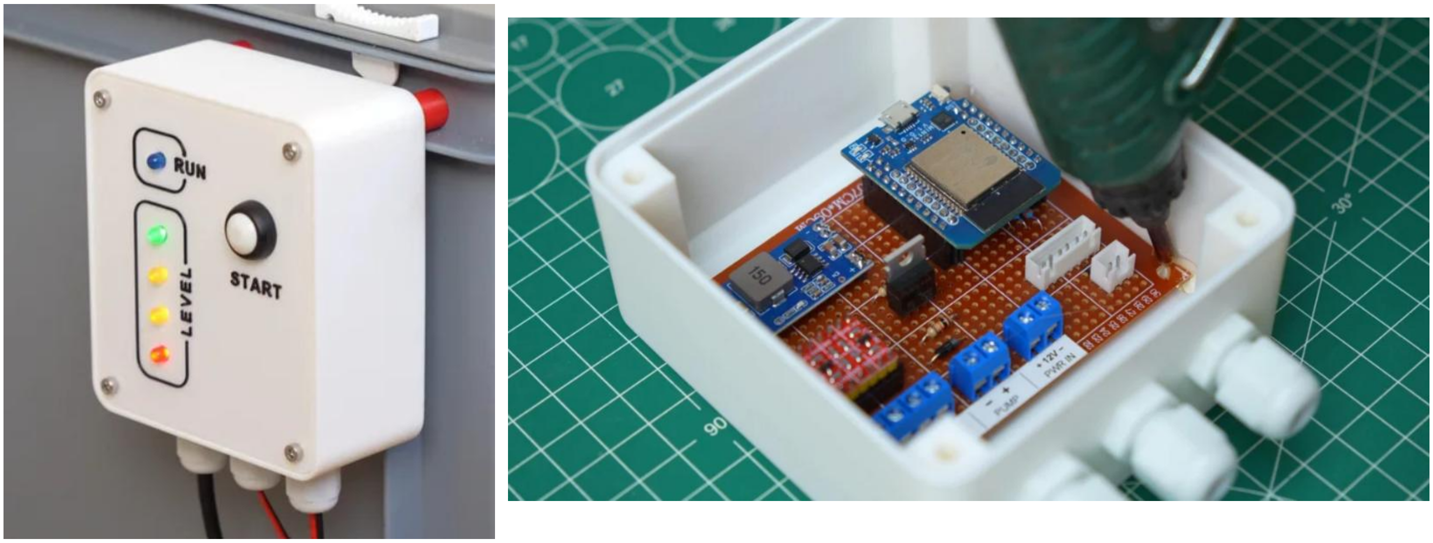

Figure 5 – Housing

The purpose of a mechanical enclosure is to provide support for the electronics component parts, such as the microcontroller, relay module, and power supply. This enhances protection of the system against moisture, physical damage, and environmental factors.

2.2 Electrical System Design

The electrical system design involved in the Smart Irrigation System is responsible for monitoring environmental factors, data processing, water flow regulation, as well as user interface communication. The electrical system design is focused on power efficiency, reliability, as well as integration with mechanical systems.

1. MCU - Microcontroller Unit

Figure 6 – MCU

It uses a microcontroller such as ESP32 -based boards to provide the system with a central processing unit. The microcontroller will receive data from sensors, perform the processing of control logics, and produce output signals to actuate the pump and valves. This allows for immediate automation and real-time decisions based on current soil and environmental conditions.

2- Soil Moisture Sensor

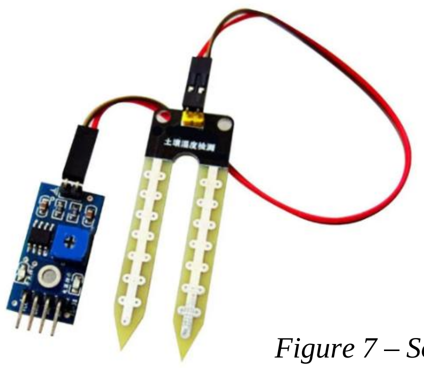

Figure 7 – Soil moisture

A soil moisture sensor is used to measure the amount of water in the soil. The sensor gives out signals that are proportional to the soil moisture and indicate to the system whenever irrigation is needed. The soil sensor plays an essential part in preventing the waste of water and optimizing irrigation.

3- Temperature and Humidity Sensor

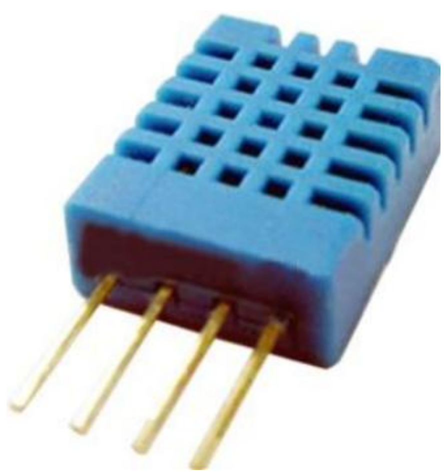

Figure 8 – Temperature & Humidity

It monitors the ambient environmental condition with temperature and humidity sensors-such as DHT-series sensors. These will be used to decide on the need of crops, assessing the environmental stress for better irrigation scheduling and control in crop management.

4– HCSR04 Ultrasonic Sensor

Figure 9 – HCSR04

The sensor sends out a sound wave at a specific frequency. It then listens for that specific sound wave to bounce off of an object and come back. The sensor keeps track of the time between sending the sound wave and the sound wave returning.

5- Yl-83 Rain Detector

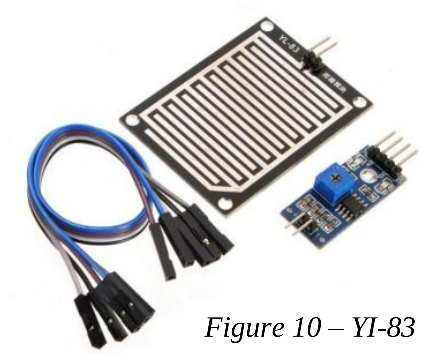

Figure 10 – YI-83

The rain sensor works by detecting water droplets on its conductive surface, which then changes its electric resistance or voltage levels.

6- Relay Module

Figure 11 – Relay

A relay module or motor driver circuit is used to provide electrical isolation and control of the high-power devices like the water pump and the solenoid valve. The low-voltage control signal from the microcontroller controls the relay, which in turn turns on/off the high-voltage circuit. The circuit works to provide a safe switch to the high-voltage circuit.

7- Battery

Figure 12 – Battery

A 12v battery system composed of 8AA batteries connected in series is used as the primary power source for our system. The main function for it is to supply stable DC power to all electrical components, supports off-grid and remote operation.

8- Jumper wire

Figure 13 – Wires

Jumper wires are used to establish temporary and flexible electrical connection between system components. Also, support rapid prototyping, reduce development time and complexity.

Sensor comparison and selection rationale:

1- ***Temperature and humidity sensor***

Selected sensor: DHT11

Function: measuring ambient temperature and relative humidity.

Why this sensor was selected:

- Low power consumption.
- Low cost and wide availability.
- Simple digital output.
- Easy to interface with ESP32.

Alternative sensors:

Sensor	Limitation
LM35	Measures temperature only no humidity
SHT31	Higher accuracy but significantly higher cost
DS18B20	Temperature only no humidity data

The DHT sensor provides both temperature and humidity measurements in a single module, reducing system complexity and cost while providing sufficient accuracy.

2- ***Resistive Rain sensor***

Selected sensor: ***YI-83***

Function: detect rainfall presence on a conductive surface.

Why this sensor was selected:

- Low power consumption.
- Fast response to rainfall.
- Simple working principle.
- Easy to interface with ESP32.

Alternative sensors:

Sensor	Limitation
Weather API-based rain data	Requires continuous internet access
Tripping bucket rain gauge	Complex mechanical structure and higher cost
Optical rain sensor	Expensive and overqualified for small systems

3- ***Ultrasonic sensor***

Selected sensor: ***HC-SR04***

Function: measure water level in the tank by calculating distance using ultrasonic waves.

Why this sensor was selected:

- Non-contact measurement
- High reliability for liquid level detection.
- Low cost and low power consumption.
- Easy implementation using digital signals.

Alternative sensors:

Sensor	Limitation
Float switch	Mechanical wear and limited accuracy
Pressure sensor	Requires direct contact with water
Capacitive level sensor	Higher cost and complex calibration

4- ***Resistive Soil moisture***

Selected sensor: ***SMR110***

Function: measure soil water content.

Why this sensor was selected:

- Simple design and easy calibration.
- Low cost and readily available.

Alternative sensors:

Sensor	Limitation
Capacitive soil moisture sensor	Higher cost
Tensiometer	Complex installation and maintenance
Neutron probe	Expensive and impractical for small systems.

2.3 Control System Design

The control system design is responsible for deciding how the Smart Irrigation System operates its functions based on inputs received from the sensors. The control system is the intellect of the Smart Irrigation System, which ensures that irrigation only takes place when needed and when the weather is adequate. The control system takes input readings from soil moisture, temperature, humidity, and rainfall sensors. All these readings are processed by the microcontroller according to control logic. Depending on the output of the microcontroller analysis, control signals are given to turn on/off the actuators, such as the water pump.

The main aim of the control system is the automation of irrigation, which will minimize human involvement and optimize water usage. This system will ensure efficient irrigation practices while preventing waterlogging or overwatering during rainfall or when the soil has adequate levels of water.

Control Strategy it begins by:

• If the soil moisture is depleted below certain levels, the irrigation subsystem will become operative.

• If there is rainfall indicated by the rain sensor, watering will be stopped irrespective of the level of soil moisture.

• Environmental factors like temperature and humidity are utilized to optimize times for irrigation.

•The process will automatically be stopped once the required level of soil moisture is achieved.

By this, accuracy in water supply is achieved, and sustainable agriculture is promoted.

Feedback Mechanism:

The control system is a closed-loop system. This means that the feedback from the sensors keeps the system state updated. The feedback from the soil moisture levels after the irrigation process enables the control system to realize whether the control goal has been met.

Manufacturing and Assembly

The design and assembly process for the Smart Irrigation System is based on combining mechanical, electric, and control components. The final product is intended to be functional and created with low-cost materials and components which are easily manufactured.

To reach the optimal design we need to choose everything carefully

1- Component Selection and Preparation

The mechanical and electrical components are chosen depending on their feasibility of availability, their suitability, and their specifications. In mechanical components, items like the water tank, pumps, tubing, and casing are chosen to be easily used and tested for fitment and leakage. In electrical components, items such as sensors, a microcontroller, a relay module, a battery, and cables are tested.

2- For the assembly:

The mechanical system begins with the water pump system inside the water tank. Tubing is done to link the pump outlet with the irrigation point. This helps avoid leakage. The sensors such as the soil sensor should be installed at a depth where it can take the correct readings. A housing system is designed to protect the electronics part away from water.

3- Electrical Assembly and Wiring

The electrical assembly entails the interfacing of sensors, the relay module, the pump, and the power source to the ESP32 through the use of wires. The polarity and insulation are carefully observed. The relay module serves to isolate the control circuit and the actuators. Cable organization methods are utilized to avoid loose connections.

4- Testing and Validation

The assembled system is tested for functionality. The testing includes soil moisture variation testing, rain sensor operation, and continuity testing. If there are any problems, they are eliminated by modifying wiring, sensors, or parameters. The testing makes sure that everything functions as expected and all design specifications are met. All of this data is sent to the user to be able to follow each percentage and change whenever he wants.

Chapter 3.

Conclusions and Future Directions

3.1 Conclusions

The project successfully illustrates the design and development of a Smart Irrigation System that incorporates the concepts of mechanical engineering, electrical engineering, and control engineering to successfully manage the usage of water. The proposed Smart Irrigation System employs the concept of various sensors such as soil moisture sensors, temperature sensors, humidity sensors, rain sensors, and ultrasonic sensors.

The mechanical design is done to ensure proper water storage, water distribution, and protection of the electronics. The electrical design is done to ensure the integration of low-power sensors, microcontrollers, power management, and communication for the autonomous functionality. The control system is a closed loop with real-time sensor inputs for the control of irrigation operations to avoid wasting water in case of rainfall or sufficient soil moisture.

On the whole, the Smart Irrigation System helps save water, minimize human interaction, and increase the productivity of agricultural crops. The proposed system design acts as a scalable and flexible system which can further be extended and improved by integrating advanced data analysis techniques, cloud-based systems, and AI methods in the future.

3.2 Future Work

The future extensions can involve the incorporation of a light intensity sensor (LDR) for tracking sunlight exposure, pH sensors for the evaluation of soil pH levels, and humidity sensors for tracking air humidity. Moreover, the AI vision system can also be integrated using a camera to track plants throughout the varied seasons. The purpose of such a system is to monitor the impact caused by the factors surrounding plant growth, such as sunlight, rainfall, cold, and wind. The AI system can then produce research reports for the user based on plant monitoring.

Future workers also can add:

1. Vision-based Plant Monitoring and Long-Term Decision Making: By integrating camera-based systems and artificial intelligence, it is possible to continuously monitor plant conditions (growth, stress, leaf color, and wilting) throughout the year with minimum human input and with more productive growth.
2. Predictive and Adaptive Control using AI/ML — Future systems will use predictive Machine and Deep Learning models (e.g., LSTM, ANN) to predict watering needs based on sensor history, weather forecasts, and plant growth stage rather than simple reactive threshold-based irrigation.
3. Hybrid Edge/Cloud Architectures for Urban Implementations — The combination of microcontrollers and cloud (dashboard, analytics) allows for effective real-time operation, remote access, and long-term data storage, particularly for balcony systems.
4. Low-Cost implementation, Scalability, and User-Friendliness — a lot of smart irrigation systems are still in the prototype stage. Systems that are low-maintenance, dependable, affordable, and appropriate for end users are one of the biggest objectives of the future.

Bibliography

1- A. J. Jha et al., "Systematic Review of Smart Irrigation Systems Based on IoT, ML, and DL," Agronomy, vol. 13, no. 2, pp. 342, MDPI, 2023. Available: https://www.mdpi.com/2073-4395/13/2/342

2- M. El-Sayed, "IoT-based Smart Irrigation for Mechanized Systems," International Journal of Innovation in Science and Development, 2022. Available: https://ijisd.journals.ekb.eg/article_252796.html

3- R. Kumar and A. Patel, "Sensor-Based Irrigation System: A Review," International Journal of Engineering Research & Technology (IJERT), 2016. Available: https://www.ijert.org/sensor-based-irrigation-system-a-review

4- S. P. Dey et al., "Vision-Based Irrigation Decision System using Raspberry Pi," Electronics, MDPI, 2022. Available: https://www.mdpi.com/2673-4052/5/2/7

5- M. Khalid et al., "AI and IoT in Precision Irrigation: Trends and Challenges," Turkish Journal of Agriculture - Food Science and Technology, 2024. Available: https://www.turjaf.com/index.php/TURSTEP/article/view/551

6- L. Navarro et al., "Edge-Cloud Hybrid Architectures for IoT Irrigation Systems," Open Ukrainian Citation Index (OUCI), 2024. Available: https://ouci.dntb.gov.ua/en/works/4baa8Lv7/

7- J. Ortega et al., "Predictive Irrigation System using LSTM and ESP32," arXiv preprint, 2025. Available: https://arxiv.org/abs/2506.11835

8- K. Ogata, Modern Control Engineering, 5th ed., Pearson, 2010. FAO, "Water Management and Smart Irrigation Systems." https://www.fao.org/land-water/water/en/

9- A. R. Al-Ali et al., "IoT-based Smart Irrigation Systems," IEEE, 2017 https://ieeexplore.ieee.org/document/8352558

10- MDPI Sensors, "Soil Moisture and Environmental Sensors." https://www.mdpi.com/1424-8220/19/9/2141

11- USDA, “Irrigation System Components.”
https://www.nrcs.usda.gov/resources/guides-and-instructions/irrigation-systems

12- https://www.electronicwings.com/sensors-modules/dht11

13- https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

14- https://www.datasheethub.com/rain-drops-detection-sensor-module/

Appendix A

Algorithm Design

```
// =====================================================
// SMART IRRIGATION – INSTANT PUMP, SLOW SERIAL
// =====================================================

#include <DHT.h>

// ------------------ PINS ------------------
#define DHTPIN 27
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define RAIN_ANALOG 35
#define SOIL_ANALOG 34

#define TRIG_PIN 5
#define ECHO_PIN 18

#define PUMP_PIN 26

// ------------------ RELAY SETTINGS ------------------
// If pump works backward, swap HIGH/LOW here
#define RELAY_ON_LEVEL  LOW
#define RELAY_OFF_LEVEL HIGH

// ------------------ TIMING ------------------
unsigned long lastSerialTime = 0;
```

```
const unsigned long serialDelay = 2000; // Update screen every 2 seconds

// ------------------ TANK VARS ------------------
float tankEmptyDistance = 0;
float tankFullDistance  = 0;
float tankHeightCM      = 0;
bool tankCalibrated     = false;

// ------------------ SOIL SETTINGS ------------------
const int soilDryRaw = 4095;
const int soilWetRaw = 1800;

// Tracks pump state for logic
bool isPumpOn = false;
String pumpReason = "System Start";

// =====================================================
float readUltrasonicCM() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH, 25000);
  if (duration == 0) return -1;
  return duration * 0.034 / 2;
}
```

```
// ====================================================
void calibrateTank() {
  Serial.println("=== TANK CALIBRATION ===");
  Serial.println("Leave tank EMPTY...");
  delay(5000);
  tankEmptyDistance = readUltrasonicCM();
  Serial.println("Fill tank FULL...");
  delay(5000);
  tankFullDistance = readUltrasonicCM();
  tankHeightCM = tankEmptyDistance - tankFullDistance;
  if (tankHeightCM <= 0) tankHeightCM = 10;
  Serial.print("Height: "); Serial.println(tankHeightCM);
  tankCalibrated = true;
  Serial.println("=== DONE ===\n");
}

// ====================================================
void setup() {
  Serial.begin(115200);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(PUMP_PIN, OUTPUT);

  digitalWrite(PUMP_PIN, RELAY_OFF_LEVEL); // Start OFF

  dht.begin();
  calibrateTank();
```

```
}


// =====================================================
void loop() {
  // ==================================================
  // PART 1: INSTANT PUMP LOGIC (Runs every millisecond)
  // ==================================================

  int soilRaw = analogRead(SOIL_ANALOG);

  // LOGIC:
  // > 3000: Turn ON Immediately
  // < 1800: Turn OFF Immediately

  if (soilRaw > 3000) {
    if (!isPumpOn) { // Only change if not already on
      isPumpOn = true;
      pumpReason = "Raw > 3000 (Dry)";
      digitalWrite(PUMP_PIN, RELAY_ON_LEVEL);
    }
  }
  else if (soilRaw < 1800) {
    if (isPumpOn) { // Only change if not already off
      isPumpOn = false;
      pumpReason = "Raw < 1800 (Wet)";
      digitalWrite(PUMP_PIN, RELAY_OFF_LEVEL);
    }
  }
```

```
// If between 1800 and 3000, do nothing (keep old state)


// ==================================================
// PART 2: SLOW SERIAL UPDATE (Runs every 2 seconds)
// ==================================================


if (millis() - lastSerialTime >= serialDelay) {
  lastSerialTime = millis();


  // Read other sensors just for display
  float temp = dht.readTemperature();
  float hum  = dht.readHumidity();


  float distance = readUltrasonicCM();
  float waterHeight = tankEmptyDistance - distance;
  waterHeight = constrain(waterHeight, 0, tankHeightCM);
  float tankPercent = (waterHeight / tankHeightCM) * 100.0;


  int rainVal = analogRead(RAIN_ANALOG);
  String rainStatus = (rainVal > 3800) ? "CLEAR" : (rainVal > 2500) ? "LIGHT RAIN" : "HEAVY RAIN";


  int soilPercent = map(soilRaw, soilDryRaw, soilWetRaw, 0, 100);
  soilPercent = constrain(soilPercent, 0, 100);


  Serial.println("================ STATUS ================");
  Serial.print("Soil Raw   : "); Serial.print(soilRaw);
  Serial.print(" ("); Serial.print(soilPercent); Serial.println("%)");
```

```
    Serial.print("Pump State : "); Serial.println(isPumpOn ? "ON (Watering)" : "OFF (Idle)");
    Serial.print("Last Action: "); Serial.println(pumpReason);

    Serial.print("Tank Level : "); Serial.print(tankPercent, 0); Serial.println("%");
    Serial.print("Rain       : "); Serial.println(rainStatus);
    Serial.print("Temp/Hum   : "); Serial.print(temp, 1); Serial.print("C / "); Serial.print(hum, 0); Serial.println("%");
    Serial.println("========================================\n");
  }
}
```

Appendix B