Industrial Internship Report on

”Arduino Based Smart Irrigation System”

Prepared by

Soham Bagde

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time. My project was Arduino based smart irrigation system. It uses a soil moisture sensor and DHT11 to monitor soil and weather conditions. When the soil is dry, the system automatically waters the plants. The aim is to save water, reduce manual effort, and improve farming efficiency This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

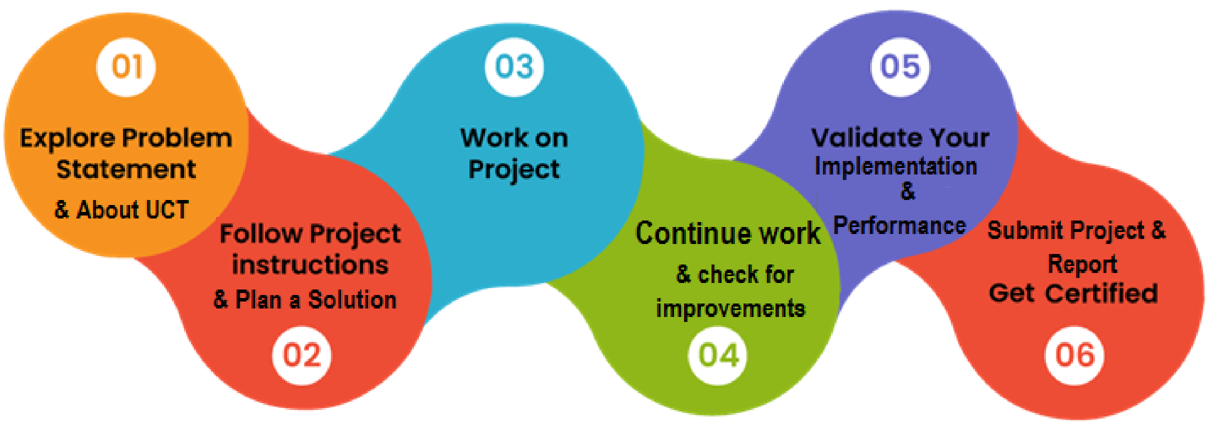
1 Preface

Over the 6-week **online internship** in Embedded Systems and IoT, I learned how to work with microcontrollers like Arduino and ESP32, along with various sensors. The program helped me apply theoretical concepts through hands-on virtual projects, including my main project—a **Smart Irrigation System** that automates watering based on soil moisture and climate data.

Such internships are important for career growth as they provide real-world experience, improve technical skills, and boost confidence in handling embedded systems.

My project addressed the need for efficient water use in agriculture by using sensors and Arduino to automate irrigation. It was both practical and impactful.

I'm grateful to **USC/UCT** for offering this valuable opportunity. The internship was well-planned, with weekly modules, video lessons, and assignments that gradually advanced from basic to complex topics.



During this 6-week online internship, I learned about sensors, Arduino programming, and real-time IoT applications. The experience was valuable and helped me build practical skills. I sincerely thank USC/UCT, and everyone who supported me. To juniors—grab such opportunities, stay curious, and keep learning.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

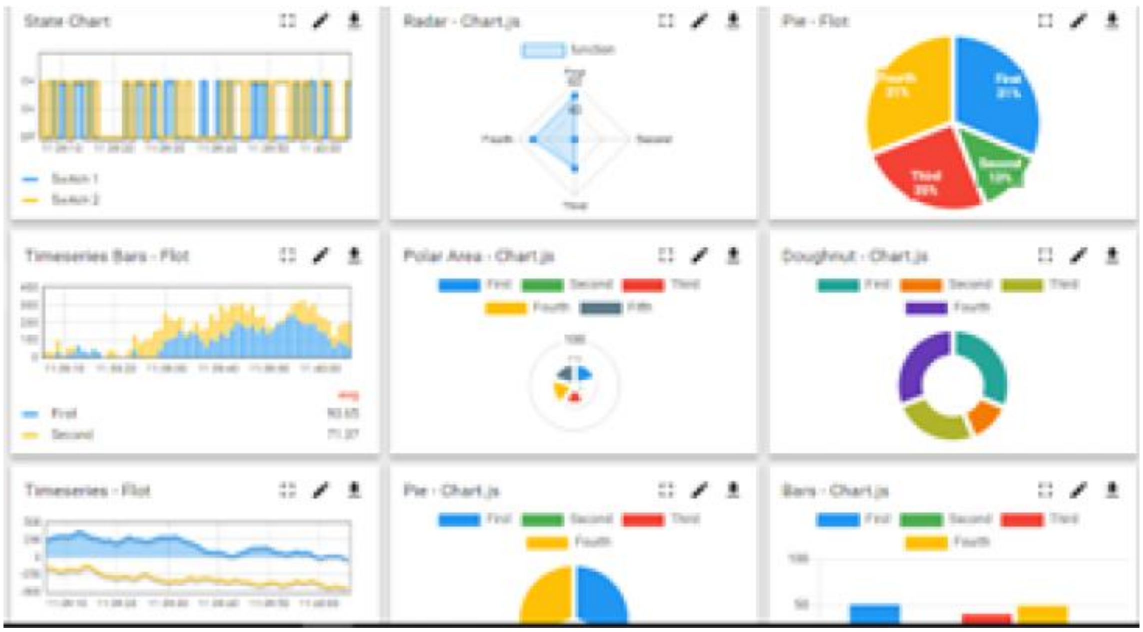
i. UCT IoT Platform (uct Insight)

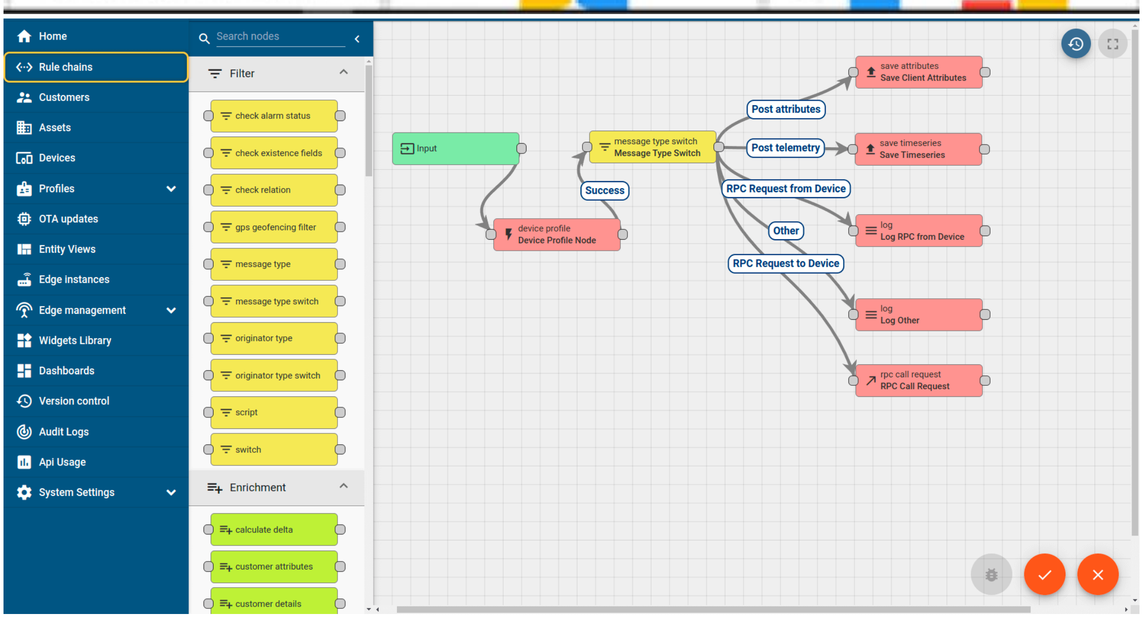
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

• Build Your own dashboard
• Analytics and Reporting
• Alert and Notification
• Integration with third party application(Power BI, SAP, ERP)
• Rule Engine





ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

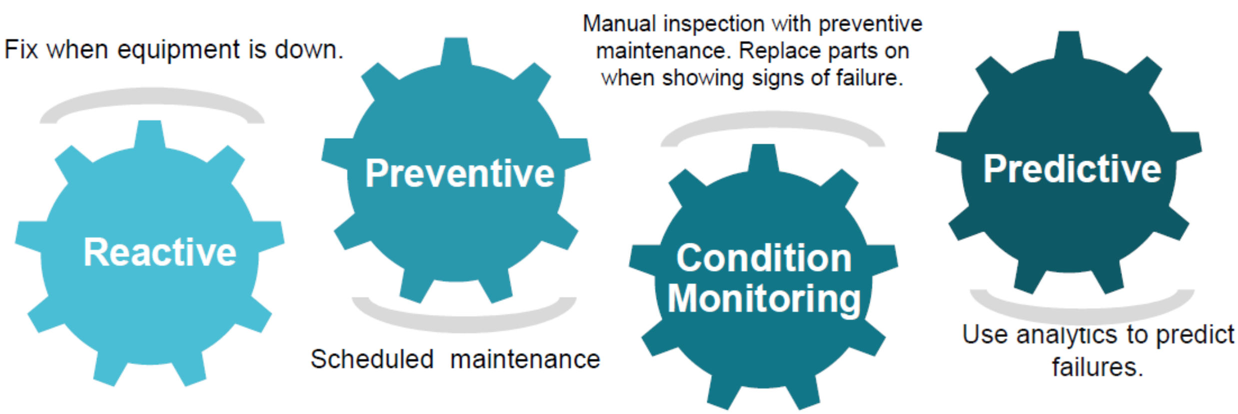
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN™ based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

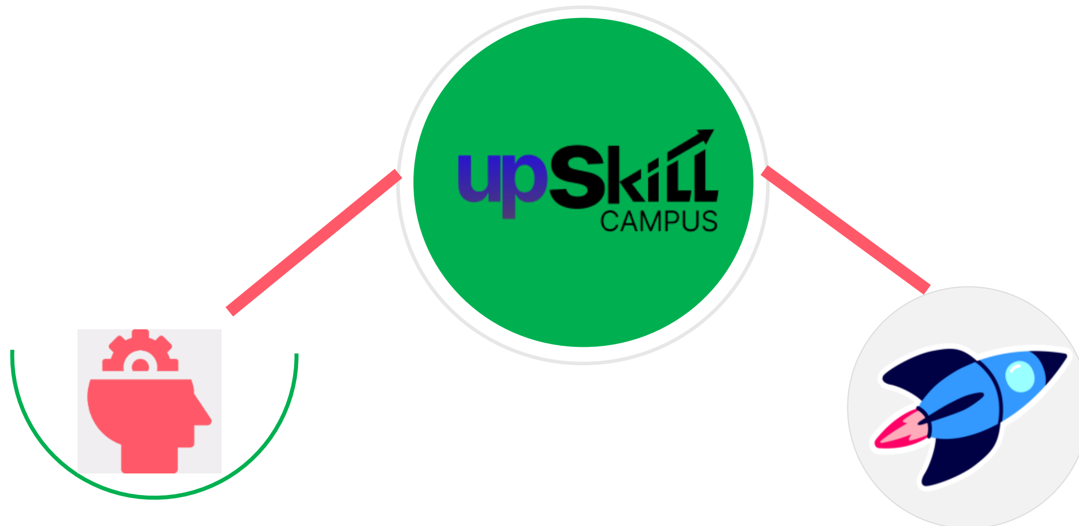
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

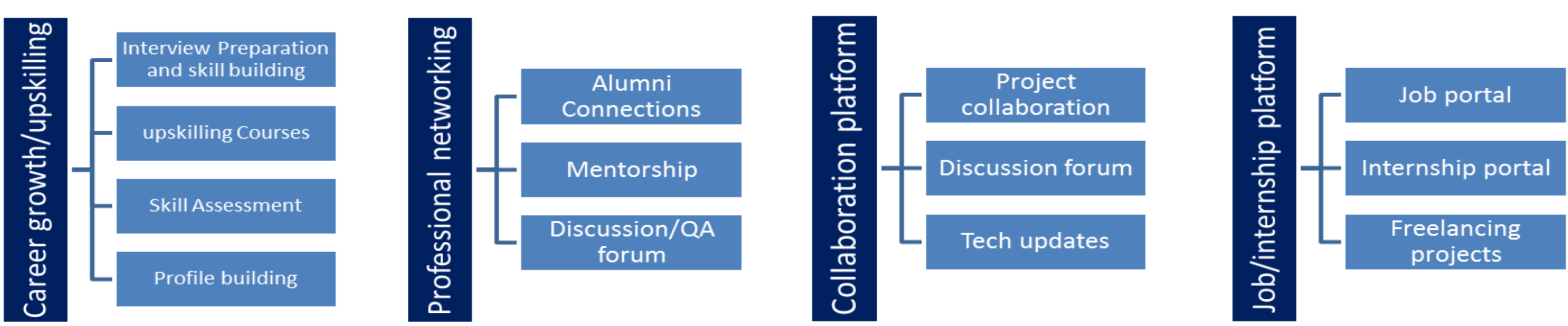
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] **Arduino Official Website** – https://www.arduino.cc

[2] **I2C LCD Display Guide** – https://randomnerdtutorials.com/arduino-i2c-lcd-guide/

[3] **Smart Irrigation System using Arduino** – International Journal of Engineering Research & Technology (IJERT), Volume 9, Issue 6, 2020.

2.6 Glossary

Terms	Acronym
Arduino	An open-source microcontroller platform used for building electronics projects.
DHT11	A sensor used to measure temperature and humidity.
I2C (Inter-Integrated Circuit)	A communication protocol that allows multiple devices to communicate using just two wires.
OLED (Organic Light Emitting Diode)	A type of display that emits light without a backlight, used in electronics projects for showing data like sensor readings. It is energy-efficient and offers high contrast and readability.
Soil Moisture Sensor	A device that measures the water content in soil.

3 Problem Statement

Traditional irrigation methods often result in excessive water usage and inefficient crop management due to the lack of real-time environmental monitoring. With increasing concerns over water scarcity and the need for sustainable agriculture, there is a critical need for an automated irrigation system that optimizes water usage based on real-time soil moisture and climate conditions. This project aims to design and implement a smart irrigation system using Arduino, sensors, and display modules to automate irrigation, reduce water wastage, and support efficient farming practices.

4 Existing and Proposed solution

Existing Solutions:

Many existing irrigation systems use timers or manual control to water crops, which can lead to over-irrigation or under-irrigation. Some automated systems rely on weather predictions or fixed schedules without sensing the actual soil moisture. Commercial smart irrigation systems are available but are often expensive and not suitable for small-scale farmers or students.

Limitations of Existing Systems:

- Lack of real-time monitoring of soil moisture.
- High cost and complexity.
- Energy inefficient or require constant human monitoring.
- Poor adaptability for small farms or gardens.

Proposed Solution:

Our proposed system is a low-cost, Arduino-based smart irrigation setup that uses a soil moisture sensor, DHT11 (for temperature and humidity), and an I2C LCD/OLED display. The system reads real-time data and activates irrigation only when necessary, conserving water and reducing manual effort.

4.1 **Code submission (Github link)**: https://github.com/soham-lgtm/upskillCampus/blob/main/SmartIrrigationSystem.ino

4.2 **Report submission (Github link)**: https://github.com/soham-lgtm/upskillCampus/blob/main/SmartIrrigationSystem_Soham_USC_UCT%20.pdf

5 Proposed Design/ Model

Proposed Design:
The smart irrigation system is built using an **Arduino Uno** microcontroller, which collects real-time data from various sensors to control the irrigation process automatically. The system uses a **soil moisture sensor** to detect water levels in the soil, a **DHT11 sensor** to monitor temperature and humidity, and displays information on a **16x2 I2C LCD or OLED**. When the soil moisture drops below a set threshold, the Arduino activates a **relay module** connected to a **water pump** to irrigate the field or plant area.

Key Components:

- **Arduino Uno** – Controls the system logic.
- **Soil Moisture Sensor** – Checks soil moisture level.
- **DHT11 Sensor** – Measures temperature and humidity.
- **16x2 I2C LCD / OLED Display** – Displays live sensor data.
- **Relay Module** – Switches the water pump ON/OFF.
- **Water Pump / Valve** – Delivers water to the field.

Working Model Description:

1. The soil moisture sensor checks the soil's water content.
2. If moisture is below the predefined level, Arduino turns the pump ON via the relay.
3. DHT11 readings are also taken for reference, and data is shown on the display.
4. Once moisture is sufficient, the pump is turned OFF automatic

5.1 High Level Diagram (if applicable)

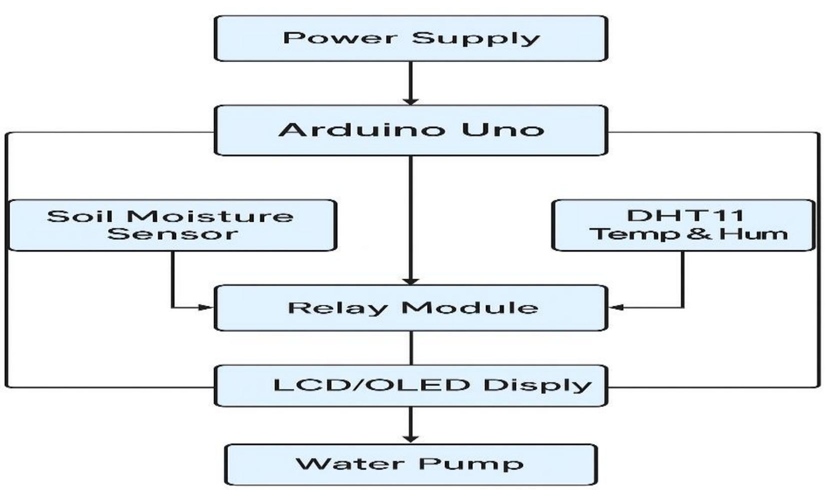
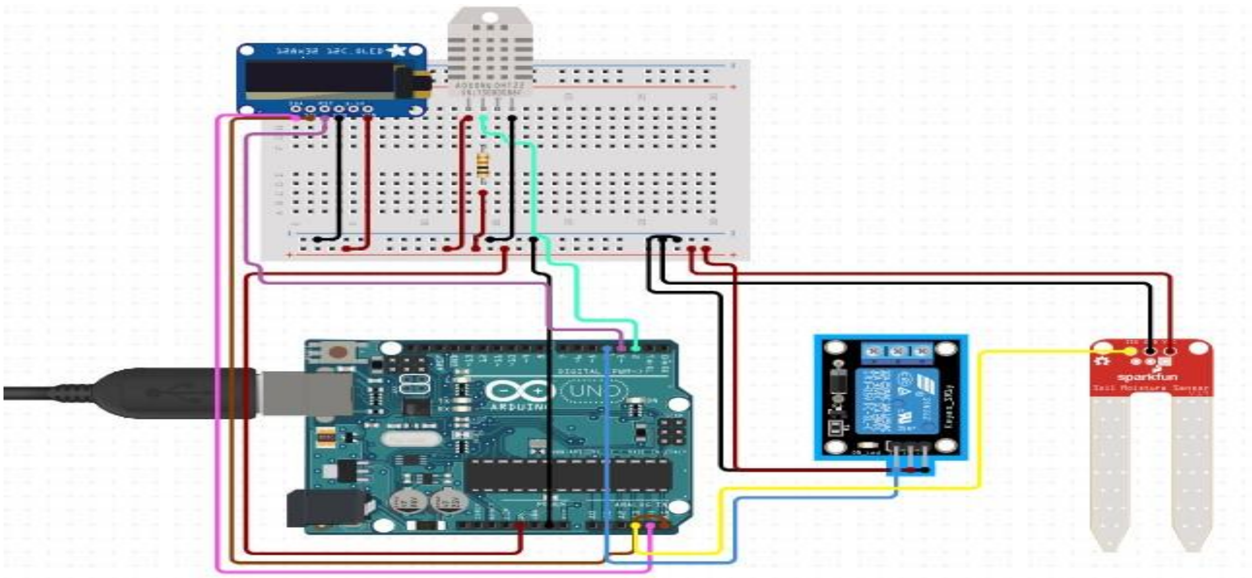


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Low Level Diagram (if applicable)



6 Performance Test

The smart irrigation system was tested under various conditions to assess its functionality, accuracy, and response time.

- The system **successfully activated the water pump** when the soil moisture dropped below the threshold and **turned it off** when adequate moisture was detected.
- The **DHT11 sensor provided accurate temperature and humidity readings**, which were correctly displayed on the OLED screen.
- The **relay responded quickly**, with a delay of less than one second, ensuring timely actuation of the pump.
- After a **power failure**, the system resumed operation effectively, reading fresh data from the sensors without any need for manual reset.
- However, under **direct sunlight**, the OLED display was **difficult to read**, which could be a limitation in outdoor use.
- Also, in case of a **sensor disconnection**, the system lacked fault detection and assumed zero moisture, causing the pump to turn ON unnecessarily.

Despite minor limitations, the system demonstrated stable and reliable performance. To improve robustness, future versions can include **sensor fault alerts**, **a more sunlight-readable display**, and **backup power options**.

5.3 Test Plan/ Test Cases

The smart irrigation system was evaluated through multiple test cases to ensure its functional correctness and reliability:

- **TC01:** When the soil moisture level dropped below the set threshold, the system correctly activated the pump, indicating successful moisture detection and response.
- **TC02:** When the soil was already moist, the pump remained OFF, proving that the system avoids unnecessary irrigation.
- **TC03:** The OLED display accurately showed real-time environmental data, including temperature, humidity, and soil moisture values.
- **TC04:** The relay module responded quickly—within approximately 0.5 seconds—when triggered by the Arduino, confirming timely activation of the pump.
- **TC05:** After a power failure and system reboot, the Arduino resumed normal operation without manual intervention, showing good recovery behavior.
- **TC06:** In the case of a disconnected soil moisture sensor, the system falsely assumed dry soil and turned the pump ON. This indicates a lack of fault detection, which may lead to over-irrigation.
- **TC07:** The OLED display was difficult to read under direct sunlight, suggesting a limitation in outdoor visibility.

5.4 Test Procedure

To validate the functionality and performance of the smart irrigation system, the following testing procedure was followed:

- **Step 1:** Power on the Arduino board and ensure all components are correctly connected, including the soil moisture sensor, DHT11 sensor, OLED display, and relay module.
- **Step 2:** Check the OLED display to confirm that temperature, humidity, and soil moisture values are being read and updated correctly.
- **Step 3:** Simulate **dry soil conditions** by removing the sensor from moist soil or inserting it into dry material. Observe if the **pump activates** automatically through the relay.
- **Step 4:** Simulate **wet soil conditions** by inserting the sensor into moist soil. Ensure the **pump remains OFF** and the system reflects correct readings.
- **Step 5:** Disconnect the soil moisture sensor and observe the system's behavior. Record if it falsely activates the pump or handles the disconnection.
- **Step 6:** Expose the OLED display to direct sunlight and assess its readability.
- **Step 7:** Turn OFF and then ON the power supply to test the system's recovery and automatic re-initialization.

5.5 Performance Outcome

The smart irrigation system performed effectively in controlled test conditions, demonstrating its ability to automate watering based on soil moisture levels.

- The **system consistently responded** to varying soil conditions by accurately controlling the water pump.
- **Sensor readings were stable and reliable**, and the OLED display correctly presented real-time environmental data.
- The **response time of the relay was quick**, ensuring timely irrigation without noticeable delays.
- The **system resumed normal operation after power interruption**, confirming its basic fault tolerance.
- However, the system **lacked fault detection** for sensor disconnection, leading to unnecessary irrigation in such cases.
- Also, **outdoor display readability** was limited due to OLED visibility issues under sunlight.

Overall, the system proved to be **functional, responsive, and efficient**, with a few areas identified for enhancement in future versions.

6 My Learnings

Through this project, I gained hands-on experience in working with Arduino and integrating multiple sensors, such as the DHT11 and soil moisture sensor, to build a functional automation system. I learned how to use an OLED display for real-time data visualization and how to control hardware components like a water pump using a relay module. The project improved my understanding of IoT concepts, sensor interfacing, circuit connections, and debugging common hardware and software issues. I also developed skills in testing, analyzing performance outcomes, and documenting technical work effectively. Overall, this project gave me a practical foundation in embedded systems and automation using microcontrollers.

6 Future work scope

The smart irrigation system has strong potential for future enhancements to improve efficiency, scalability, and real-world application:

- **Integration with IoT/cloud platforms** (like Blynk or Thingspeak) can enable remote monitoring, data logging, and control via smartphone apps.
- **Solar-powered operation** can make the system energy-efficient and suitable for remote or rural areas without reliable electricity.
- **Multiple sensor nodes** can be deployed across larger agricultural fields to monitor moisture at different points for more precise irrigation.
- **Weather prediction integration** using APIs can help optimize watering schedules by considering upcoming rain or temperature conditions.
- **Automated fault detection and alerts** for disconnected or faulty sensors can prevent false watering and system failure.
- **User-defined irrigation schedules** and customizable moisture thresholds can be added for greater flexibility and control.