VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590014.

Internship Report

On

"Smart Agriculture System Using AIOT"

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

By

Milan K P	**USN:1AP21IS019**
Venugopal	**USN:1AP21IS046**

Under the guidance of:

Akhil Sai

2023 - 2024

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

A P S COLLEGE OF ENGINEERING

Anantha Gnana Gangothri,

NH-209, Kanakapura Road, Somanahalli, Bengaluru-560116.

Project Completion Certificate

I, **Milan K P**(Roll No: 1AP21IS019), hereby declare that the material presented in the Project Report titled **"Smart Agriculture System using AIOT"** represents original work carried out by me in the **Department of information science** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: **Guide Name:**

Dr.Shivamurthaiah M Akhil Sai

Advisor's Signature **Guide Signature**

Project Completion Certificate

I, **Venugopal**(Roll No: 1AP21IS046), hereby declare that the material presented in the Project Report titled **"Smart Agriculture System using AIOT"** represents original work carried out by me in the **Department of information science** at the **APS college of Engineering, Bangalore** during the tenure
2 October, 2024 – 12, December, 2024.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: **Guide Name:**

Dr.Shivamurthaiah M Akhil Sai

Advisor's Signature **Guide Signature**

Evaluation Sheet

Title of the Project: Smart Agriculture System using AIOT

Name of the Students:1 Milan K P (1AP21IS019)

2 Venugopal(1AP21IS046)

External Supervisor:

____

Internal Supervisor:

____

____

Date:

Place:

TABLE OF CONTENTS

Chapter 1

ABSTRACT

The Smart Agriculture System will seamlessly integrate IoT components, automation, and real-time weather data to enhance irrigation efficiency. Powered by a Raspberry Pi, the system will manage a stepper motor to regulate irrigation based on soil moisture levels and weather forecasts. Soil moisture data will be captured through an MCP3008 analog-to-digital converter, while weather updates will be retrieved from the OpenWeatherMap API.

A Flask web application will act as the system's dashboard, providing a clear display of key metrics such as temperature, humidity, soil moisture levels, and irrigation status. The dashboard will also feature an upcoming irrigation schedule tailored to weather predictions. Users will receive real-time notifications via the Twilio API, ensuring they are informed about irrigation activities and system updates. The stepper motor will operate as needed, initiating irrigation when soil moisture is low and halting it once the soil reaches optimal hydration.

By combining IoT, machine learning, and real-time data analysis, this system offers an automated and efficient solution for farmers, promoting effective water management and improved crop productivity.

Chapter 2

INTRODUCTION

The Smart Agriculture System is an automated solution designed to optimize irrigation using Raspberry Pi. It incorporates a soil moisture sensor to monitor soil dryness and a temperature and humidity sensor to track environmental conditions. When the soil is dry, the system activates a stepper motor to start irrigation, and when the soil is sufficiently wet, it halts the motor to conserve water. The system analyzes soil conditions in real time, ensuring efficient water usage. Key data such as soil moisture levels, temperature, humidity, and irrigation status are displayed on a user-friendly web page, offering farmers a comprehensive view of their field conditions.

2.1 Objective

The Smart Agriculture System aims to automate irrigation management by utilizing soil moisture data to control a stepper motor, ensuring efficient water usage and activation only when necessary. It enables real-time monitoring of soil moisture, temperature, and humidity, providing up-to-date insights into field conditions. The system promotes resource conservation by halting irrigation when the soil is adequately hydrated and supports environmental adaptability by tracking temperature and humidity to assist farmers in making informed decisions. With a user-friendly web-based dashboard, it displays key metrics such as soil moisture levels, temperature, humidity, and irrigation status in an accessible format. Ultimately, the system enhances farming productivity by streamlining operations, conserving resources, and reducing manual effort.

2.2 Problem Statement

1. Water is often wasted in agriculture due to inefficient irrigation practices.
2. Traditional irrigation methods take a lot of time and effort, making them difficult to scale.
3. Farmers struggle to monitor soil moisture, temperature, and humidity in real time.
4. A better solution is needed to save water, reduce manual work, and improve farming efficiency.

Chapter 3

APPLICATIONS

- **Precision Irrigation:**
  Precision irrigation is a key feature of the system, where water is supplied to crops based on accurate soil moisture readings. Using sensors, the system detects when the soil becomes dry and activates the irrigation motor to hydrate the crops. Once the soil moisture level reaches an optimal point, irrigation is automatically stopped. This method minimizes water wastage and ensures crops receive the exact amount of water needed for healthy growth, avoiding overwatering or underwatering, which can negatively affect plant health.

- **Remote Monitoring:**
  The system offers a user-friendly web dashboard that displays real-time data on soil moisture, temperature, humidity, and irrigation status. Farmers can access this information remotely from their devices, enabling them to monitor field conditions without being physically present. This application is especially useful for large-scale farms or areas that are difficult to access regularly, providing convenience and enhancing decision-making with up-to-date insights.

- **Research and Education:**
  The Smart Agriculture System can be used in educational settings to demonstrate the principles of Internet of Things (IoT), automation, and sustainable farming practices. Students and researchers can interact with the system to understand how sensors work, how data is collected and analyzed, and how automated systems can be used to improve agricultural efficiency. It serves as a hands-on learning tool for those studying agriculture, environmental science, and technology.

- **Sustainable Farming:**
  By integrating automation and real-time monitoring, the system promotes the sustainable use of resources. It conserves water by ensuring irrigation occurs only when needed and reduces energy consumption through precise control of irrigation systems. This application aligns with the principles of sustainable farming, supporting long-term agricultural productivity while minimizing environmental impact, making it a valuable tool for eco-conscious farming practices.

- **Crop yield Management:**
  The Smart Agriculture System enhances growing conditions by providing precise data and automated controls, helping farmers optimize their farming strategies. By ensuring crops receive the right amount of water and are grown in the right environmental conditions, the system contributes to improved crop health and higher yields. This application is particularly beneficial in regions where inconsistent weather patterns or resource limitations affect agricultural productivity.

- **Labor Reduction:**
  Automating tasks such as irrigation monitoring and control significantly reduces the need for manual labor. Farmers no longer need to manually check soil conditions or manage irrigation schedules, as the system takes over these responsibilities. This allows farmers to focus on other critical tasks, improving overall efficiency and productivity. The reduction in manual effort also makes farming more accessible to individuals with limited labor resources.

Chapter 4

COMPONENTS

4.1 Raspberry pi 4 Model B

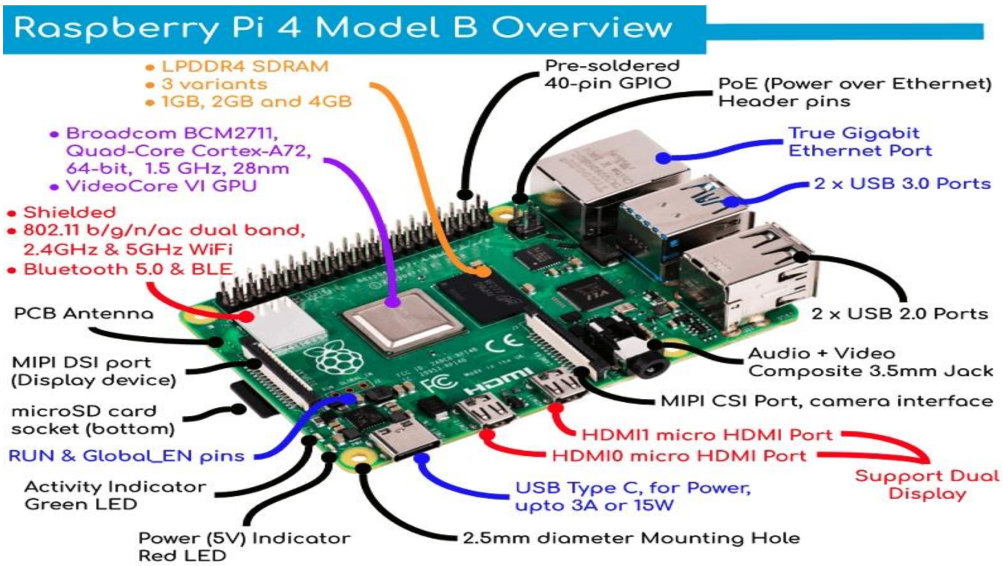

Figure 4.1: Raspberry pi 4 Model B.

The Raspberry Pi 4 Model B is a powerful single-board computer designed for a variety of applications, from education to industrial automation. It features a quad-core ARM Cortex-A72 processor, up to 8GB of RAM, dual micro-HDMI ports supporting 4K output, and USB 3.0 connectivity. Its GPIO pins enable easy interfacing with sensors, motors, and other peripherals, making it ideal for hardware projects. Built-in Wi-Fi, Bluetooth, and Ethernet provide versatile networking options. The Pi 4 is a versatile platform for learning, prototyping, and deploying IoT and embedded systems solutions.

4.2 Stepper Motor

Figure 4.2: Stepper Motor.

A stepper motor is a precision motor that moves in discrete steps, allowing for accurate control of position and speed. It operates by energizing coils in a specific sequence to produce controlled rotations, making it ideal for applications requiring precise movement, such as robotics, CNC machines, and 3D printers. Stepper motors are known for their reliability and ability to function without the need for feedback systems in many cases.

4.3 Soil Moisture Sensor

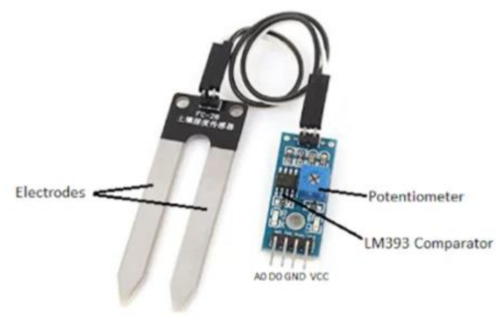

Figure 4.3: Soil Moisture Sensor.

The soil moisture sensor detects soil water content using two electrodes that measure conductivity, which increases with higher moisture. It includes an LM393 comparator and a potentiometer to set a moisture threshold. The sensor has four pins: VCC (power), GND (ground), DO (digital output for threshold detection), and AO (analog output for precise readings). It outputs a signal based on the soil's moisture level, making it ideal for smart agriculture, automated irrigation, and garden monitoring systems.

4.4 Temperature and Humidity Sensor

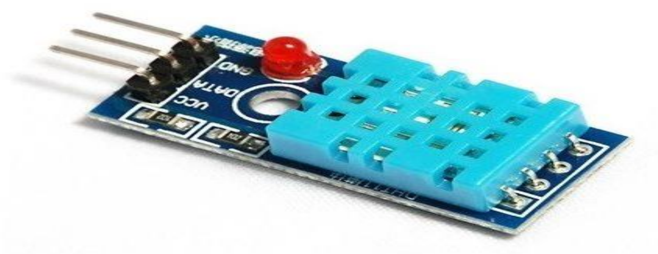

Figure 4.4: Temperature and Humidity Sensor.

The DHT11 is a digital sensor used to measure temperature (0°C to 50°C) and humidity (20% to 90%) with moderate accuracy. It operates on 3.3V to 5V and communicates via a single data pin, making it easy to interface with microcontrollers like Arduino and Raspberry Pi. Compact and affordable, it is ideal for applications like home automation, weather monitoring, and educational projects.

Chapter 5

FLOW CHART

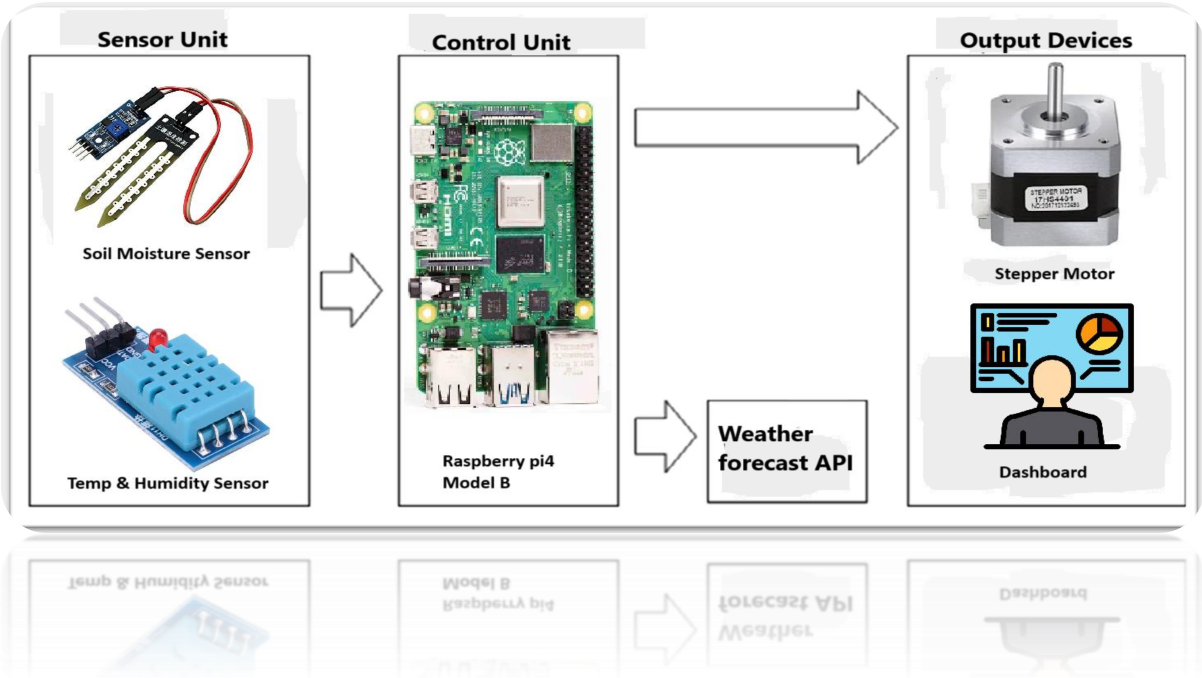

Figure 5.1: Flowchart

The brief explanation of the components in the figure 5.1 which is the flowchart of the project is as follows

Sensor Unit:

- The **Soil Moisture Sensor** detects the moisture level in the soil.
- The **Temperature and Humidity Sensor** monitors the environmental conditions, such as air temperature and humidity, which influence the irrigation process.

Control Unit:

- The **Raspberry Pi 4** acts as the central control unit. It receives data from the sensors (soil moisture, temperature, and humidity).
- Weather API: Provides external weather data for decisions.

Output Devices:

- Stepper Motor: Controls irrigation/mechanical tasks.
- Dashboard: Displays weather and system status.

Chapter 6

CONCLUSION

In conclusion, the Smart Agriculture System utilizing Raspberry Pi, sensors, and weather data integration provides an efficient, automated approach to managing irrigation. By continuously monitoring soil moisture, temperature, and humidity, the system ensures that crops receive the right amount of water, reducing water wastage and improving crop yield. The use of the Weather API allows for better planning based on real-time forecasts, further enhancing water conservation and resource management.The system's ability to display key metrics on a dashboard and sends alerts keeps farmers informed ,making it an effective tool for sustainable farming and optimized agricultural practices.

Chapter 7

FUTURE WORK

- **Machine Learning Integration**: Use predictive algorithms for better irrigation scheduling and crop management based on historical and real-time data.
- **Drone Integration**: Employ drones for real-time field monitoring, improving irrigation precision and crop health tracking.
- **Mobile Application:** Develop a user-friendly mobile app for remote access and control.
- **Energy Efficiency**: Incorporate renewable energy sources like solar panels to reduce energy consumption and make the system more sustainable.
- **Pest and Disease Detection**: Add IoT-based sensors or image recognition to detect pests and diseases, enabling timely interventions.

Chapter 8

APPENDIX

8.1 Pseudo Code

Initialization

1. **Setup GPIO**
    - Configure GPIO pins for LCD, stepper motor, and sensors.
    - Initialize LCD for display.
    - Setup stepper motor pins and define step sequence.
2. **Setup ADC (MCP3008)**
    - Initialize SPI for ADC communication to read soil moisture levels.
3. **Initialize Flask and WebSocket**
    - Create Flask app and enable WebSocket for real-time communication.

Functions

1. **LCD Functions**
    - Initialize LCD and display messages.
    - Toggle enable pin for LCD commands and data.
2. **Stepper Motor Functions**
    - Define motor step sequence and GPIO pin setup.
    - Rotate motor based on steps and direction.
3. **Sensor Reading Functions**
    - **Soil Moisture**: Read analog data from ADC.
    - **Temperature & Humidity**: Read data from the DHT sensor.

Flask Application

1. **Route**: Serve HTML page for the user interface.
2. **WebSocket Emission**: Emit sensor data (soil moisture, temperature, humidity, and status) to the web interface every 5 seconds.

Background Thread

1. Continuously perform:
    - Read soil moisture, temperature, and humidity.
    - Determine system status (Dry/Normal).
    - Emit real-time sensor data to WebSocket.
    - Activate stepper motor if soil moisture exceeds the threshold.

Main Server

1. Setup GPIO and initialize components.
2. Start the Flask server with WebSocket.
3. Handle cleanup (GPIO reset, close SPI) on exit.

Execution

- Start the Flask server when the script is executed.