BACHELOR OF INFORMATION TECHNOLOGY (HONS) IN INTERNET OF THINGS MARCH 2025

SMART PLANT MONITORING SYSTEM FOR CUCUMBER

AFIQ HAZIM BIN AZADDIN

52224122132

MALAYSIAN INSTITUTE OF INFORMATION TECHNOLOGY

UNIVERSITI KUALA LUMPUR

MARCH 2025

SMART PLANT MONITORING SYSTEM FOR CUCUMBER

AFIQ HAZIM BIN AZADDIN

52224122132

REPORT SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS

FOR THE BACHELOR OF INFORMATION TECHNOLOGY (HONS) IN INTERNET OF THINGS

MALAYSIAN INSTITUTE OF INFORMATION TECHNOLOGY,

UNIVERSITI KUALA LUMPUR

MARCH 2025

DECLARATION

I declare that this report entitles “Smart Plant Monitoring System for Cucumber” is my original work and all references have been cited adequately as required by the University.

afiqhazim

Date: Signature: ……………………………………

Full Name: Afiq Hazim Bin Azaddin

ID No: 52224122132

APPROVAL PAGE

I have examined this dissertation and verify that it meets the program and institute's requirements for Bachelor of Information Technology (Hons) in Internet of Things.

Date: 16/6/2025

Signature: ……………………………………………

Supervisor: Ts. Faridah Binti Yahya

Official Stamp: UniKL MIIT

ACKNOWLEDGEMENT

I would like to express my heartfelt appreciation to everyone who supported and guided me throughout the course of this project. The successful completion of this report would not have been possible without the encouragement, guidance, and assistance of numerous individuals. First and foremost, I would like to sincerely thank Ts. Faridah Binti Yahya, my project supervisor, for her constant guidance, encouragement, and constructive feedback to my Final Year Project. Her deep knowledge, technical expertise, and patient mentorship have been invaluable in shaping this project and helping me overcome challenges at various stages. Her insightful suggestions and attention to detail significantly enhanced the quality and outcome of this project. I am also grateful to Mr Mohd Sallehin Bin Mohd Kassim as my co-supervisor for his additional support and valuable inputs, which helped broaden the scope of this project. I would like to extend my gratitude to Mr Azmi Ahmad for providing the necessary project guidelines and coordinating the final year project session with success. I also appreciate the contributions of my classmates, whose collaboration and exchange of ideas fostered a productive working environment. Their encouragement and assistance, both academically and personally, helped me stay motivated and focused throughout the journey. Furthermore, I am deeply thankful to my family for their unconditional love, patience, and emotional support. Their belief in me has been a continuous source of strength and inspiration. Finally, I acknowledge the role of various online and offline platforms, tutorials, and open-source communities such as GitHub, that provided valuable information and tools used during the project.

ABSTRACT

This project presents the development of a Smart Plant Monitoring System specifically designed for the efficient and sustainable cultivation of cucumber plants. Cucumber growth is highly sensitive to environmental changes, making continuous monitoring essential for maintaining healthy conditions. To achieve this, the system uses several sensors to track key parameters such as temperature, humidity, soil moisture, and electrical conductivity (EC) of the soil. In addition, a PIR motion sensor is included for detecting movement around the plant area, which can help in identifying the presence of animals or intruders, and an ultrasonic sensor is used to monitor the water level in the irrigation tank or reservoir. The system is powered by an ESP32 microcontroller, which gathers real-time data from the connected sensors and sends it to the Blynk IoT platform, allowing users to monitor plant conditions remotely via a smartphone. When any sensor reading goes beyond the desired range, the system sends instant notifications, enabling quick action. Furthermore, all sensor data is automatically logged to Google Sheets through HTTP requests for easy tracking and analysis over time. This smart monitoring setup not only reduces the need for constant manual supervision but also helps ensure that cucumbers are grown in the best possible conditions. It supports better decision-making related to irrigation and fertilization, enhances security through motion detection, and promotes efficient resource management, hence contributing to healthier plants.

CONTENTS

DECLARATION .... **ii**

APPROVAL PAGE .... **iii**

ACKNOWLEDGEMENT .... **iv**

ABSTRACT .... **v**

LIST OF TABLES .... **viii**

LIST OF FIGURES .... **viii**

LIST OF ABBREVIATIONS .... **x**

CHAPTER 1: INTRODUCTION **1**

1.1 Introduction 1

1.2 Project Theme 2

1.2.1 DHT22 Temperature & Humidity sensor 2

1.2.2 Soil moisture sensor 2

1.2.3 PIR Motion Sensor 2

1.2.4 Ultrasonic Sensor 2

1.2.5 Actuators 3

1.2.6 TDS Sensor 3

1.3 Problem Statement 3

1.4 Project Objectives 4

1.5 Project Scope 4

1.6 Project Limitation 5

CHAPTER 2: LITERATURE REVIEW **6**

2.1 Introduction 6

2.2 Original Issue 7

2.3 Project Design 7

2.4 Review on Existing Systems 8

2.4.1 Smart Plant Monitoring System using Aquaponics Production Technological with Arduino Development Environment (IDE) and SMS Alert. 9

2.4.2 The Prototype of the Greenhouse Smart Control and Monitoring System in Hydroponic Plants. 10

2.4.3 Automatic Plants Watering System for Small Garden. 12

2.5 Literature Review Matrix 14

CHAPTER 3: METHODOLOGY **16**

3.1 Introduction 16

3.2 Research Methodology Structure 16

3.2.1 Project Planning Phase 17
3.2.2 System Design Phase 17
3.2.3 System Development Phase 17
3.2.4 Integration and Testing Phase 18
3.2.5 Deployment Phase 18
3.2.6 Maintenance Phase 18
3.3 System Design & Architecture 19
3.4 Project Costing 22
3.5 Gantt Chart 23
3.6 System Requirement Analysis 23
3.6.1 Hardware Requirement 24
3.6.2 Software Requirement 29
CHAPTER 4: RESULTS AND DISCUSSION 32
4.1 Introduction 32
4.2 Equipment Installation 32
4.2.1 Hardware Installation 32
4.3 Project Testing and Result 34
4.3.1 Hardware Testing 35
4.3.2 Integration Testing 36
4.3.3 System and Functional Testing 41
4.3.4 Notification Testing 45
4.4 Discussion 47
CHAPTER 5: CONCLUSION 48
5.1 Introduction 48
5.2 Project Review 48
5.3 Future Improvements 49
REFERENCE 50
APPENDIX A: WORK BREAKDOWN STRUCTURE 52
APPENDIX B: GANTT CHART 54
APPENDIX C: PLAGIARISM 57
APPENDIX D: ESP32 CODE 59
APPENDIX E: FINAL YEAR PROJECT POSTER 76
APPENDIX F: TECHNICAL PAPER 78

LIST OF TABLES

Table 2.1 Literature Review Matrix....14

Table 3.1 Project Cost....22

Table 4.1 ESP32 Pins Breakdown....33

Table 4.2 Objectives Outcome of the project....34

Table 4.3 Hardware Testing Results....35

LIST OF FIGURES

Figure 2.1 Block diagram for system....7

Figure 2.2 Block diagram Literature Review 1....9

Figure 2.3 Greenhouse System design....11

Figure 2.4 Schematic of a humidification control system....12

Figure 2.5 Schematic of a hydroponic monitoring system....12

Figure 2.6 Monitoring and control system architecture....13

Figure 3.1 Waterfall Method....16

Figure 3.2 Flowchart of the system....19

Figure 3.3 Block Diagram of the System....21

Figure 3.4 Circuit Diagram of the System....22

Figure 3.5 ESP32 Dev Kit (Xukyo, 2024)....24

Figure 3.6 Soil Moisture Sensor (Yu et al., 2021)....25

Figure 3.7 DHT22 Sensor (Syahputra Novelan, 2020)....25

Figure 3.8 Ultrasonic Sensor (Zhang et al., 2023)....26

Figure 3.9 PIR Motion Sensor (Ada, 2023) 26
Figure 3.10 TDS sensor (Ada, 2023) 27
Figure 3.11 Relay (ANONIM, 2020) 27
Figure 3.12 water pump (Prof. Praveen Rathod et al., 2023) 28
Figure 3.13 Buzzer (Malvinos, 2006) 28
Figure 3.14 Arduino IDE Environment 29
Figure 3.15 Blynk Dashboard 30
Figure 3.16 Data Logging using Google Sheets 31
Figure 4.1 Hardware Connections 33
Figure 4.2 ESP32 Integration with Sensors & Expected Result 37
Figure 4.3 Integration Testing Diagram 38
Figure 4.4 Blynk Dashboard Integration 39
Figure 4.5 Google Sheets Integration 39
Figure 4.6 Blynk Login 41
Figure 4.7 Devices Page 42
Figure 4.8 Blynk Dashboard 43
Figure 4.9 Google Sheets Access 44
Figure 4.10 Data Logging in Google Sheets 44
Figure 4.11 Events & Notifications Setting 45
Figure 4.12 ”Intrusion Detected” Configuration 45
Figure 4.13 Testing Email Notification 46

LIST OF ABBREVIATIONS

Abbreviation	Descriptions
EC	Electronic Conductivity
TDS	Total Dissolved Solids
IDE	Integrated Development Environment
COM	Common Terminal
NO	Normally Open
IoT	Internet Of Things
PIR	Passive Infrared
SMS	Short Message Service
EMAIL	Electronic Mail
LED	Light Emitting Diode
GSM	Global System Mobile Communications
pH	Potential Hydrogen
MaH	milliampere-hour
Us/cm	microsiemens per centimeter
Ppm	Parts Per Million
DHT	Digital temperature and humidity

CHAPTER 1: INTRODUCTION

1.1 Introduction

With increasing consumer demands, agriculture has therefore focused its purpose on increasing the outputs of its crops while relying on fewer resources. Cucumber, one of the widely grown vegetable crops, requires specific conditions to grow well. Most small farmers employing irrigation techniques still observe watering management manually along with fertilizing practices leading to loss of resources and increasing chances of poor plant performance(Kirci et al., 2022).

Smart agriculture has been proposed to promote the modernization of agriculture for increasing food production greatly. Specially, smart agriculture systems introduce many advanced computer and information technologies such as Internet of Things(Pathmudi et al., 2023). To solve such problems, this project proposes the development of an IoT enabled Smart Plant Monitoring System for cucumber to enhance the automation and efficiency of cultivation processes.

. The Smart Plant Monitoring System for Cucumber utilizes modern technologies such as IoT (Internet of Things) and sensor-based automation to revolutionize traditional farming practices. By integrating sensors, wireless connectivity, and data processing, this project is made to both novice gardeners and farmer enthusiasts aiming for precision in plant care (Kirci et al., 2022). The ESP32 microcontroller, serves as the brain of this technology. Paired with an array of sensors measuring crucial parameters such as soil moisture sensor, DHT11 humidity & temperature sensor, PIR sensor and rain sensor, this system not only captures real-time data but also interprets it into actionable insights. Furthermore, a smart plant monitoring system is not complete if a user couldn't control a water pump from their smart devices. Hence, this project also concludes a function where the user could control the water pump automation with their smartphones or computers using IOT platform (Pereira et al., 2023).

The adaptability of the ESP32 platform promotes endless customization possibilities. In summary, smart planting is vital for modernizing agriculture, making it more sustainable, efficient, and resilient (Abouelmehdi et al., 2022). The integration of technology in planting processes has the potential to revolutionize traditional farming practices and address the challenges posed by a growing global population.

1.2 Project Theme

The IoT-based Smart Plant Monitoring System for cucumber is designed to monitor and maintain the growth and health of cucumbers. The system works by collecting data from various sensors and then sending that data to a mobile application through the internet. The initial sensors that will be used includes:

1.2.1 DHT22 Temperature & Humidity sensor

Temperature and humidity are important variables to determine changes in the state of a substance or object. By knowing the change in temperature, we can know the direction of the climate change (Syahputra Novelan, 2020).

1.2.2 Soil moisture sensor

Soil moisture sensors measure or estimate the amount of water in the soil. These sensors can be stationary or portable such as handheld probes.(Yu et al., 2021).

1.2.3 PIR Motion Sensor

PIR Motion Sensor is a small, energy efficient, low cost, and accurate security management system that uses microcontroller-based passive infrared sensor module to generate an alert to the farm owner if there is an intrusion event at the crop field store. (Yu et al., 2021).

1.2.4 Ultrasonic Sensor

An ultrasonic sensor is used to measure the water level in a container in real time. In this project, the water level of a container filled with a fertilizer-water mixture needs to be monitored to keep track of the remaining liquid. The sensor operates by emitting ultrasonic pulses that reflect off the water surface; the time taken for the echo to return is used to calculate the distance to the water.

1.2.5 Actuators

Actuators that will be used in this project are 12v water pump connected to a relay. It is used to control water outage from water tank to the plant.

1.2.6 TDS Sensor

TDS (Total Dissolved Solids) Sensor is utilized to take advantage of its ability to measure water quality and mixture from other sources in a water. This sensor is implemented in my project to give reference on the optimum quantity for the cucumber plant.

1.3 Problem Statement

Growing crops in an acute manner especially delicate ones such as cucumbers, depend on workers who are physically present, which is a lot more work. The need to conserve resources inevitably leads to unattended rise or drop in watering and fertilizing as well as generational crop underperformance.

The rise in water scarcity is one of the crucial problems for farmers. They are strained to use traditional tools in making the decision on the water to be used for conserving them on resources whilst boosting health and yield of the crops.

In security manners, there is no way farmers could identify if their plants are well protected from pests or outsiders.

Other than that, farmers are forced to manually check their plants every single day which consumes a substantial amount of energy and effort to do it. Without any technology or intervention of internet of things, they will have to spend some time to check the conditions of the plant.

1.4 Project Objectives

1. To implement PIR motion sensor configuration to turn on buzzer and send notification to the dashboard and app if it detects any movements or intruders.
2. To develop irrigation control systems remotely, allowing for efficient management from anywhere which solves water scarcity.
3. To gather data from the environment sensors. The data is sent to an IoT platform where it can be monitored in real time through a mobile app or dashboard.

1.5 Project Scope

The soil moisture sensor served as an apparatus to measure water content in soil. DHT22 connectivity enables the system to detect temperature and humidity all at once. Nevertheless, this system implements the automated irrigation process to control the water pump through the dashboard or the app which enables it to turn off or on, giving system versatility.

The PIR motion sensor is integrated with the buzzer that serves as an alarm to warn nearby pests or intruders around it. The PIR motion sensor then sends notification through the IOT platform for any alerts. The IOT platform that is implemented is the Blynk platform. The system uses network architecture (Wi-Fi) to connect ESP32 to Blynk for real-time remote monitoring and notification alert to the user.

To monitor the fertilizer-water solution in the container, the system utilizes an ultrasonic sensor to estimate the remaining liquid volume. Additionally, a Total Dissolved Solids (TDS) sensor is employed to measure the concentration of fertilizer dissolved in the water. The TDS value is displayed in microsiemens per centimeter (µS/cm) on the dashboard, allowing for precise control and preventing both over-fertilization and under-application.

1.6 Project Limitation

a. Costs Associated with IoT Equipment and Data Services.

The cost of sensors, devices, and other equipment may be varied, and there may be additional costs associated with data storage and analysis.

b. Reliability of Internet for IoT Functionality

Obligated to reliable internet connection. IoT devices and sensors rely on stable and fast internet connection to transmit data.

c. Scalability Limitations Due to ESP32 Dependency

This project has a very limited scalability because only the ESP32 is used to transmit data which makes the project less suitable for further areas outside the Wi-Fi connection range.

d. Lack of automated fertilizer mixture capability

The system relies on manual mixing of fertilizer into the water tank as it provides no automation to mix them automatically into the water. The quantity of us/cm is provided in the dashboard for reference.

e. Detection Limitations of PIR Motion Sensors

PIR motion sensor detection is limited as the sensor only detect anything on its cone of vision.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter will outline the literature reviews as they relate to the study topic and objectives. Reading books, online journals, papers, and other electronic sources is used to produce the literature review. All secondary data collected is based to the research topic. A review of the literature might also serve as a guideline for the subsequent chapter.

This project can bring benefits to small farmers. These new conditions are easy to use to help users during the decision-making process. This case study describes how plant monitoring can be improved using Blynk software, to take control of the plant environment and optimize its management. According to *(Tech Explorations, 2022),* with Blynk, we can create smartphone applications and IOT dashboard that allow us to easily interact with microcontrollers or even full computers such as the Raspberry Pi. The focus of the Blynk platform is to develop the mobile phone application with IOT dashboard. With Blynk, we can use it to monitor the soil humidity of our vegetable garden and turn on the water, with our phone and computer. By using Blynk with smart plant monitoring, it aims to create a system that can remotely monitor plants' health and provide water to the plant moisture level is low ("SMART AUTOMATED PLANT MONITORING SYSTEM USING BLYNK APPLICATION," 2023).

Based on article (Mohanty & Roy, 2021),This smart gardening system using Blynk will provide convenience and comfort to the user by sensing and controlling the parameters of the plants without their physical presence. Arduino IDE is used for compiling and uploading the program to ESP32 and Blynk IoT platform is used for displaying of temperature, humidity, soil moisture and rain conditions that can be accessed from any distance. In terms of security, this system also integrates the PIR motion sensor to turn on the buzzer if it detects any sudden changes of movement around the plant's area. It will also send a notification as an alert for the user on the dashboard.

2.2 Original Issue

Traditional plant monitoring frequently lacks the technical advances found in other areas of agriculture. Referring to (Mohabuth & Nem, 2023),basic regular checking is sufficient but do not handle key issues like temperature and humidity data, soil moisture content, and irrigation efficiency. Farmers would manually have to check plant health, soil moisture, and environmental conditions, which can be infrequent and inconsistent.

2.3 Project Design

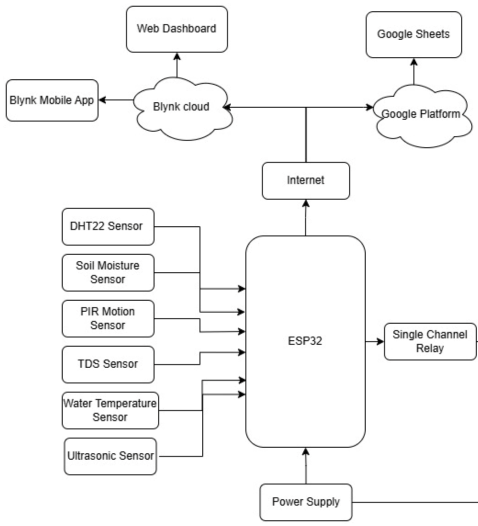


Figure 2.1 Block diagram for system

The smart plant monitoring system for cucumber project focuses on how it can monitor plant environments for healthier growth. We can also take further actions if any situation is out of our control before it worsens. Below is the block diagram for this project that I am working on.

a. Hardware Breakdown

Based on Figure 2.1 above, sensors inclusive soil moisture sensor, DHT11 sensor and PIR motion sensor are connected through ESP32 microcontroller input. From the ESP32 microcontroller, the data gathered from the sensors are sent in the IoT platform which is BLYNK through Wi-Fi connection. Then, the dashboard displays the data from the sensors. A buzzer will turn on when the PIR motion sensor detects any movement in its vision. In addition, a notification will also be sent to the dashboard. The water pump can also be controlled manually through the dashboard, activating the relay to turn water pump on or off.

2.4 Review on Existing Systems

The vital task in this phase is to identify the similarities, differences, advantages, and disadvantages of the existing system. Smart plant monitoring system has existed for years. Thus, there are three chosen fertigation monitoring system will be studied and compares with each other and the proposed system, Smart Plant Monitoring System for cucumber.

Those three chosen plant monitoring systems are Smart Plant Monitoring System Using Aquaponics Production Technological with Arduino Development Environment (IDE) and SMS Alert (Shilpa Devram Pawar & Dr. Damala Dayakar Rao, 2022), The prototype of the Greenhouse Smart Control and Monitoring System in Hydroponic Plants (Arif Supriyanto & Fathurrahmani, 2019), and Automatic Plants Watering System for Small Garden developed by (Astutiningtyas et al., 2021).

2.4.1 Smart Plant Monitoring System using Aquaponics Production Technological with Arduino Development Environment (IDE) and SMS Alert.

To achieve full Smart Plant Monitoring System functionality, this project involves the design and development of a Wireless Aquaponic Automation and Monitoring System, Automation Plantation Control Unit System, and Mobile Dashboard & Notification System. The Power Supply Unit, Process Controller Unit, Plant Sensor Network System, Wireless Data Platform and Notification System are essential parts of this Smart Plant Monitoring System. As the system hardware is intended to be installed outside, it is combined with adequate safety standards that include all areas, such as the electronic standard and the weatherproof requirement. The complete block diagram architecture of the system is shown in Figure 2.2.

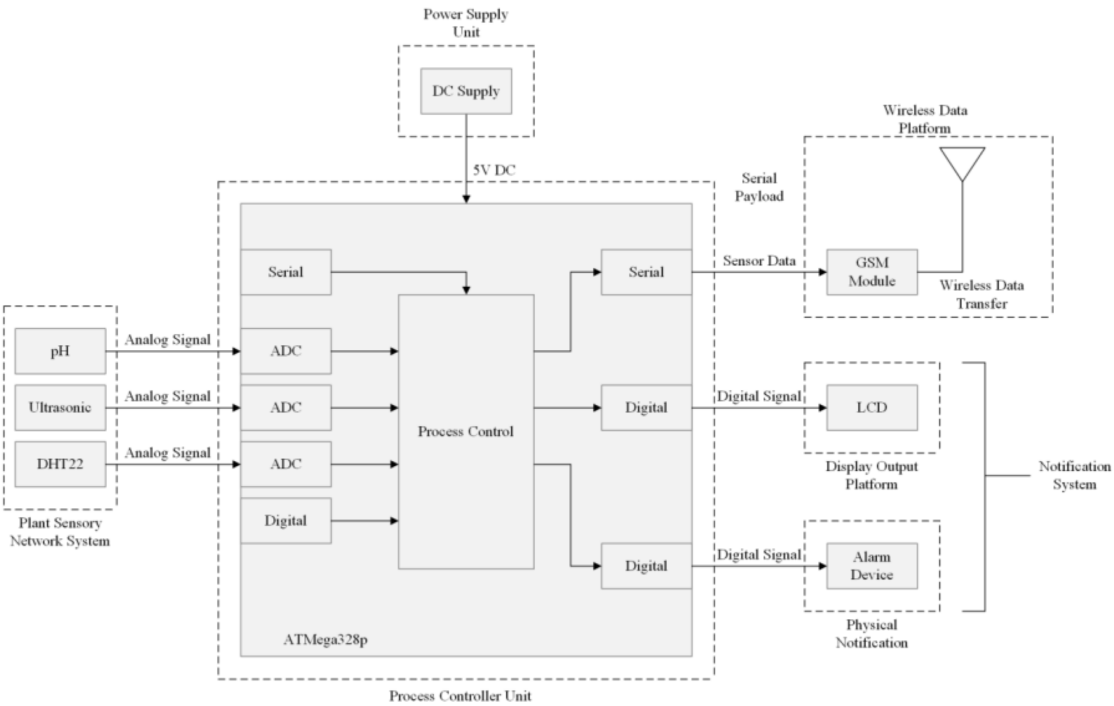


Figure 2.2 Block diagram Smart Plant monitoring System using Aquaponics Production Technological

a. Plant sensory network system

As shown in Figure 2.2, the sensory network for this system comprises of several sensors, including pH, humidity, and temperature sensors, as well as ultrasonic sensor, which measure the ambient temperature and humidity, water condition, and water level. The system is equipped with a GSM data transfer module that operates on the serial data communication protocol.

b. Process controller unit

The project uses the embedded processor ATMega328p as the main controller to process all the sensor data parameters and drive the actuator, which includes the LED indicator and water pump system. To communicate the processed data to the selected dashboard platform, the controller is additionally integrated with the GSM Module of the controller.

c. Mobile dashboard and notification system

All the processed data obtained from the designed plantation system is displayed on the mobile dashboard. Meanwhile, all the processed data from the plantation control is wirelessly transmitted, consequently with the setup period time to the mobile device via the serial communication using the GSM Module. The notifications are sent to the owner via SMS using the GSM Module. The owner can then view real-time data from the appropriate sensors, including water quality (pH), ambient temperature and humidity, and water level. In addition, this Smart Plant Monitoring System includes a warning notification system that will alert the owner to take immediate preventative action if an operating parameter, such as water level or water condition (pH), is out of range. The relay module should be engaged by the controller to activate the alerting mechanism and power the warning buzzer and LED. During the process, the controller should also send the notification to the owner with the mobile dashboard, which has been installed earlier.

2.4.2 The Prototype of the Greenhouse Smart Control and Monitoring System in Hydroponic Plants.

In contrast to previous IoT Smart Plant Monitoring projects, this project makes a smart greenhouse that can control the humidification process by adding a watering system to the air to increase the humidity of the room, in which later all these systems are going to be controlled automatically by a microcontroller based on the reading of the sensors inside the greenhouse room. For the hydroponic planting method in the greenhouse, the system monitoring is used to determine the quality of water in hydroponics in the form of information on the levels of nutrients dissolved in water, water pH, water temperature, and water level in a hydroponic water storage container. The scheme of the research design can be seen in Figure 2.2.

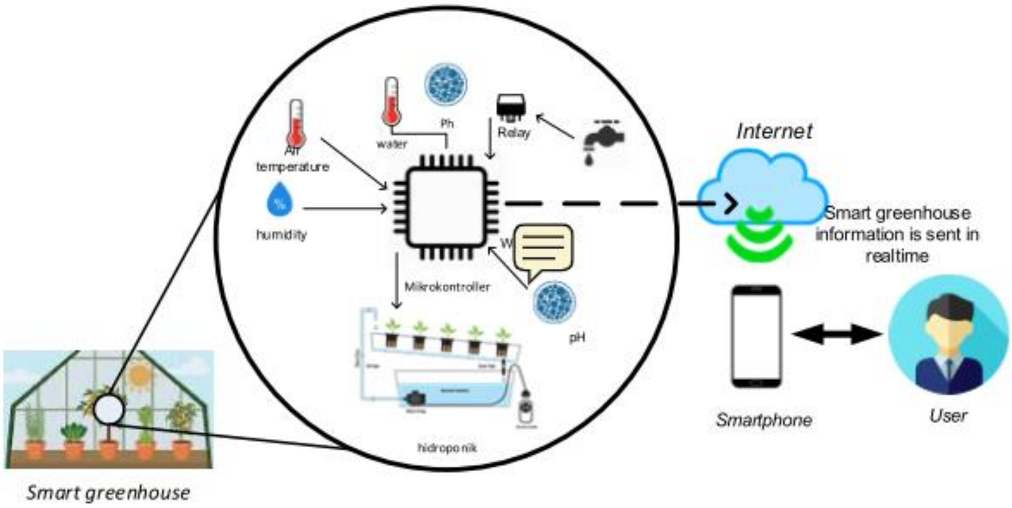


Figure 2.3 Greenhouse System design

Referring to Figure 2.3 above, the smart greenhouse is divided into two parts, which are hardware and software. The humidification control system works when the humidity in the greenhouse starts to deteriorate. When this happens, the watering pump will start automatically to boost the moisture in the room and set the humidity level as befitting.

a. **Hardware Design**

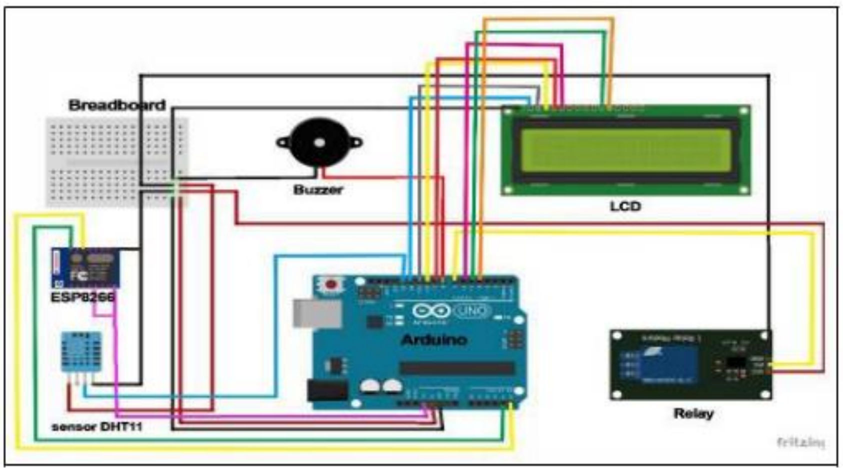


Figure 2.4 Schematic of a humidification control system

In Figure 2.4, it shows the schematic design of the hardware that is used to measure the temperature, humidity and humidification control in a greenhouse. Hardware used is DHT11 Sensor. It is used to measure temperature and humidity, relay, pump, buzzer, LCD screen and Arduino R3 as a microcontroller.

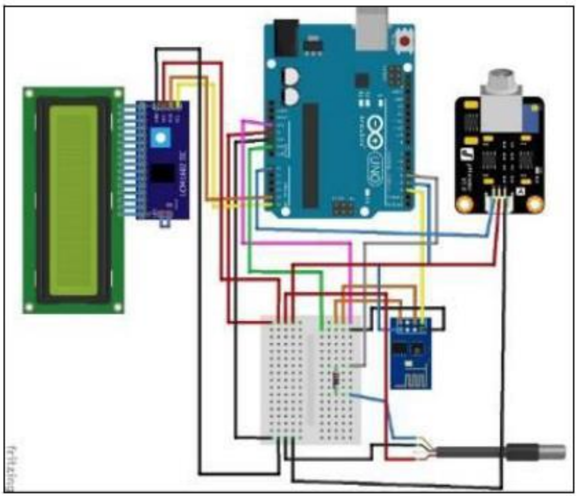


Figure 2.5 Schematic of a hydroponic monitoring system

With reference from Figure 2.5, it shows a schematic of a hydroponic monitoring system that is used to measure water quality in a hydroponic water reservoir. This is the second part of circuit that is included in this project. It uses hardware such as DF Robot analogue water pH sensor kit, DS18b20 water temperature sensor, TDS meter analogue DF Robot sensor, ultrasonic sensor, esp8266 Wi-Fi module, LCD screen, and Arduino as a microcontroller.

2.4.3 Automatic Plants Watering System for Small Garden.

This system focuses on humidity variations that connect with temperature change data by sensors and can control the watering system. To provide cloud-based computing to the system, the level of precision has increased according to farmers' use of the system. The IoT system used is Arduino and cloud to track real-time data from the crop field.

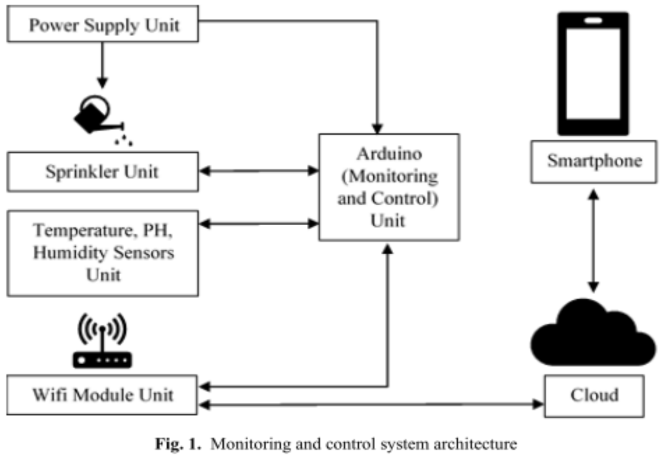


Figure 2.6 Monitoring and control system architecture

As illustrated in Figure 2.6, all system operations are controlled by Arduino as monitoring and controlling. The units connected to Arduino are the power supply unit, sprinkler unit, temperature, PH, humidity sensors unit, Wi-Fi unit, and smartphone. Arduino will receive power from the power supply and collect temperature, PH, and Humidity data from the sensor's unit. Furthermore, the data is programmed displayed on a smartphone due to monitoring the garden's soil moisture using Wi-Fi.

This process is compiled by the Arduino Uno Board and becomes the primary control. The volume of water in the soil is measured by a soil moisture sensor consisting of 2 probes. These probes allow an electric current to pass through the land and measure soil moisture levels based on its resistance. When there is more water, the soil conducts more electricity and causes less resistance, so its humidity becomes high. Conversely, when there is less water, the ground conducts little electricity to lower soil moisture.

2.5 Literature Review Matrix

Table 2.1 Literature Review Matrix

Reference	Implementation Cost	Technology Used	Key Features	Weaknesses
Smart Plant Monitoring System using Aquaponics Production Technological with IDE and SMS Alert.(Shilpa Devram Pawar & Dr. Damala Dayakar Rao, 2022)	• High • Many actuators used.	Atmega328p, GSM module, Environment Sensors	• Wireless Automation • Notification system • Weatherproof • Suitable to cover large areas of crops.	• Higher cost • Complexity in setup • Many signal transmission in a process.
The Prototype of the Greenhouse Smart Control and Monitoring System in Hydroponic Plants.(Arif Supriyanto & Fathurrahmani, 2019)	• Medium	Arduino UNO, Environment Sensors, Relay, water pump	• Easy to setup. • Able to measure water quality. • Able to view real-time data from LCD screen.	• Uses separate microcontrollers which could cost space. • Data transmission delay.
Automatic Plants Watering System for Small Garden.(Astutiningtyas et al., 2021)	• Low	Arduino UNO, Wi-Fi Module, environment sensors, sprinkler.	• Own personalised app. • Capable to set watering time. • Uncomplicated system.	• No security sensors that could detect pests or intruders. • Unable to manually turn on the sprinkler
Smart Plant Monitoring System for Cucumber	• Low	ESP32, Environment Sensors, Water pump.	• Easy to set up • Automated Water pump • Able to detect motion.	• Manual Fertilizer mixture

A unique aspect of this project is the incorporation of mobile devices using the Blynk App for monitoring the cucumber plants and controlling water pump autonomously or manually, dependi87ng on the situation. This addition provides an extra layer of convenience and flexibility, allowing users to access and manage the system from anywhere via their smartphones or computers. This mobile compatibility means this system can monitor and control the water pump whether anywhere as long as there is internet connection.

Efficient management of the water pump through mobile control also reduces energy waste and operational costs which in this project, is paired with rain sensors. Rain sensors could detect if rain occurred in the plant's area. Logically, if rain occurred, there is no need to water the plant anymore which could save water. This system also provides insights on rain occurrence every week which sends the data through an excel file. In a nutshell, this allows for users to take forward actions to avoid future errors.

CHAPTER 3: METHODOLOGY

3.1 Introduction

During research, the methodology is the most important part. Methodology refers to a documented process or step-by-step approach used to aid in developing a successful project. This chapter discusses the phases required to understand the research objectives as stated in Chapter 1.

The project methodology explains the system requirements, design and project cost. The chosen methodology can help determine suitable methods for implementation, testing, and documenting the project intended to be developed.

3.2 Research Methodology Structure

A project management technique called waterfall methodology arranges tasks and activities in a sequential and linear order. Before going to the next process, the current step of each phase needs to be completed first. This methodology provides a clear roadmap and detailed planning and documentation. By using this method, the project is easier to control and manage.

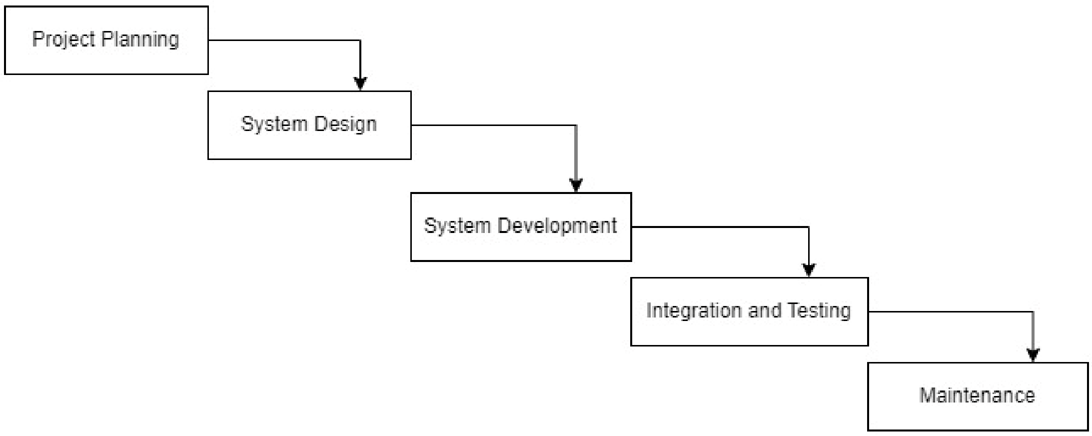


Figure 3.1 Waterfall Method

3.2.1 Project Planning Phase

This phase will gather all the project requirements to get a brief understanding of the project and how the project benefits the consumer. This phase involves identifying the problem statement of the current situation and making sure the objectives are achieved at the end of the project. Literature reviews from previous projects are analysed in this phase. A noticeably clear outline of the project hardware and software requirements will be established after this phase has been completed. A Gantt Chart is created in Appendix B for the full scheduling of tasks and project timeline. This will allow for efficient project management and monitoring of progress throughout the duration of the project.

3.2.2 System Design Phase

In this phase, detailed schematics will be designed based on the hardware that was chosen in the previous phase. Diagrams such as circuit diagrams, block diagrams, and project flowcharts will be designed. These diagrams help define how different components will interact to fulfil the specified functionalities. Blynk web app and phone app will be designed to make sure that the sensors implemented with the hardware will integrate properly with the system. These interfaces will provide real-time data from the environment sensors for remote monitoring. This phase is elaborated in point 3.3: System Design & Architecture

3.2.3 System Development Phase

This phase involved assembling all the components that had been designed in the previous phase into a product. The product will be connected to the IoT platform to make sure that it can be remotely monitored and controlled. All the hardware and software will be integrated together in accordance with both the Project Planning Phase and the System Design Phase. To test the system's full functionality before assembling, I tested it using Wokwi simulation software. This simulation is crucial to ensure that all components are working properly and communicate effectively with each other. This step is important to minimise the error occurrence during the final assembly process.

3.2.4 Integration and Testing Phase

Thorough testing in this phase will be done to find and fix any mistakes or malfunctions, guaranteeing maximum performance and dependability. Each component will go through various checks to detect potential errors or bugs inside the project. It is a crucial phase in improving the project to meet the quality standards and function properly under various conditions. Software like the IoT platform needs to have the monitoring system working properly and the hardware can collect the input and give the desired output. This phase is recorded in Chapter 5: Results and Discussion.

3.2.5 Deployment Phase

Once the testing phase is completed, the product will be deployed to the end user after the testing phase has been completed. This is the phase where the product will be ready to be fully utilised along with the IOT platform. The final prototype is shown in Chapter 4: Results & Discussion with all features and functionalities fully functional as intended. The end user can now provide feedback and suggest any final adjustments before the product is officially launched.

3.2.6 Maintenance Phase

After the deployment phase has been completed, continuous support will be given to the consumer in case any components are faulty. This can result in the product being functional and can do the operation well. This ongoing support ensures that any issues that arise post-deployment can be quickly addressed, minimising downtime and maximising the product's efficiency. Additionally, regular maintenance and updates may also be provided to ensure the product continues to meet the consumer's needs over time.

3.3 System Design & Architecture

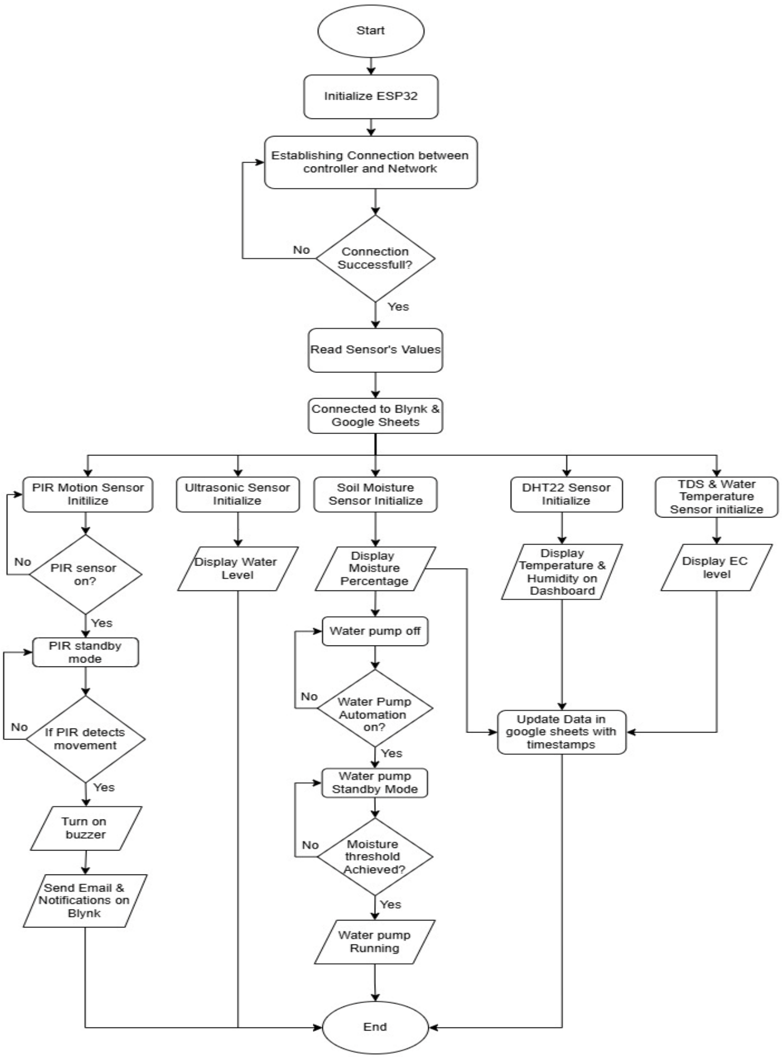


Figure 3.2 Flowchart of the system

Referring to Figure 3.2, the smart plant monitoring system for cucumber starts with the initialization of the ESP32 microcontroller. Once powered on, the ESP32 tries to connect to a network. If the connection isn't successful, the system waits or keeps trying until it gets connected. Once the connection is established, the ESP32 begins reading data from various sensors and connects to both the Blynk IoT platform and Google Sheets. This allows real-time monitoring and automatic data logging.

The system includes several sensors, each serving a specific purpose. The PIR (Passive Infrared) motion sensor is used for security. After it's initialized, the system checks if it's active. If the sensor detects any movement, it triggers a buzzer and sends alerts via email and Blynk notifications. This helps in identifying any unauthorized movement in the monitored area.

An ultrasonic sensor is used to measure the water level in a tank or container. Its readings are displayed on the Blynk dashboard, so users can always know how much water is available. Meanwhile, the soil moisture sensor checks how much moisture is present in the soil. This data is also displayed in real time. If the water pump automation feature is turned on, the system checks whether the soil moisture has dropped below a set threshold. If it has, the pump turns on automatically to water the soil. Once the desired moisture level is reached, the pump switches off.

The DHT22 sensor measures the ambient temperature and humidity. These values are shown on the Blynk dashboard and are also saved to Google Sheets with timestamps. This helps in tracking environmental conditions over time. Lastly, a TDS (Total Dissolved Solids) sensor and a water temperature sensor measure the quality of water by checking its electrical conductivity (EC) and temperature. These values are important for ensuring that the water being used is suitable for plants, especially in hydroponic or precision farming setups.

Overall, this system is used to monitor important environmental factors like soil moisture, temperature, humidity, water level, and water quality, all from a mobile device or computer. It also automates irrigation when needed and alerts the user in case of movement or any unusual activity. This makes plant management easier, more efficient, and smarter.

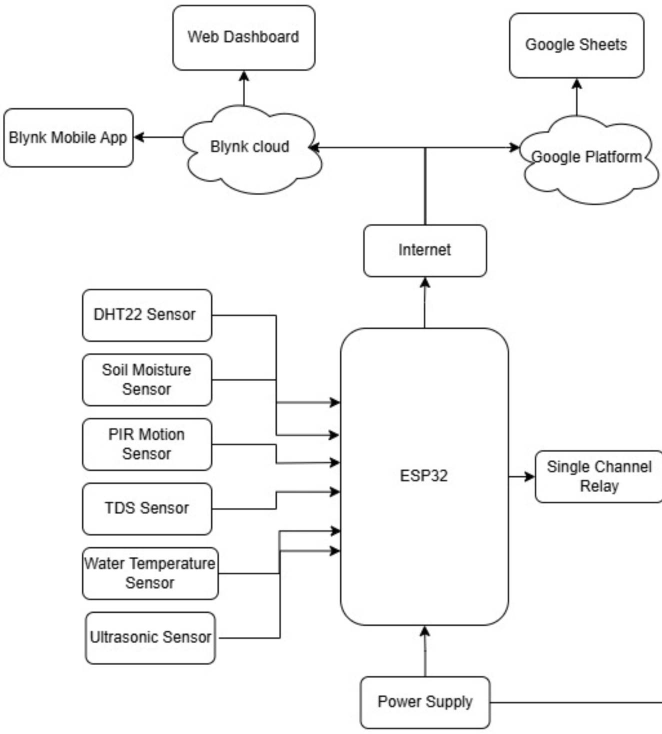


Figure 3.3 Block Diagram of the System

Based on the block diagram shown in Figure 3.3, it shows how the smart plant monitoring system for cucumber is set up. The ESP32 microcontroller, which acts as the main controller for the entire setup. It’s connected to several sensors which includes a DHT22 sensor for checking temperature and humidity, a soil moisture sensor to see how much water is in the soil, a PIR motion sensor for detecting any movement, a TDS sensor to measure water quality, a water temperature sensor, and an ultrasonic sensor to check the water level in a tank.

Once the ESP32 collects data from all these sensors, it connects to the internet to share the information. The data is sent to the Blynk cloud, which allows users to view live readings through the Blynk mobile app or a web dashboard. At the same time, the system also sends the data to Google Sheets through the Google platform, so everything is recorded and can be reviewed later. The system also includes a water pump for automatic irrigation. The ESP32 controls this pump through a NO (normally Open) single-channel relay. Since the pump runs on 12 volts and the ESP32 can’t provide that directly, a step-up voltage converter is used to supply the necessary power. The entire setup is powered through a main power supply connected to the ESP32.

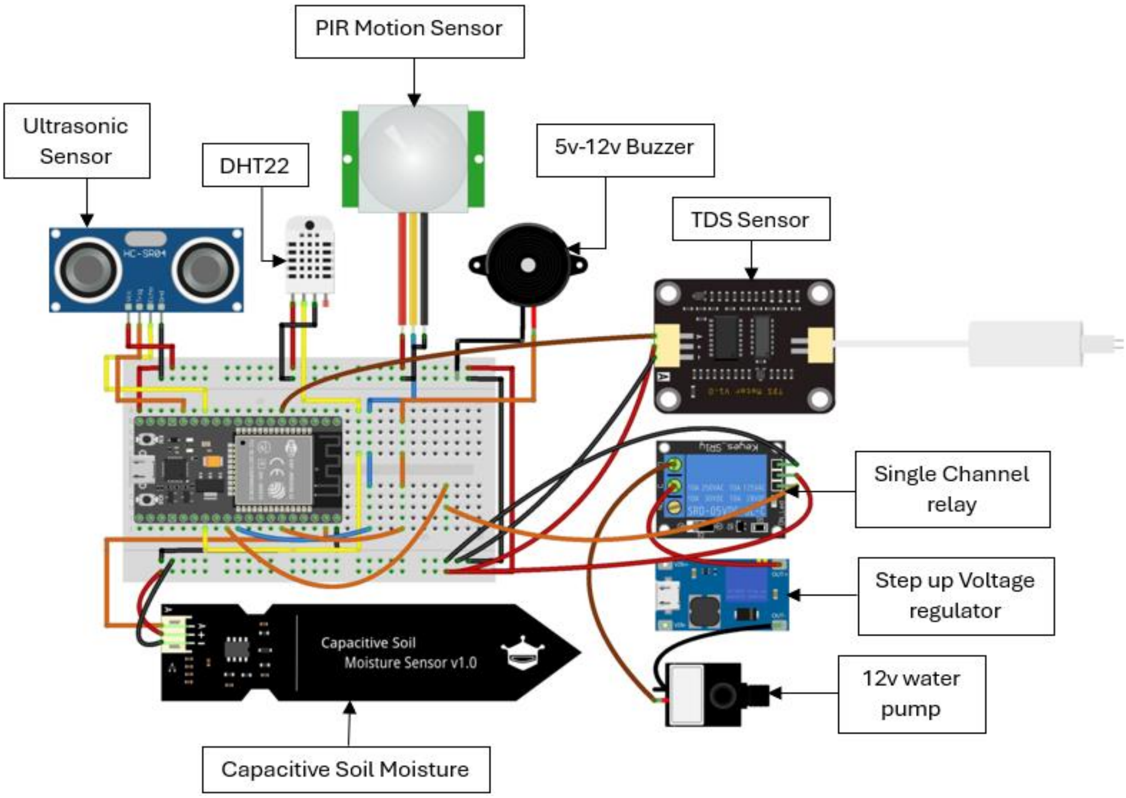


Figure 3.4 Circuit Diagram of the System

3.4 Project Costing

In project costing shows the components that has been use for the project and the specification item that has been use. In Table 3.1 below shows the name of every component that been use and the price for each of them. All items used were separated according to their own specification.

Table 3.1 Project Cost

No.	Components	Price	Quantity	Total
1.	Moisture Sensor	RM5.00	1	RM5.00
2.	TDS Sensor	RM46.00	1	RM46.00
3.	PIR Motion Sensor	RM6.00	1	RM6.00
4.	DHT22 Sensor	RM20.00	1	RM20.00
5.	ESP32	RM30.00	1	RM30.00
6.	ESP32 Shield	RM25.00	1	RM25.00
7.	Step up regulator	RM7.00	1	RM7.00

8.	12v Water pump	RM26.00	1	RM26.00
9.	Buzzer	RM3.00	1	RM3.00
10.	10000 mAh Powerbank	RM20.00	1	RM20.00
11.	3 Slots Battery case	RM3.00	1	RM3.00
12.	Single-Channel Relay	RM8.00	1	RM8.00
13.	Jumper wire	RM3.50	4	RM14.00
14.	Acrylic Box	RM36.00	1	RM36.00
			Exact Total	RM249.00

3.5 Gantt Chart

Gantt chart is a tool used to develop the schedule of this project. It is used to illustrate the start and dates of each phase along with-it sub-tasks. The element in the Gantt chart will follow the tasks that have been created in work breakdown structure as shown in Appendix A. There will be 5 figures that will show the Gantt chart of the project, which is Overall Phase, Project Planning Phase, System Design Phase, System Development Phase, Testing Phase and Maintenance Phase. The Gantt charts for this project can be found at Appendix B.

3.6 System Requirement Analysis

This project involves two major components: hardware and software. The following sections will discuss the specifications, setup and connections, coding, and unit verification and testing for both hardware and software components. The specification section will outline the characteristics of each component. The setup and connection section will detail the installation process and physical connections of the components to the ESP32.

3.6.1 Hardware Requirement

To create a smart plant monitoring system for cucumber, the hardware must be carefully chosen to as it is one of the important parts to get this project done. All the analysis has been made, and the decision has been chosen to select the best hardware for the suitable environment for the system to run. Below is an overview of the hardware components used in building this system.

Microcontroller:

Figure 3.5 ESP32 Dev Kit *(Xukyo, 2024)*

In this project, I utilize the ESP32 as the microcontroller as show in Figure 3.5, which acts as the system's brain and connects to various sensors and actuators used. This microcontroller is specifically designed to be Wi-Fi enabled, allowing all data collected from Environmental Sensors and Actuators condition to be sent to the cloud. This enables remote monitoring and control of the fertigation system through mobile applications or web interfaces.

Environmental sensors:

a. Soil moisture Sensor

Figure 3.6 Soil Moisture Sensor (Yu et al., 2021)

Soil sensors play an important role in fertigation practices by providing valuable information about soil moisture levels, nutrient content and other important parameters. These sensors help farmers regulate irrigation schedules, so plants receive enough water without wasting it or experiencing waterlogging by tracking soil moisture levels.

b. DHT22 Sensor

Figure 3.7 DHT22 Sensor (Syahputra Novelan, 2020)

The DHT11 sensor is used to measure temperature and humidity of the environment. These parameters are crucial for the cucumber growth because cucumbers thrive in specific temperature and humidity conditions.

c. Ultrasonic Sensor

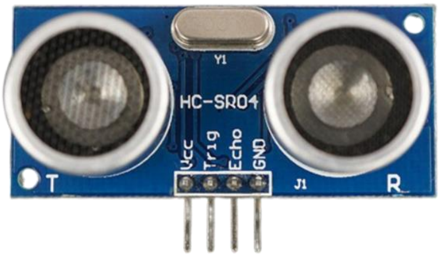


Figure 3.8 Ultrasonic Sensor (Zhang et al., 2023)

The ultrasonic sensor shown in Figure 3.8 is used for real-time detection of the water level. Its function is crucial for effective water resource management and continuous monitoring of water levels. This enables the user to stay informed about the remaining water in the tank and take timely action when refilling is needed.

d. PIR motion sensor

Figure 3.9 PIR Motion Sensor (Ada, 2023)

A PIR (Passive Infrared) motion sensor plays a significant role in security systems, particularly in an IoT-based monitoring system. This sensor is capable to detect motion by sensing changes in infrared radiation (heat) emitted by objects, such as humans or animals, within its field of view. It is highly effective for identifying the presence of a person in a secured area. In addition, when motion is detected, the PIR sensor sends a signal to the ESP32. This signal is processed and used to trigger actions, such as sending notifications to an IoT platform.

e. TDS (Total Dissolved Solids) sensor

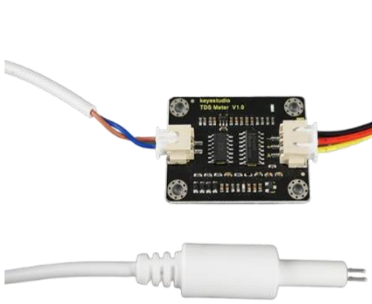


Figure 3.10 TDS sensor (Ada, 2023)

A TDS (Total Dissolved Solids) sensor plays a significant role in managing the quantity of dissolved fertilizers in the water.

<u>Actuators:</u>

a. Single Channel Relay

Figure 3.11 Relay *(ANONIM, 2020)*

A relay connected to a water pump is a critical component in an automated irrigation system or other water control setups. It acts as an interface between the ESP32 and the 5v water pump. The relay enables automation by turning the water pump on or off based on soil moisture data. Once the soil reaches the desired moisture level, the microcontroller turns off the relay, stopping the pump.

b. 5v water pump

Figure 3.12 water pump (Prof. Praveen Rathod et al., 2023)

The water pump as shown in Figure 3.12 in an automated system, such as a smart irrigation or plant monitoring system, plays a vital role in water distribution. It ensures that water is delivered efficiently to plants or specific areas as needed, based on sensor readings or predefined conditions. In this system, the water pump is controlled by data from soil moisture and rain sensors. The relay will activate the pump when soil becomes too dry. The system shuts the pump off when the moisture reaches optimal levels, ensuring the cucumbers receive just the right amount of water. Besides, the water pump is also controlled by the rain sensor. It will turn off the pump during rainfall to avoid overwatering.

c. Buzzer

Figure 3.13 Buzzer (Malvinos, 2006)

Integrating a buzzer with a PIR motion sensor in this smart plant monitoring system enhances its security functionality. Purpose of the Buzzer and PIR Motion Sensor Integration is for intrusion detection and alert. The buzzer serves as an actuator for the PIR Motion sensor that activates immediately when motion is detected. It functions as an alarm, alerting nearby personnel or deterring intruders. Hence, this enables to prevent theft tampering with the plants or equipment.

3.6.2 Software Requirement

a. Arduino IDE

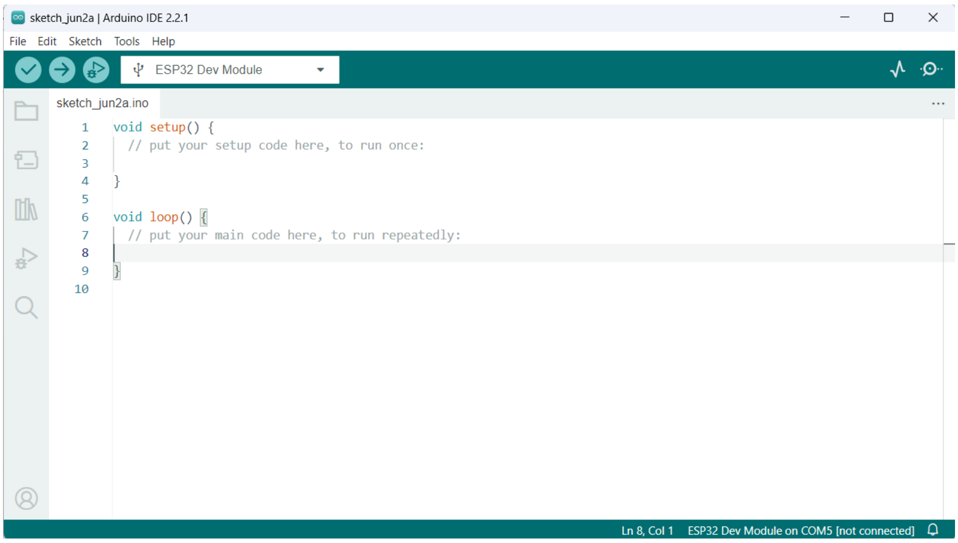


Figure 3.14 Arduino IDE Environment

The Arduino Integrated Development Environment (IDE) is mandatory for developing code for the ESP32 microcontroller, using C++ as the programming language. This open-source platform provides a streamlined and accessible interface for writing, compiling, and uploading code to the ESP32, which is central to controlling various components of the system.

The Arduino IDE supports a wide range of libraries, facilitating the integration of sensors, actuators, and other hardware elements. Its cross-platform compatibility ensures that developers can use it on Windows, macOS, and Linux, making it versatile and user-friendly.

b. Blynk IOT platform

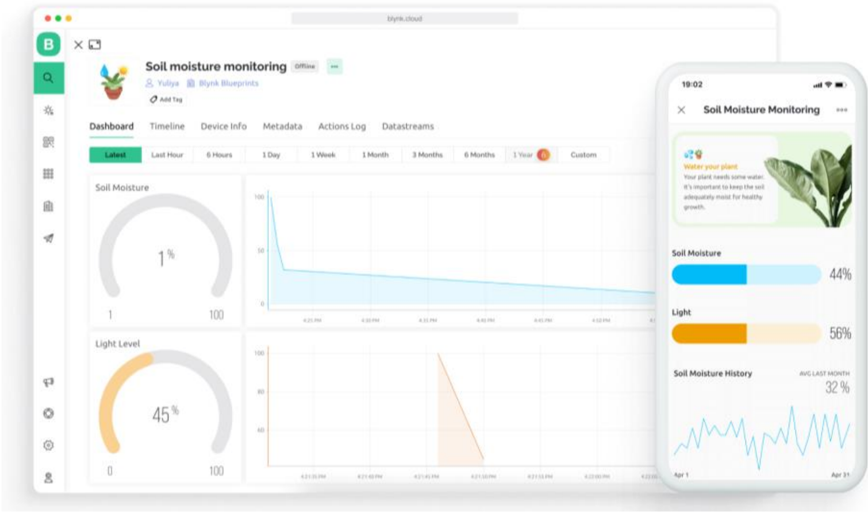
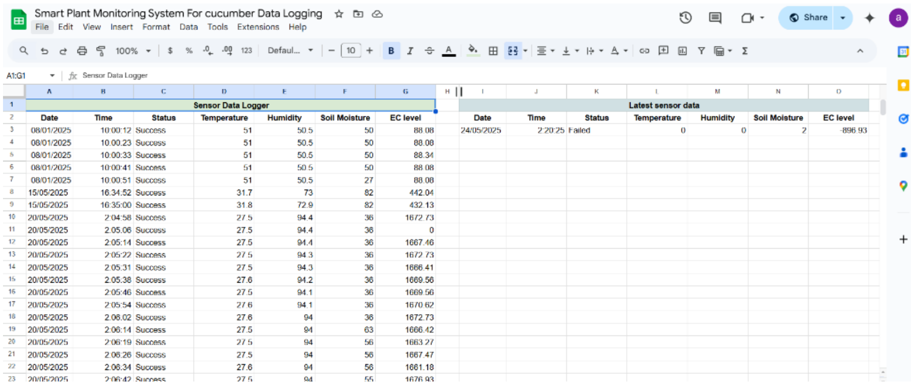


Figure 3.15 Blynk Dashboard

When data is sent to Blynk, it flows through a DataStream using Blynk protocol which is the main communication protocol for this project. Then, every value is automatically timestamped and stored in the Blynk.Cloud database. Furthermore, Blynk comes with its own dashboard with various configurations which makes it an efficient tool in a smart plant monitoring system project. It plays a vital role in enabling real-time interaction between the user and the IoT devices involved in the project.

The Blynk dashboard is capable to monitor environment sensors integrated with the ESP32 in real time. Even the water pump is automated based on the sensor value from the soil moisture and rain sensor, it is also enabling full control of the water pump to manually stop or start the water pump via a virtual button if automated irrigation is not sufficient. Furthermore, in this system, notification system is also developed which allows the system to send alert to the dashboard if it detects any intruders.

c. Google Sheets


Smart Plant Monitoring System For cucumber Data Logging

File Edit View Insert Format Data Tools Extensions Help

Sensor Data Logger						
Date	Time	Status	Temperature	Humidity	Soil Moisture	EC level
08/01/2025	10:00:12	Success	51	50.5	50	88.08
08/01/2025	10.00.23	Success	51	50.5	50	88.08
08/01/2025	10:00:33	Success	51	50.5	50	88.34
08/01/2025	10:00:41	Success	51	50.5	50	88.08
08/01/2025	10:00:51	Success	51	50.5	27	88.08
15/05/2025	16:34:52	Success	31.7	73	82	442.04
15/05/2025	16:35:00	Success	31.8	72.9	82	432.13
20/05/2025	2:04:58	Success	27.5	94.4	36	1672.73
20/05/2025	2.05.06	Success	27.5	94.4	36	0
20/05/2025	2:05:14	Success	27.5	94.4	36	1667.46
20/05/2025	2:05:22	Success	27.5	94.3	36	1672.73
20/05/2025	2:05:31	Success	27.5	94.3	36	1666.41
20/05/2025	2:05:38	Success	27.6	94.2	36	1669.56
20/05/2025	2:05:46	Success	27.5	94.1	36	1669.56
20/05/2025	2:05:54	Success	27.6	94.1	36	1670.62
20/05/2025	2.06.02	Success	27.6	94	36	1672.73
20/05/2025	2:06:14	Success	27.5	94	63	1666.42
20/05/2025	2:06:19	Success	27.5	94	56	1663.27
20/05/2025	2:06:26	Success	27.5	94	56	1667.47
20/05/2025	2:06:34	Success	27.6	94	56	1661.18
20/05/2025	2:06:42	Success	27.5	94	55	1676.93

Latest sensor data						
Date	Time	Status	Temperature	Humidity	Soil Moisture	EC level
24/05/2025	2:20:25	Failed	0	0	2	-896.93

Figure 3.16 Data Logging using Google Sheets

Using Google Sheets to log data from the sensors is a practical choice. It's free, easy to access from anywhere, and doesn’t require any complex setup or extra servers. User were able to view all the sensor data in real time, which made it much easier to monitor what was happening and spot any issues quickly. Google Sheets also gave simple tools to create charts, calculate averages, and track trends over time, something that’s especially useful in my project where it needs to understand how conditions change.

Another big advantage is how easy it is to share the data with others. Whether it’s a teammate, a teacher, or someone else involved in the project, they can view or collaborate on the sheet instantly. Plus, it connects well with other tools like Google Data Studio or automation services, which opens even more possibilities for reporting and alerts. Overall, it helped the system keep things organized, understandable, and accessible without much hassle.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Introduction

This chapter represents the results achieved from the implementation and testing of the Smart Plant Monitoring System for cucumber cultivation. It details the installation process, testing procedures, and system performance outcomes.

Additionally, it highlights key observations and insights gained during the testing phase. To support optimal cucumber plant growth, this section also outlines the plant care procedures followed throughout the project.

4.2 Equipment Installation

The implementation process of the Smart Plant Monitoring System for Cucumber was conducted in a structured manner after completing the design phase. To ensure proper functionality and reliability, the installation was done with attention to detail referring to the references gathered. Key components, including sensors, actuators, relay module, and the ESP32 microcontroller, were interconnected according to the system requirements.

Each component was installed with precise alignment to ensure smooth operation and minimal troubleshooting. Figures and tables are provided to illustrate the connections and pin configurations for both actuators and sensors, offering a comprehensive understanding of the installation setup.

4.2.1 Hardware Installation

The sensors and actuators, which are key to monitor various environmental factors, were connected to the ESP32 microcontroller. The sensors include an ultrasonic sensor for measuring water levels, a soil moisture sensor, a TDS sensor, a DHT22 sensor for temperature and humidity, a PIR sensor connected to a buzzer for security alarm and water temperature sensor. The pin configurations for each sensor were carefully assigned to ensure accurate data collection. Figure 4.1 shows how the sensors are connected, and Table 4.1 outlines the pin descriptions for each sensor. Furthermore, each GPIO pin on the ESP32 is linked to a specific relay input on the single channel relay module, which in turn controls the corresponding actuator.

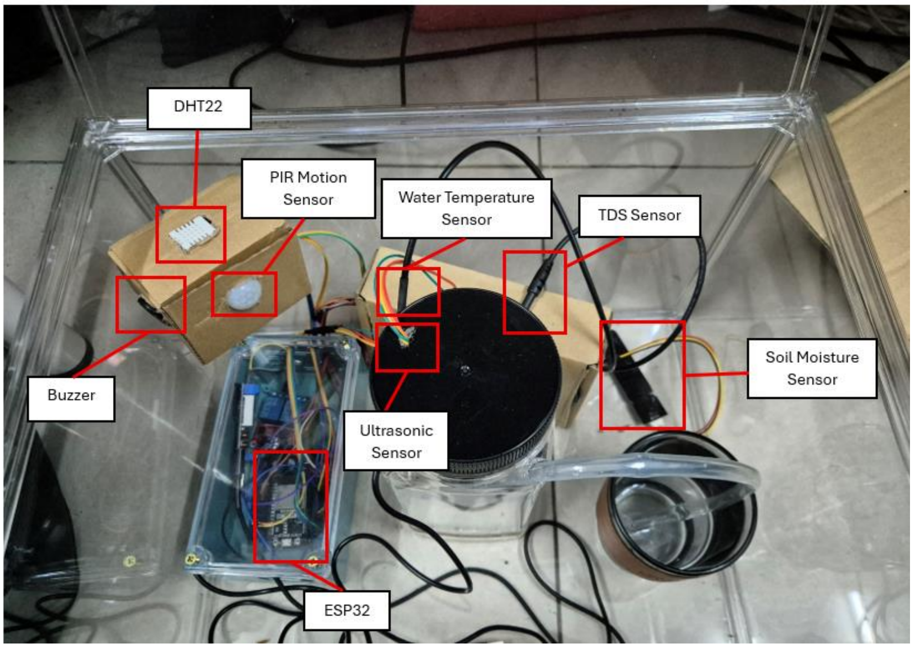


Figure 4.1 Hardware Connections

Table 4.1 provides the pin configuration for each sensor connected to the ESP32 microcontroller. It includes details on the power supply and signal connections for each of the sensors used.

Table 4.1 ESP32 Pins Breakdown

ESP32 Pin	Pin Used	Component
	Sensor Connections	
3.3VDC	VCC	
GPIO12	TRIG	Ultrasonic Sensor (Water Tank)
GPIO13	ECHO	
GND	GND	
3.3VDC	VCC	
GPIO23	AOUT	Soil Moisture Sensor
GND	GND	
3.3VDC	VCC	
GPIO5	OUT	Passive Infrared (PIR) Sensor

GND	GND	
3.3VDC	+	Total Dissolved Solids (TDS) Sensor
GPIO35	A	
GND	-	
3.3VDC	VCC	Water Temperature Sensor (DS18B20)
GPIO2	AOUT	
GND	GND	
3.3VDC	+	DHT22
GPIO4	OUT	
GND	-	
Actuator Connections		
GPIO21	+	Buzzer
GND	-	
5VDC	DC+	12v Single Channel Relay
GND	DC-	
GPIO17	IN1	

4.3 Project Testing and Result

The Smart Plant Monitoring System for cucumber aims to streamline agricultural practices by incorporating monitoring and automation technologies. The system is designed to achieve key objectives such as automated irrigation control, real-time notifications for intruders or pests, and to be able to gather data from sensors, enabling them to be stored in a database for future analysis. The following Table 4.2 outlines the objectives of the project, and the corresponding results achieved during testing.

Table 4.2 Objectives Outcome of the project

No.	Objective	Result
1.	To implement PIR motion sensor configuration to turn on buzzer and send notification to the dashboard and app if it detects any movements or intruders.	Notifications are received reliably via the app and email.

2.	To develop irrigation control systems remotely, allowing for efficient management from anywhere which solves water scarcity.	Successfully Implemented and tested.
3.	To gather data from the environment sensors. The data is sent to an IoT platform where it can be monitored in real time through a mobile app or dashboard.	Data is effectively stored in a database (google sheets) and can be viewed real-time in the Blynk Dashboard.

4.3.1 Hardware Testing

To ensure the successful implementation of the Smart Fertigation System, extensive hardware testing was conducted. Each sensor and hardware component were evaluated for its ability to meet the expected outputs. The results of these tests are summarized in the following table.

Table 4.3 Hardware Testing Results

Hardware	Expected Output	Actual Output
Soil Moisture Sensor	Display soil moisture readings as percentage in Blynk Dashboard and google sheets	Soil Moisture readings are successfully displayed in percentage on both platforms
Ultrasonic sensor	Measure water level and display the data as a percentage.	Water level measurements are displayed accurately as a percentage.
TDS Sensor	Display Total Dissolved Solids (TDS) in ppm and calculate Electrical Conductivity (EC) value.	TDS values are displayed in ppm, and EC calculations are performed accurately.
Water Temperature Sensor	Display water temperature in degrees Celsius (°C) and use the data to assist TDS	Water temperature is displayed in °C, and values are integrated into EC and ppm calculations effectively.

	sensor in calculating EC and ppm values.	
PIR Motion Sensor	To detect any sudden changes in its field of vision while integrating with the buzzer and could be toggled on or off for flexibility.	PIR Motion sensor was able to capture movement of objects. Buzzer were also turned on if it detects any anomaly in its vision.
DHT22 Sensor	To display ambient Temperature in degrees Celsius (°C) and humidity in percentage.	Ambient temperature is displayed in °C, humidity is displayed in percentage and integrated in the system.

4.3.2 Integration Testing

Integration testing is an important phase to develop any IoT-based system, where individual components are combined and tested as an integrated system to ensure uniform communication, data flow, and overall functionality.

Relative on this smart plant monitoring system for cucumber project, integration testing focuses on verifying the correct interaction between sensors, actuators, the ESP32 microcontroller, and cloud platforms such as Blynk and Google Cloud. Based on Figure 4.2, each of the sensors are tested with the microcontroller to achieve the expected result.

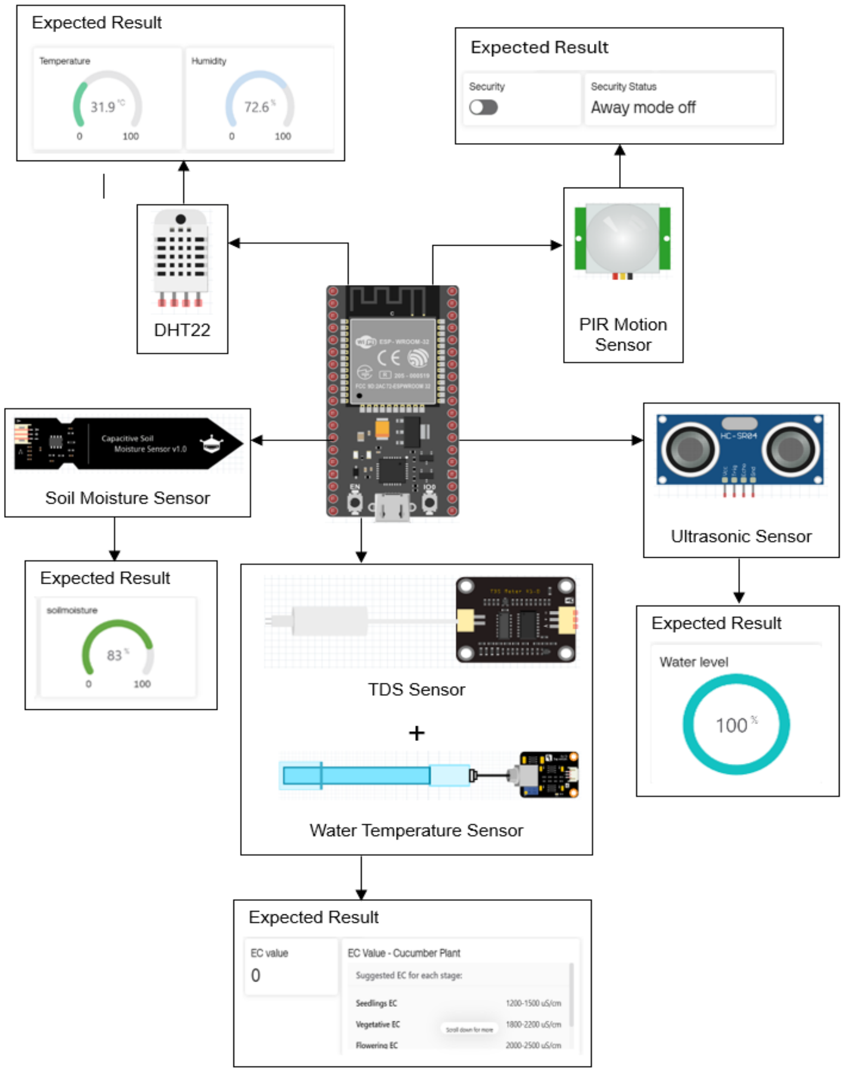


Figure 4.2 ESP32 Integration with Sensors & Expected Result

By managing integration testing as in Figure 4.2, I can validate that the system behaves as expected when all modules are working together, and any inconsistencies or failures can be identified and corrected before deployment. This ensures the robustness, efficiency, and user-friendliness of the complete system.

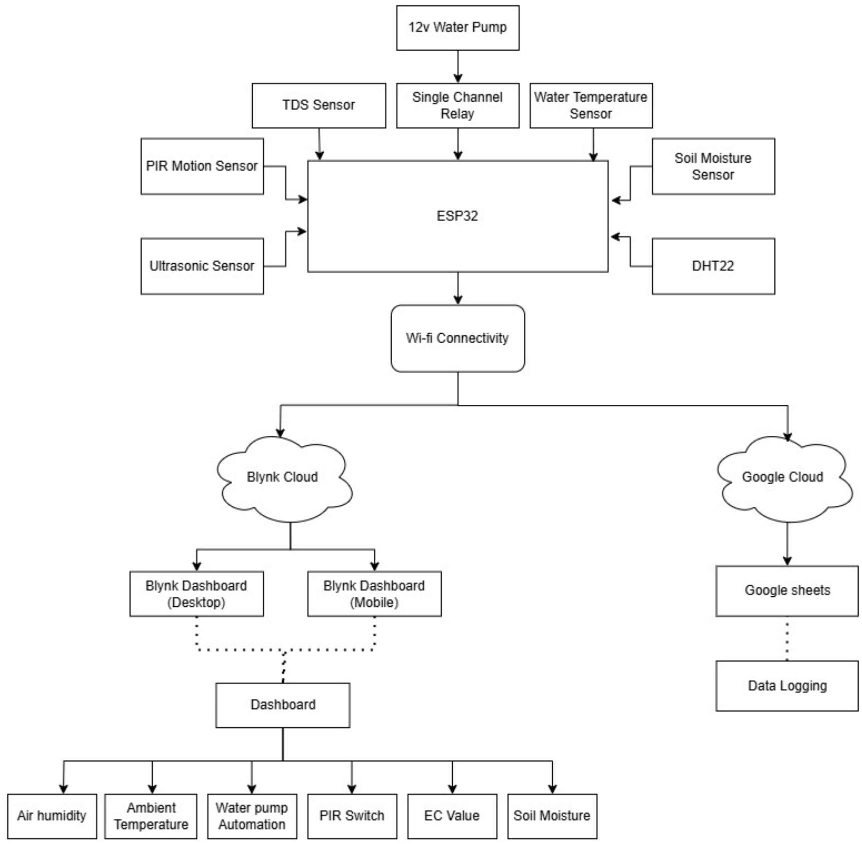


Figure 4.3 Integration Testing Diagram

Indicating Figure 4.3, The ESP32 microcontroller acts as the central hub, interfacing with various environmental sensors such as the DHT22 (for temperature and humidity), soil moisture sensor, PIR motion sensor, ultrasonic sensor, TDS sensor, and a water temperature sensor. It also controls a 12v water pump through a single-channel relay, enabling automated irrigation when required.

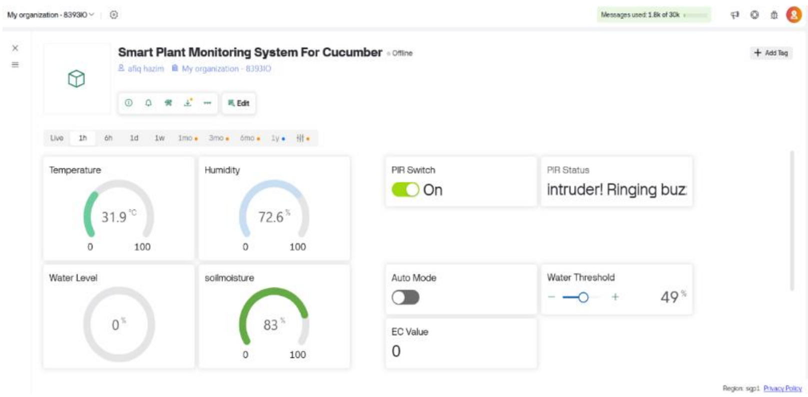
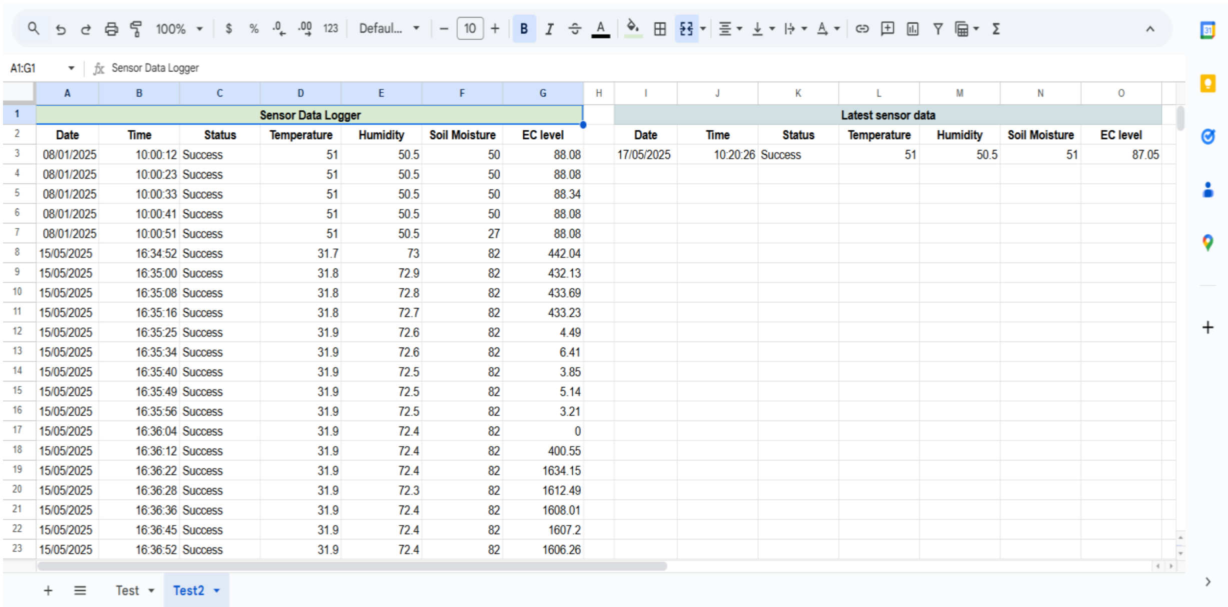


Figure 4.4 Blynk Dashboard Integration

During integration testing, the system is connected to the internet via Wi-Fi, allowing communication with Blynk Cloud and Google Cloud. As shown in Figure 4.3, Blynk Cloud facilitates real-time monitoring and control through both desktop and mobile dashboards.



Sensor Data Logger

Date	Time	Status	Temperature	Humidity	Soil Moisture	EC level
08/01/2025	10:00:12	Success	51	50.5	50	88.08
08/01/2025	10:00:23	Success	51	50.5	50	88.08
08/01/2025	10:00:33	Success	51	50.5	50	88.34
08/01/2025	10:00:41	Success	51	50.5	50	88.08
08/01/2025	10:00:51	Success	51	50.5	27	88.08
15/05/2025	16:34:52	Success	31.7	73	82	442.04
15/05/2025	16:35:00	Success	31.8	72.9	82	432.13
15/05/2025	16:35:08	Success	31.8	72.8	82	433.69
15/05/2025	16:35:16	Success	31.8	72.7	82	433.23
15/05/2025	16:35:25	Success	31.9	72.6	82	4.49
15/05/2025	16:35:34	Success	31.9	72.6	82	6.41
15/05/2025	16:35:40	Success	31.9	72.5	82	3.85
15/05/2025	16:35:49	Success	31.9	72.5	82	5.14
15/05/2025	16:35:56	Success	31.9	72.5	82	3.21
15/05/2025	16:36:04	Success	31.9	72.4	82	0
15/05/2025	16:36:12	Success	31.9	72.4	82	400.55
15/05/2025	16:36:22	Success	31.9	72.4	82	1634.15
15/05/2025	16:36:28	Success	31.9	72.3	82	1612.49
15/05/2025	16:36:36	Success	31.9	72.4	82	1608.01
15/05/2025	16:36:45	Success	31.9	72.4	82	1607.2
15/05/2025	16:36:52	Success	31.9	72.4	82	1606.26

Latest sensor data

Date	Time	Status	Temperature	Humidity	Soil Moisture	EC level
17/05/2025	10:20:26	Success	51	50.5	51	87.05

Figure 4.5 Google Sheets Integration

Air humidity, ambient temperature, soil moisture, EC value and motion detection status are displayed on the dashboard. Users can also test manual and automated control of the water pump via the Blynk interface. Simultaneously, sensor data is sent to Google Sheets via Google Cloud, enabling data logging for validation and analysis as indicated in Figure 4.4. Analysing from Figure 4.4, google sheets captures key environmental parameters essential for monitoring the cucumber plant. A review of the data over several days reveals fluctuations in temperature, humidity, soil moisture, and electrical conductivity (EC) levels.

From the readings on 08/01/2025, environmental conditions appeared stable, with temperature at 51°C, humidity at 50.5%, soil moisture at 50%, and EC level around 88 μS/cm. However, a temperature of 51°C is generally higher than the optimum for most plants, which typically thrive between 20°C to 35°C. If this high temperature is not intentional (e.g., for sterilization or drying), it may negatively affect plant health and productivity.

Humidity levels on both 08/01/2025 and 17/05/2025 were around 50.5%, which is moderately acceptable for many crops, though certain plants require higher humidity for optimal transpiration and growth. On 15/05/2025, the humidity spiked to 72.4%, which is beneficial for tropical or moisture-loving crops, but may pose a risk of fungal growth or rot if not properly ventilated.

Soil moisture showed a significant increase from 50% to 82% on 15/05/2025. While moderate soil moisture (30–60%) is typically ideal for most plants, the 82% level suggests potential overwatering or saturated soil. Consistently high moisture levels can hinder root oxygenation and promote root rot, which may affect plant development if not addressed.

The EC level, which indicates the concentration of dissolved salts and nutrients in the soil, ranged from 432.13 to 1606.26 μS/cm on 15/05/2025. While ideal EC levels depend on the crop type, a typical safe range is 700–1200 μS/cm. Values above this range, such as the peak of 1606.26 μS/cm, indicate high salinity or over-fertilization, which can impair plant nutrient uptake and cause toxicity.

The latest sensor data recorded on 17/05/2025 shows a return to initial levels: temperature at 51°C, humidity at 50.5%, soil moisture at 51%, and EC level at 87.05 μS/cm. While this may indicate a reset or recovery of conditions, the low EC level for cucumber plant could imply insufficient fertilization, potentially leading to nutrient deficiencies over time.

4.3.3 System and Functional Testing

This section evaluates the performance of the Smart Plant Monitoring System for Cucumber through functional testing of the IoT platform and google sheets. The testing focused on validating the user interface, real-time updates, and the functionality of critical system features. These tests ensured that Blynk reliably interacts with the hardware components and meets the project requirements.

a. Blynk Login Page

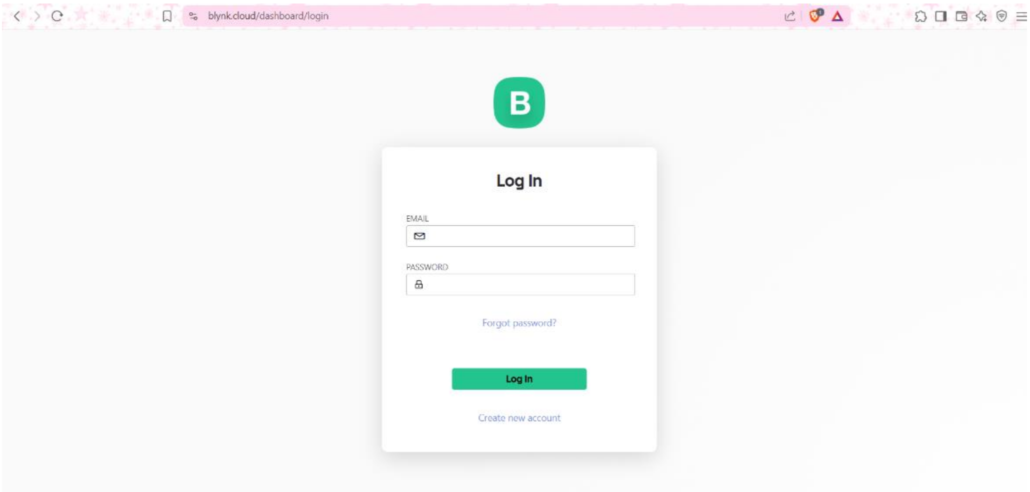


Figure 4.6 Blynk Login

Referring Figure 4.6, shows the login page for the Blynk IoT Cloud Platform, which is accessible at Blynk.cloud. This login page is a critical component of the Blynk IoT platform, as it serves as the secure entry point to access and manage the IoT system. It ensures that only authorized users can view, control, and configure the connected devices and sensor data associated with the smart plant monitoring system. By requiring valid login credentials (email and password), the platform protects sensitive information such as real-time environmental readings, device status, and automation settings from unauthorized access. This login page provides the user with personalized access to the Blynk dashboard, where user can remotely monitor cucumber plant conditions, receive alerts, and make informed decisions regarding irrigation and maintenance. Without this secure access layer, the reliability and integrity of the system could be compromised.

b. Devices page of the Blynk Console.

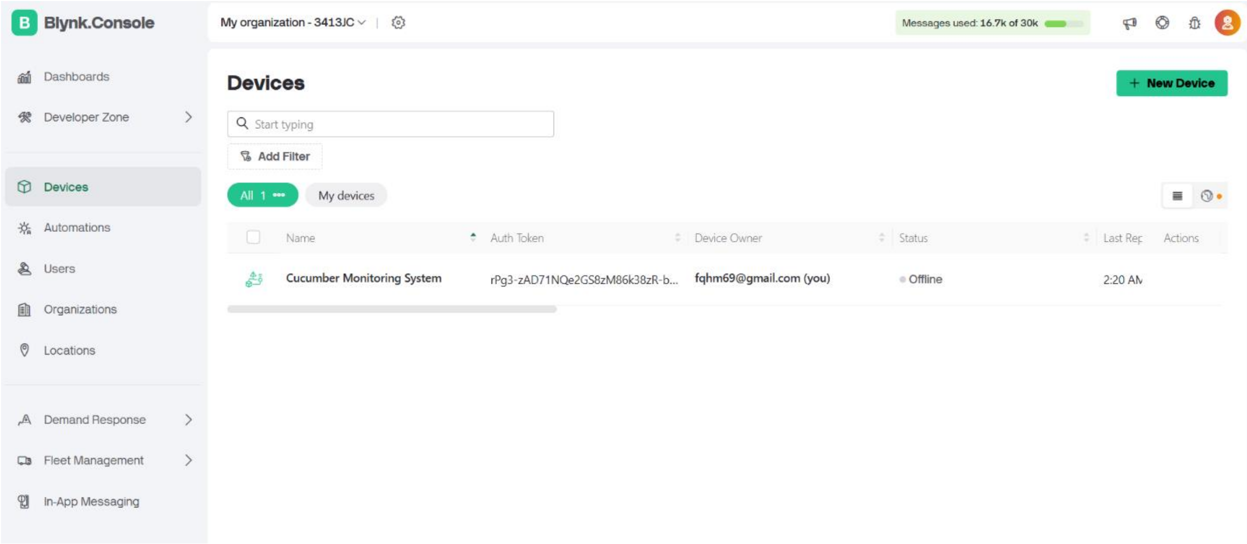


Figure 4.7 Devices Page

Figure 4.7 shows the Devices section of the Blynk Console, where registered devices for an IoT project are listed and managed. In this project, a device named "Cucumber Monitoring System" has been added, representing the hardware setup used for monitoring cucumber plant conditions. Each device is assigned a unique Auth Token that enables secure communication between the ESP32 and the Blynk Cloud.

The console displays essential information including the device name, authentication token, device owner’s email, connection status, and the last reported activity time. In this case, the device is currently marked as Offline, which may indicate that the hardware is not actively transmitting data to the cloud at the moment. This section is important for device management, as it allows users to monitor the status of all connected devices, troubleshoot issues.

c. Blynk Dashboard Access

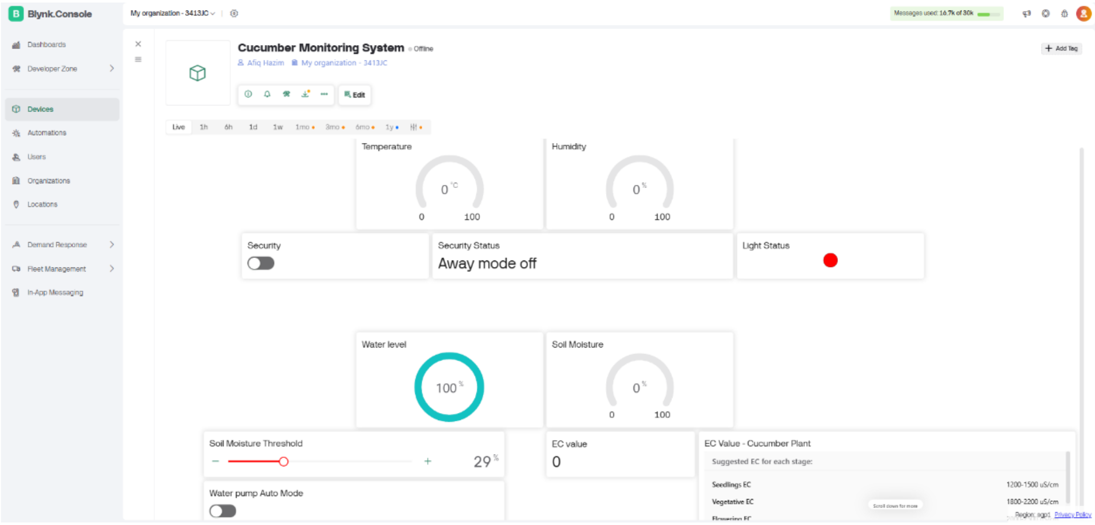


Figure 4.8 Blynk Dashboard

In Figure 4.8, the Blynk Console dashboard displays real-time monitoring data for the “Cucumber Monitoring System.” Key environmental parameters such as temperature, humidity, soil moisture, and electrical conductivity (EC) are updated every 10 seconds from the sensors. The dashboard includes security features with a toggle switch and an "Away mode off" status. Light status is indicated by a red dot indicating it is in off state, and the water level is shown at its current state.

Users can manually set the soil moisture threshold using a slider, which is currently set to 29%, and there is also an option to enable or disable the water pump’s auto mode. The water pump is used to maintain the soil moisture set by the user. This is beneficial to save water if the water level is at a low state and to make sure the soil stays moisture as intended. A reference section for EC values tailored to cucumber plant growth stages is included, providing useful guidance for optimal plant development. Overall, the dashboard integrates sensor data visualization, system control, and reference information to support efficient cucumber cultivation.

d. Google sheets Access

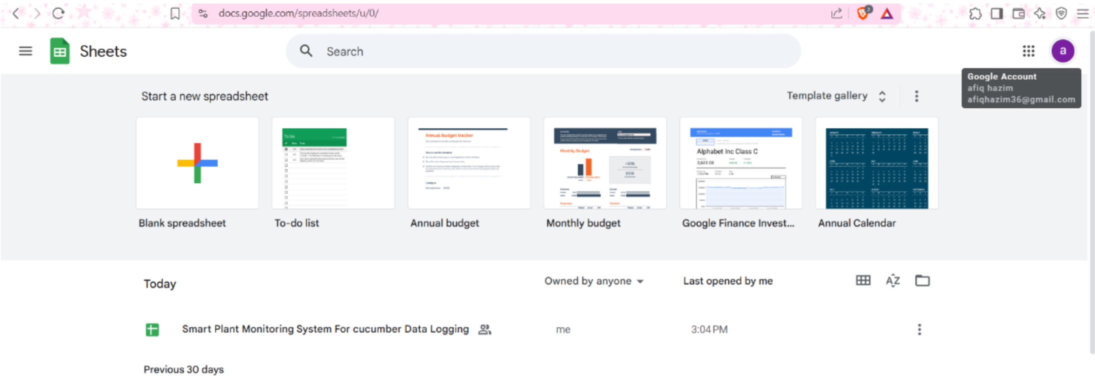
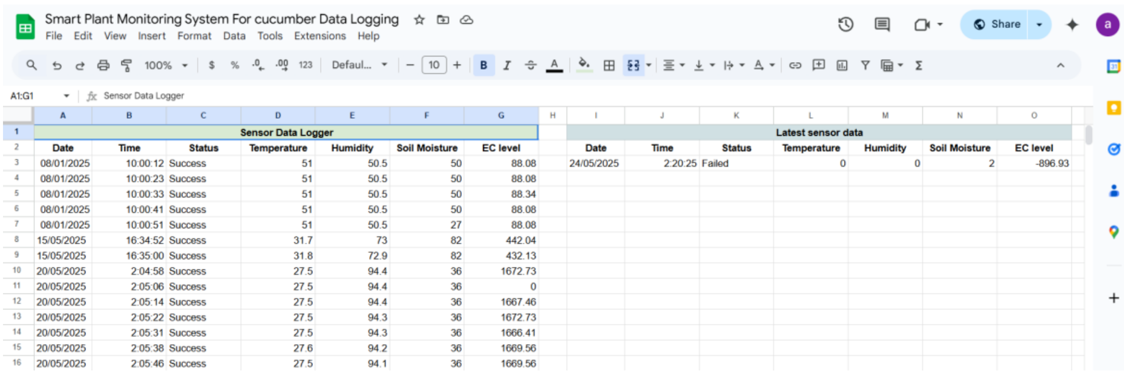


Figure 4.9 Google Sheets Access



Sensor Data Logger								Latest sensor data						
Date	Time	Status	Temperature	Humidity	Soil Moisture	EC level		Date	Time	Status	Temperature	Humidity	Soil Moisture	EC level
08/01/2025	10:00:12	Success	51	50.5	50	88.08		24/05/2025	2:20:25	Failed	0	0	2	-896.93
08/01/2025	10:00:23	Success	51	50.5	50	88.08								
08/01/2025	10:00:33	Success	51	50.5	50	88.34								
08/01/2025	10:00:41	Success	51	50.5	50	88.08								
08/01/2025	10:00:51	Success	51	50.5	27	88.08								
15/05/2025	16:34:52	Success	31.7	73	82	442.04								
15/05/2025	16:35:00	Success	31.8	72.9	82	432.13								
20/05/2025	2:04:58	Success	27.5	94.4	36	1672.73								
20/05/2025	2:05:06	Success	27.5	94.4	36	0								
20/05/2025	2:05:14	Success	27.5	94.4	36	1667.46								
20/05/2025	2:05:22	Success	27.5	94.3	36	1672.73								
20/05/2025	2:05:31	Success	27.5	94.3	36	1666.41								
20/05/2025	2:05:38	Success	27.6	94.2	36	1669.56								
20/05/2025	2:05:46	Success	27.5	94.1	36	1669.56								

Figure 4.10 Data Logging in Google Sheets

By logging in personal email in Google Sheets as in Figure 4.9, user will be able to access the spreadsheet titled "Smart Plant Monitoring System for Cucumber Data Logging". This spreadsheet is used for recording and analyzing sensor data collected from the cucumber monitoring system, such as temperature, humidity, soil moisture, electrical conductivity (EC), and water level as shown in Figure 4.10. Utilizing Google Sheets enables easy access, real-time updates, and secure data storage, supporting better decision-making in the plant monitoring and automation process. This integration demonstrates the effective use of cloud-based tools for IoT data logging and system management.

4.3.4 Notification Testing

Notification testing ensures that alerts are correctly sent and received by user's email. It involves checking message accuracy, proper formatting, and reliable delivery. Based on this project's main objective, it is to configure the system to send notification to the dashboard and app if it detects any movements or intruders. Overall, this process ensures timely and effective communication, alerting users to critical events and enabling prompt action.

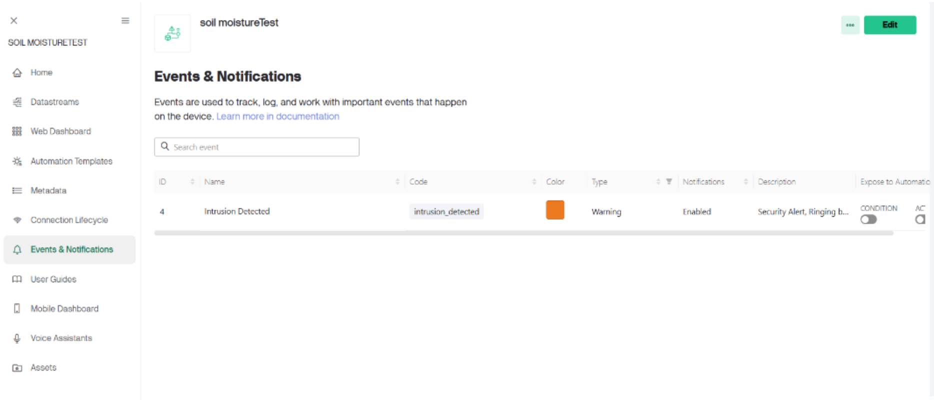


Figure 4.11 Events & Notifications Setting

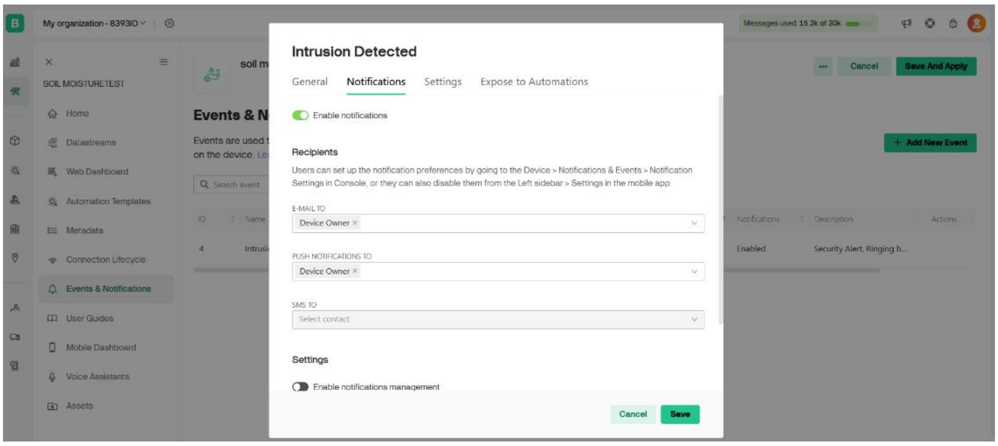


Figure 4.12 "Intrusion Detected" Configuration

Figure 4.12 shows the Events & Notifications configuration in the Blynk Console for a device labelled "soil moistureTest." This section is used to monitor and manage important events that occur on the device. In this case, an event titled "Intrusion Detected" has been set up with the code intrusion_detected, which is used to identify and trigger the event in the system.

The event is categorized as a Warning, indicated with an orange colour for visual emphasis. Notifications for this event are enabled, meaning that users will receive alerts whenever the system detects an intrusion. The description indicates the message that will be sent to the device owner, which is configured in Figure 4.12, representing that the event is linked to a security alert and triggers a buzzer from the hardware.

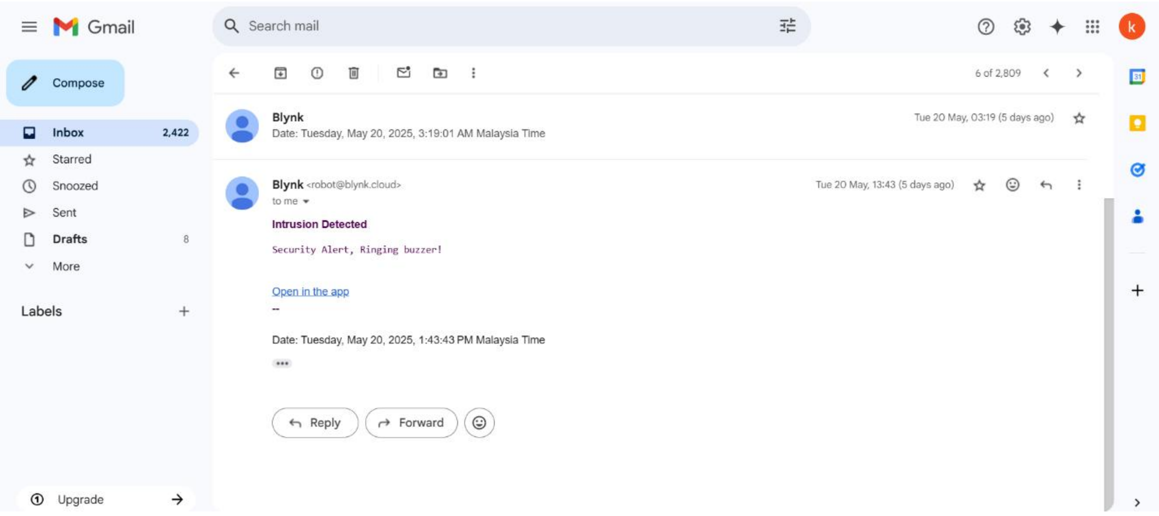


Figure 4.13 Testing Email Notification

Testing the Email notification involves ensuring that the alert message is delivered successfully to the device owner. The message format includes key information such as date & Time with the event name that intrigued the alert labelled “Intrusion Detected” with the description “security Alert, Ringing buzzer!”. Moreover, user can also set the frequency of time for the intrusion email to be sent. This is useful to avoid spammed email from Blynk.

4.4 Discussion

The implementation of the Smart Cucumber Monitoring System successfully demonstrates the integration of various sensors with the ESP32 microcontroller to enable real-time environmental monitoring and automation. By utilizing sensors such as DHT22 (temperature and humidity), soil moisture, water level, TDS (total dissolved solids), PIR motion, and an ultrasonic sensor, the system provides a comprehensive overview of the conditions necessary for healthy cucumber plant growth.

The Blynk platform allows users to view current temperature, humidity, soil moisture percentage, water level, EC (electrical conductivity) value, and light status. The dashboard also includes features for setting soil moisture thresholds and enabling automatic control of a water pump via a relay module. This automation helps to maintain optimal soil moisture levels without requiring manual intervention, which is especially beneficial for precision agriculture and remote plant management.

Additionally, the system logs historical sensor data to Google Sheets, allowing for analysis over time. This feature is crucial for understanding environmental trends and making data-driven decisions regarding plant care. Cloud-based storage also ensures accessibility and backup of the collected data. A notable feature of the system is the integration of a security alert mechanism, where a PIR motion sensor detects potential intrusions. When unauthorized movement is detected, an alert is triggered in the Blynk app, and a buzzer may sound as part of a warning system. This adds an extra layer of utility to the project by incorporating smart home security elements.

However, during testing, it was observed that the system's responsiveness and reliability depend on stable Wi-Fi connectivity, as both Blynk and Google Sheets integration rely on internet access. Any disruptions in the network can temporarily affect data logging and real-time updates. Overall, the project showcases the effective application of IoT technology in agriculture, combining automation, data visualization, and cloud-based logging. It not only improves plant health monitoring but also reduces manual labour and enhances plant security.

CHAPTER 5: CONCLUSION

5.1 Introduction

This chapter summarizes the Smart Plant Monitoring System for Cucumber project, reflecting on the system's performance, its impact, and potential improvements. It encapsulates the findings from the implementation and testing phases, offering an overview of how the system meets its design objectives, alongside a look at possible future developments.

5.2 Project Review

The Smart Cucumber Monitoring System was designed to provide a reliable and automated method for monitoring environmental and soil conditions in cucumber cultivation. By integrating multiple sensors and IoT platforms, the system addresses the common limitations of manual monitoring, such as inconsistency, time consumption, and lack of real-time data access.

The system is built around the ESP32 microcontroller, which acts as the central control unit, connecting with a variety of sensors including the DHT22 for temperature and humidity, soil moisture sensor, TDS sensor for water quality, ultrasonic sensor for water level detection, LDR sensor for light intensity, and a PIR motion sensor for intrusion detection. A single-channel relay is used to control a 12V water pump, enabling automatic irrigation based on soil moisture levels.

All sensor data is visualized and controlled through the Blynk IoT platform, allowing users to monitor real-time values and receive alerts through a mobile interface. In addition, data is logged to Google Sheets, providing a historical view of environmental conditions for analysis and decision-making. The system was successfully tested in a controlled environment and performed reliably under different scenarios. It accurately recorded environmental data, triggered irrigation when needed, and issued alerts when motion was detected. The automation of the water pump, based on soil moisture thresholds, significantly reduced the need for manual intervention and contributed to efficient water usage.

A major highlight of the project was the successful integration of Blynk for remote monitoring and control, as well as Google Sheets for cloud-based data storage. These integrations ensured that users could manage the system from anywhere, at any time, using a smartphone or computer.

5.3 Future Improvements

To further improve the capabilities and effectiveness of the Smart Cucumber Monitoring System, several enhancements can be considered for future development which includes:

1. Integration of solar power, which would make the system more sustainable and suitable for use in remote agricultural areas with limited access to electricity. This would ensure uninterrupted operation while reducing reliance on conventional power sources.
2. Implementing mobile app push notifications or SMS alerts would enhance user responsiveness by immediately informing users of critical events such as low water levels, extreme temperatures, or unauthorized motion detection.
3. Addition of an automatic nutrient dosing system based on real-time TDS or EC readings. This would allow the system to automate the mixing and delivery of nutrient solutions, thereby supporting fertigation and reducing manual input.
4. Incorporating AI-based decision support could enable the system to predict irrigation needs by analysing historical sensor data, environmental conditions, and crop growth stages. This would result in smarter resource management and improved efficiency.
5. The addition of a camera module, such as an ESP32-CAM, would offer visual monitoring for both plant health and security. Image processing algorithms could be used to detect signs of disease, pests, or intrusions.

REFERENCE

Abouelmehdi, K., Elhattab, K., & El Moutaouakkil, A. (2022). Smart Agriculture Monitoring System using Clean Energy. *International Journal of Advanced Computer Science and Applications*, *13*(5). https://doi.org/10.14569/IJACSA.2022.0130544

Ada. (2023). PIR Motion Sensor. *Adafruit Industries*.

ANONIM. (2020). 5V Single-Channel Relay Module - Pin Diagram, Specifications, Applications, Working. In *Https://Components101.Com/Switches/5V-Single-Channel-Relay-Module-Pinout-Features-Applications-Working-Datasheet*.

Arif Supriyanto, & Fathurrahmani, F. (2019). The prototype of the Greenhouse Smart Control and Monitoring System in Hydroponic Plants. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, *10*(2). https://doi.org/10.31849/digitalzone.v10i2.3265

Astutiningtyas, M. B. I., Nugraheni, M. M., & Suyoto. (2021). Automatic Plants Watering System for Small Garden. *International Journal of Interactive Mobile Technologies*, *15*(2). https://doi.org/10.3991/ijim.v15i02.12803

Kirci, P., Ozturk, E., & Celik, Y. (2022). A Novel Approach for Monitoring of Smart Greenhouse and Flowerpot Parameters and Detection of Plant Growth with Sensors. *Agriculture (Switzerland)*, *12*(10). https://doi.org/10.3390/agriculture12101705

Malvinos. (2006). 5V Active Buzzer. *Data Sheet*, *91*.

Mohabuth, A. Q., & Nem, D. (2023). An IoT-Based Model for Monitoring Plant Growth in Greenhouses. *Journal of Information Systems and Informatics*, *5*(2). https://doi.org/10.51519/journalisi.v5i2.489

Mohanty, A., & Roy, S. (2021). Smart Plant Monitoring System Using BLYNK Application. *Lecture Notes in Networks and Systems*, *165*. https://doi.org/10.1007/978-981-15-9433-5_19

Pathmudi, V. R., Khatri, N., Kumar, S., Abdul-Qawy, A. S. H., & Vyas, A. K. (2023). A systematic review of IoT technologies and their constituents for smart and sustainable agriculture applications. *Scientific African*, *19*. https://doi.org/10.1016/j.sciaf.2023.e01577

Pereira, G. P., Chaari, M. Z., & Daroge, F. (2023). IoT-Enabled Smart Drip Irrigation System Using ESP32. *Internet of Things*, *4*(3). https://doi.org/10.3390/iot4030012

Prof. Praveen Rathod, Shreerang Karambelkar, Rutvik Khaire, Mangesh Dhavale, & Pruthviraj Raut. (2023). Automated Water Pump System. *International Journal of Engineering Technology and Management Sciences*, *7*(3). https://doi.org/10.46647/ijetms.2023.v07i03.032

Shilpa Devram Pawar, & Dr. Damala Dayakar Rao. (2022). Smart Plant Monitoring System for Plant Fitness Using IoT. *Journal of Pharmaceutical Negative Results*. https://doi.org/10.47750/pnr.2022.13.s06.102

SMART AUTOMATED PLANT MONITORING SYSTEM USING BLYNK APPLICATION. (2023). *International Research Journal of Modernization in Engineering Technology and Science*. https://doi.org/10.56726/irjmets43727

Syahputra Novelan, M. (2020). Monitoring System for Temperature and Humidity Measurement with DHT11 Sensor Using NodeMCU. *International Journal of Innovative Science and Research Technology*, *5*(10).

Tech Explorations. (2022). *1. What is Blynk? - Tech Explorations*. Tech Explorations.

Xukyo. (2024, February 18). *Overview of the NodeMCU ESP32 microcontroller*. Https://Www.Aranacorp.Com/En/__trashed/.

Yu, L., Gao, W., Shamshiri, R. R., Tao, S., Ren, Y., Zhang, Y., & Su, G. (2021). Review of research progress on soil moisture sensor technology. *International Journal of Agricultural and Biological Engineering*, *14*(4). https://doi.org/10.25165/j.ijabe.20211404.6404

Zhang, N., He, F., Yuan, F., Li, L., Li, Y., & Li, H. (2023). Development and Performance Evaluation of a Infrared Rain Sensor. *Procedia Computer Science*, *228*. https://doi.org/10.1016/j.procs.2023.11.079

APPENDIX A: WORK BREAKDOWN STRUCTURE

A.1 Work Breakdown Structure

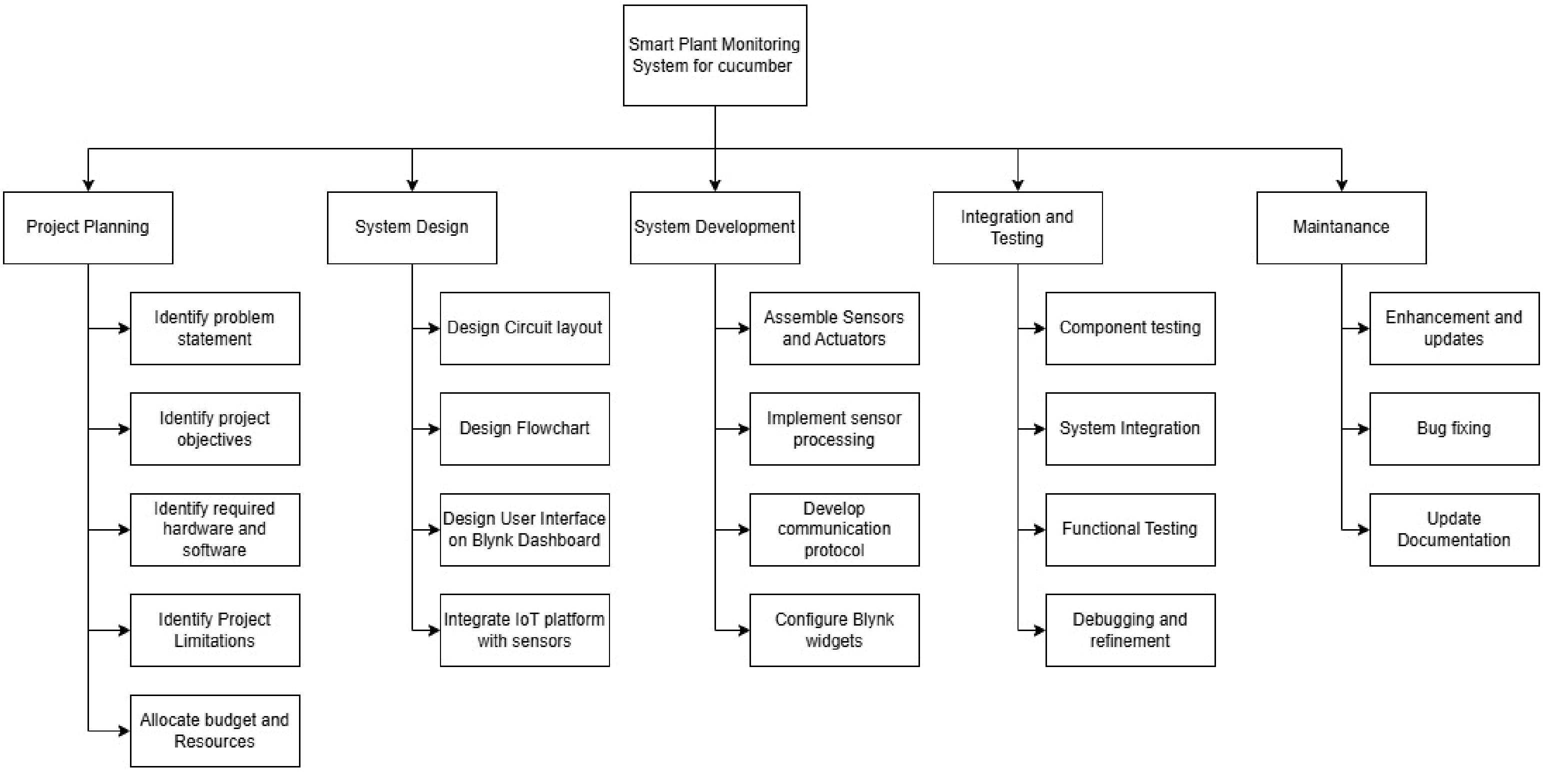


Figure A.1 WBS

APPENDIX B: GANTT CHART

Weeks					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NO	TASK	Duration	START	END															
1	Project Planning	2	1	2															
2	System Design	4	2	6															
3	System Development	3	6	9															
4	Integration and Testing	3	9	12															
5	Maintanance	3	12	15															

Figure B.1 Overall Phase Gantt Chart

Weeks					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NO	TASK	Duration	START	END															
1	Project Planning	2	1	2															
	Identify problem statement	3 Days																	
	Identify project objectives	3 Days																	
	Identify required hardware and software	3 Days																	
	Identify Project Limitations	3 Days																	
	Allocate budget and Resources	2 Days																	

Figure B.2 Project Planning Phase Gantt Chart

Weeks					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NO	TASK	Duration	START	END															
2	System Design	4	2	6															
	Design Circuit layout	10 Days																	
	Design Flowchart	8 Days																	
	Design User Interface on Blynk Dashboard	4 Days																	
	Integrate IoT platform with sensors	6 Days																	

Figure B.3 System Design Phase Gantt Chart

Weeks					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NO	TASK	Duration	START	END															
3	System Development	3	6	9															
	Assemble Sensors and Actuators	5 Days																	
	Implement sensor processing	5 Days																	
	Develop communication protocol	7 Days																	
	Configure Blynk widgets	5 Days																	

Figure B.4 System Development Phase Gantt Chart

Weeks					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NO	TASK	Duration	START	END															
4	Integration and Testing	3	9	12															
	Component testing	5 Days																	
	System Integration	5 Days																	
	Functional Testing	5 Days																	
	Debugging and refinement	6 Days																	

Figure B.5 Integration and Testing Phase Gantt Chart

Weeks					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NO	TASK	Duration	START	END															
5	Maintanance	3	12	15															
	Enhancement and updates	7 Days																	
	Bug fixing	7 Days																	
	Update Documentation	7 Days																	

Figure B.6 Maintenance Phase Gantt Chart

APPENDIX C: PLAGIARISM

FYP2 REPORT-AFIQ HAZIM BIN AZADDIN (52224122132)_final.pdf

ORIGINALITY REPORT

24%	19%	11%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	online-journals.org Internet Source	4%
2	Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur Student Paper	3%
3	journal.unilak.ac.id Internet Source	3%
4	www.learntechlib.org Internet Source	2%
5	ir.unimas.my Internet Source	1%
6	umpir.ump.edu.my Internet Source	1%
7	www.researchgate.net Internet Source	1%
8	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in	<1%

APPENDIX D: ESP32 CODE

```
#define BLYNK_TEMPLATE_ID "TMPL6JIKLSH0S"
#define BLYNK_TEMPLATE_NAME "testing pir"
#define BLYNK_AUTH_TOKEN "rPg3-
zAD71NQe2GS8zM86k38zR-bSssq"

//#define BLYNK_TEMPLATE_ID "TMPL63tXzCIsk"
//#define BLYNK_TEMPLATE_NAME "soil moistureTest"
//#define BLYNK_AUTH_TOKEN
"i9L0Q0ehs9xZKcBxoJbBG6BP4UOn0ZsG"

// Including libraries
#include "WiFi.h"
#include <WiFiClientSecure.h>
#include <HTTPClient.h>
#include <DHTesp.h>
#include <BlynkSimpleEsp32.h>
#include <OneWire.h>
#include <DallasTemperature.h>

// Defining constants and pins
//#define On_Board_LED_PIN  21 (KIV)
#define DHTPIN  16
#define DHTTYPE DHT22

// DS18B20 setup
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
```

```
// TDS sensor setup
#define TDS_PIN 35
const float VREF = 3.3;
const int SAMPLES = 10;

//LDR SENSOR SETTINGS
#define AO_PIN 32
#define LED_PIN 18
#define BLYNK_LED V10  // Virtual LED on V1


// PIR sensor settings
#define PIR 23      // D5 PIR Motion Sensor
#define buzzerPin 21
int pirstate = 0;
int awayHomeMode = 0;
int lastPIRState = LOW;

//int PIR_ToggleValue;

//const int buzzerPin = 21;
const int SOIL_MOISTURE_PIN = 34;
const int TRIG_PIN = 12;  // Ultrasonic sensor TRIG pin
const int ECHO_PIN = 13;  // Ultrasonic sensor ECHO pin
const float TANK_HEIGHT_CM = 20.0; // Tank height in cm
```

```
// Define pin configurations for the water pump
const int relay_1 = 27;
bool relay1_Status = 0;

// Define auto mode condition
int autoMode = 0;
int waterThreshold = 0;        // The soil moisture percentage threshold to activate watering
//float moisturePercentage = 0;  // The calculated soil moisture percentage



// Wi-Fi and Blynk credentials
const char* ssid = "13-1@unifi";
const char* password = "diniunikl1234";
const char* auth = BLYNK_AUTH_TOKEN;

// Google script Web App URL
String Web_App_URL = "https://script.google.com/macros/s/AKfycbwhTKwx7Ot2MKJn11OE-r62ix0ribebI8tsq53zkG9xZ7RFnPSVoOy4H7T6jTUOSltI/exec";

// Variables for sensor readings
float Temp;
float Humd;
float soilmoisture = 0;
float waterLevelPercentage = 0;
float ecValue = 0;
String Status_Read_Sensor = "Failed";
```

```
String Status_Read_Sensor1 = "Failed";
String Status_Read_Sensor2 = "Failed";

// Timer for periodic tasks
BlynkTimer timer;

// DHT sensor object
DHTesp dht;

// Function to read DHT11 sensor data
void Getting_DHT11_Sensor_Data() {
  Humd = dht.getHumidity();
  Temp = dht.getTemperature();

  if (isnan(Humd) || isnan(Temp)) {
    Serial.println("Failed to read from DHT sensor!");
    Status_Read_Sensor = "Failed";
    Temp = 0.0;
    Humd = 0.0;
  } else {
    Status_Read_Sensor = "Success";
  }

  Serial.print("Temp: ");
  Serial.print(Temp);
  Serial.print(" °C, Humidity: ");
  Serial.print(Humd);
```

```
  Serial.println(" %");
}

// Function to read soil moisture sensor data
void soil_moisture_data() {
  int rawValue = analogRead(SOIL_MOISTURE_PIN);
  int soilMoisturePercentage = map(rawValue, 0, 4095, 0, 100);

  if (rawValue < 0 || rawValue > 4095) {
    Serial.println("Failed to read from soil moisture sensor!");
    Status_Read_Sensor1 = "Failed";
    soilmoisture = 0;
  } else {
    Status_Read_Sensor1 = "Success";
    soilmoisture = soilMoisturePercentage;
  }

  Serial.print("Soil Moisture: ");
  Serial.print(soilmoisture);
  Serial.println(" %");
}

// Function to read ultrasonic sensor data
void ultrasonic_sensor_data() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);
  float distance = duration * 0.034 / 2; // Calculate distance in cm

  if (distance < 0 || distance > TANK_HEIGHT_CM) {
    Serial.println("Failed to read from ultrasonic sensor!");
    Status_Read_Sensor2 = "Failed";
    waterLevelPercentage = 0;
  } else {
    Status_Read_Sensor2 = "Success";
    waterLevelPercentage = ((TANK_HEIGHT_CM - distance) /
TANK_HEIGHT_CM) * 100;
  }

  Serial.print("Water Level: ");
  Serial.print(waterLevelPercentage);
  Serial.println(" %");
}

// This function is called every time the Virtual Pin 1 state changes
BLYNK_WRITE(V1) {
  waterThreshold = param.asInt();  // Update watering threshold
  Serial.print("Received threshold.   waterThreshold:");
  Serial.println(waterThreshold);
}
```

```
// This function is called every time the Virtual Pin 2 state changes
BLYNK_WRITE(V2) {
  autoMode = param.asInt();  // Update auto mode status

  if (autoMode == 1) {
    Serial.println("The switch on Blynk has been turned on.");
  } else {
    Serial.println("The switch on Blynk has been turned off.");
  }
}

BLYNK_WRITE(V3) {
  awayHomeMode = param.asInt();  // Set incoming value from pin
V0 to a variable

  if (awayHomeMode == 1) {
    Serial.println("The switch on Blynk has been turned on.");
    Blynk.virtualWrite(V4, "Detecting signs of intrusion...");
  } else {
    Serial.println("The switch on Blynk has been turned off.");
    Blynk.virtualWrite(V4, "Away home mode close");
  }
}


BLYNK_CONNECTED() {
  Blynk.syncVirtual(V3);
  //WRITE LABEL IN WIDGET IN BLYNK
```

```
  Blynk.virtualWrite(V93, "EC Reference:");
  Blynk.virtualWrite(V94, "1200-1500 uS/cm");
  Blynk.virtualWrite(V95, "1800-2200 uS/cm");
  Blynk.virtualWrite(V96, "2000-2500 uS/cm");
  Blynk.virtualWrite(V97, "2200-2700 uS/cm");

}

// Function to send data to Google Sheets
void sendToGoogleSheets() {
  if (WiFi.status() != WL_CONNECTED) {
    WiFi.reconnect();
    Serial.println("Wi-Fi not connected. Skipping Google Sheets update.");
    return;
  }

  String Send_Data_URL = Web_App_URL + "?sts=write";
  Send_Data_URL += "&srs=" + Status_Read_Sensor;
  Send_Data_URL += "&temp=" + String(Temp, 2);
  Send_Data_URL += "&humd=" + String(Humd, 2);
  Send_Data_URL += "&swtc1=" + String(soilmoisture);
  Send_Data_URL += "&swtc2=" + String(ecValue, 2);

  Serial.println("Sending data to Google Sheets...");
  Serial.println("URL: " + Send_Data_URL);

  WiFiClientSecure client;
```

```
  client.setInsecure();  // Disable SSL certificate validation for
testing
  HTTPClient http;

  http.begin(client, Send_Data_URL);
  int httpCode = http.GET();

  if (httpCode > 0) {
    String payload = http.getString();
    Serial.println("Payload: " + payload);
  } else {
    Serial.print("HTTP GET failed, error: ");
    Serial.println(http.errorToString(httpCode));
  }

  http.end();
}

// Function to send data to Blynk
void sendToBlynk() {
  Blynk.virtualWrite(V7, Temp);
  Blynk.virtualWrite(V8, Humd);
  Blynk.virtualWrite(V0, soilmoisture);
  Blynk.virtualWrite(V5, waterLevelPercentage);
  Blynk.virtualWrite(V6, ecValue);  // Send EC to Blynk V6


}
```

```
// Wi-Fi connection setup
void connectToWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to Wi-Fi");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("\nWi-Fi connected!");
}

// Function to control automatic watering based on soil moisture
and user settings
void autoWater() {
  if (autoMode == 1 && soilmoisture < waterThreshold) {

    if (!relay1_Status) {
      turnOnPump();
      Serial.println("-----------------------------");
      Serial.println(waterThreshold);
      Serial.print("  moisturePercentage:");
      Serial.println(soilmoisture);
      Serial.println("Watering...");

      // Turn off pump after 2 seconds
      timer.setTimeout(2000L, turnOffPump);
```

```
    }
  }
}

// Function to turn on the water pump
void turnOnPump() {
  digitalWrite(relay_1, HIGH);
  relay1_Status = 1;
}


// Function to turn off the water pump
void turnOffPump() {
  digitalWrite(relay_1, LOW);
  relay1_Status = 0;
}

void autoWaterTimer() {
  autoWater();
}

void sendECValue(){
  // Read temperature
  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);

  // Read analog
```

```
  int analogValue = 0;
  for (int i = 0; i < SAMPLES; i++) {
    analogValue += analogRead(TDS_PIN);
    delay(10);
  }
  analogValue /= SAMPLES;

  float voltage = analogValue * VREF / 4095.0;

  // Temperature compensation
  float compensationCoefficient = 1.0 + 0.02 * (temperature -
25.0);
  float compensatedVoltage = voltage / compensationCoefficient;

  // EC in µS/cm (from sensor formula)
  ecValue = 133.42 * pow(compensatedVoltage, 3)
          - 255.86 * pow(compensatedVoltage, 2)
          + 857.39 * compensatedVoltage;

  // TDS using 0.64 factor
  float tdsValue = ecValue * 0.64;

  // Output
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");

  Serial.print("EC: ");
```

```
  Serial.print(ecValue, 2);
  Serial.println(" µS/cm");

  Serial.print("TDS: ");
  Serial.print(tdsValue, 2);
  Serial.println(" ppm");

  Serial.println("-------------------------");


}

//FUNCTION TO READ LDR
void readLDR() {
  int lightValue = analogRead(AO_PIN);
  Serial.print("LDR Value: ");
  Serial.println(lightValue);

  if (lightValue > 2000) {
    digitalWrite(LED_PIN, HIGH);          // Turn on physical LED
    Blynk.virtualWrite(BLYNK_LED, 255);   // Turn on virtual LED
    Blynk.setProperty(BLYNK_LED, "color", "#00FF00");    // Green
when ON
  } else {
    digitalWrite(LED_PIN, LOW);           // Turn off physical LED
    Blynk.virtualWrite(BLYNK_LED, 255);     // Turn off virtual LED
    Blynk.setProperty(BLYNK_LED, "color", "#FF0000");    // Red
when OFF
```

```
  }
}


// Setup function
void setup() {
  Serial.begin(115200);
  sensors.begin();
  analogReadResolution(12);
  pinMode(LED_PIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(PIR, INPUT);  // Initialize PIR sensor pin as input
  pinMode(relay_1, OUTPUT);    // set relay1 as output
  digitalWrite(relay_1, LOW);  // Keep pump1B low
  pinMode(buzzerPin, OUTPUT);
  noTone(buzzerPin);  // Ensure it's off at boot



  dht.setup(DHTPIN, DHTesp::DHT22);
  Blynk.begin(auth, ssid, password);

  connectToWiFi();



////////////////////////////////////////////////////////////////
```

```
  timer.setInterval(1000L, readLDR); // Call readLDR() every 1
second
  timer.setInterval(5000L, sendECValue);
  timer.setInterval(5000L, Getting_DHT11_Sensor_Data);
  timer.setInterval(5000L, soil_moisture_data);
  timer.setInterval(5000L, ultrasonic_sensor_data);
  timer.setInterval(8000L, sendToGoogleSheets);
  timer.setInterval(5000L, sendToBlynk);
//  timer.setInterval(3000L, sendPIR);  // Call PIR sensor check
every 3 seconds
//  timer.setInterval(2000L, PIRTimerEvent);
  timer.setInterval(10000L, autoWaterTimer);   // Check watering
conditions every 10 seconds
}


// Main loop
void loop() {
  Blynk.run();
    if (awayHomeMode == 1) {
    int currentState = digitalRead(PIR);

    if (currentState != lastPIRState) {
      lastPIRState = currentState;

      if (currentState == HIGH) {
        Serial.println("Motion Detected!");
        tone(buzzerPin, 1000);
        Blynk.virtualWrite(V4, "Motion Detected!");
```

```
      Blynk.logEvent("intrusion_detected");
    } else {
      Serial.println("No Motion.");
      Blynk.virtualWrite(V4, "No Motion");
      noTone(buzzerPin);
    }
  }
 } else {
 noTone(buzzerPin);
 Blynk.virtualWrite(V4, "Away mode off");
 }
 timer.run();
}
```

APPENDIX E: FINAL YEAR PROJECT POSTER

UNIVERSITI KUALA LUMPUR
MALAYSIAN INSTITUTE OF INFORAMATION TECHNOLOGY
BACHELOR OF INFORMATION TECHNOLOGY (HONS) IN INTERNET OF THINGS
FINAL YEAR PROJECT 2
SMART PLANT MONITORING SYSTEM FOR CUCUMBER

Name: Afiq Hazim Bin Azaddin
ID: 52224122132
Supervisor : Ts. Faridah Binti Yahya

INTRODUCTION

The Smart Plant Monitoring System for Cucumber project aims to implement IoT and sensor-based automation to innovate traditional farming practices. Paired with an array of sensors this system not only captures real-time data but also interprets it into actionable insights. This system indirectly enhances plant health through continuous monitoring, data-driven decision-making, and efficient resource utilization.

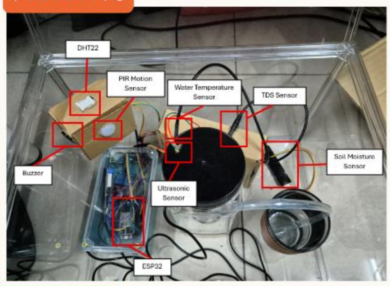
OBJECTIVES

- To implement PIR motion sensor configuration to turn on buzzer and send notification to the dashboard and app if it detects any movements or intruders.
- To develop irrigation control systems remotely, allowing for efficient management from anywhere which solves water scarcity.
- To gather data from the Environment Sensors. The data is continuously sent to an IoT platform where it can be monitored in real time through a mobile app or dashboard.

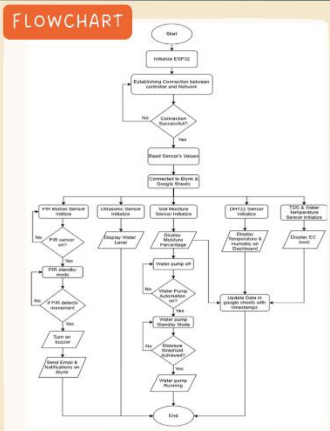
PROJECT LIMITATION

- The system relies on manual mixing of fertilizer into the water tank as it provides no automation to mix them automatically into the water. The quantity of us/cm is provided in the dashboard for reference.
- The system is powered with 2 micro USB which powers Step-up and ESP32 using a 5v output power source (e.g. Power bank). The system will not provide the battery percentage from the power bank, so user have to check it manually. However, it is highly recommended to use a solar powered power source to benefit from the renewable energy.
- PIR Motion sensor detection is limited

PROTOTYPE



FLOWCHART



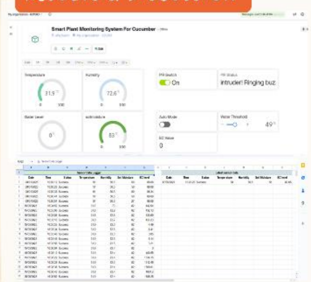
FUTURE IMPROVEMENTS

- Add 2 more water tank to have both fertilizer A & B automated mixing for the system.
- Integrating camera in the system provide enhanced security without depending on PIR Motion Sensor.
- Use solar panel as the Power Source to continuously have power while simultaneously being able to monitor the power source battery percentage.

LITERATURE REVIEW

Reference	Implementation Cost	Technology Used	Key Features	Weaknesses
Smart Plant Monitoring System using Aquaponics Production Technological with IDE and SMS Alert.(Shilpa Devram Pawar & Dr. Damala Dayakar Rao, 2022)	• High • Many actuators used.	Atmega328p, GSM module, Environment Sensors	• Wireless Automation • Notification system • Weatherproof • Suitable to cover large areas of crops.	• Higher cost • Complexity in setup • Many signal transmission in a process.
The Prototype of the Greenhouse Smart Control and Monitoring System in Hydroponic Plants.(Arif Supriyanto & Fathurrahmani, 2019)	• Medium	Arduino UNO, Environment Sensors, Relay, water pump	• Easy to setup. • Able to measure water quality. • Able to view real-time data from LCD screen.	• Uses separate microcontrollers which could cost space. • Data transmission delay.
Automatic Plants Watering System for Small Garden.(Astutiningtyas et al., 2021)	• Low	Arduino UNO, Wi-Fi Module, environment sensors, sprinkler.	• Own personalised app. • Capable to set watering time. • Uncomplicated system.	• No security sensors that could detect pests or intruders. • Unable to manually turn on the sprinkler
Smart Plant Monitoring System for Cucumber	• Low	ESP32, Environment Sensors, Water pump.	• Easy to set up • Automated Water pump • Able to detect motion.	• Unable to measure water levels.

RESULTS & DISCUSSION



- Enabled Data Logging System using google sheets for future analysis for user.
- Interactive Dashboard to view Real-Time Data from Blynk Website or Blynk Mobile.
- Allow user to control water pump and buzzer automation.
- Notification and app alert are automated if PIR detects motion. Email will also be sent to the user's email as an alert.
- EC value is displayed with data calculation from TDS Sensor & water temperature Sensor.

CONCLUSION

This project demonstrates the potential of IoT application to modernize traditional plant monitoring for cucumber practices by integrating real-time environmental Monitoring, automated irrigation, and remote access capabilities.

REFERENCES

- Mohabuth, A. Q., & Nem, D. (2023). An IoT-Based Model for Monitoring Plant Growth in Greenhouses. Journal of Information Systems and Informatics, 5(2). https://doi.org/10.51519/journalisi.v5i2.489
- Pereira, G. P., Chaari, M. Z., & Daroge, F. (2023). IoT-Enabled Smart Drip Irrigation System Using ESP32. Internet of Things, 4(3). https://doi.org/10.3390/iot4030012

APPENDIX F: TECHNICAL PAPER

SMART PLANT MONITORING SYSTEM FOR CUCUMBER

Afiq Hazim Bin Azaddin
Bachelor of Information Technology (Hons) in Internet of Things
Malaysian Institute of Information Technology
University of Kuala Lumpur
Kuala Lumpur, Malaysia
afiq.azaddin@s.unikl.edu.my

Author Ts Faridah Binti Yahya
Project Supervisor
Malaysian Institute of Information Technology
University of Kuala Lumpur
faridahy@unikl.edu.my

Abstract—The integration of Internet of Things (IoT) technologies into precision agriculture presents transformative opportunities for enhancing crop management and sustainability. This study proposes an IoT-based smart monitoring system tailored for cucumber cultivation, focusing on real-time environmental sensing, automated irrigation, and remote accessibility. Utilizing a microcontroller-based platform integrated with soil moisture, temperature, humidity, and light sensors, the system enables continuous monitoring of key growth parameters. Data is transmitted via Wi-Fi to a cloud interface, where it is visualized and used to control irrigation automatically. The system's modular and scalable design supports adaptability across various crop types and farming scales. Unlike conventional manual methods, this approach reduces resource consumption, labor dependency, and the risk of under- or over-irrigation. Comparative analysis with existing smart agriculture systems demonstrates improved efficiency in water usage and enhanced plant health metrics. This work contributes to the growing body of research on sustainable agriculture by offering a low-cost, accessible solution that combines automation with real-time analytics. The proposed system sets a foundation for future advancements in smart farming technologies, encouraging further exploration into crop-specific IoT applications.

I. INTRODUCTION

As the demand for sustainable and efficient farming practices continues to grow, the use of technology in agriculture—particularly Internet of Things (IoT) solutions—has become increasingly important. IoT allows farmers to monitor crops in real time, automate routine tasks like irrigation, and make better decisions based on data. While many smart farming systems have been developed, most are either too general, too complex, or too expensive for small-scale farmers to adopt effectively.

Cucumber is a commonly grown crop that is sensitive to changes in soil moisture, temperature, humidity, and light. Maintaining the right balance of these factors is essential for healthy growth and good yield. However, in many cases, farmers still rely on manual monitoring and guesswork, which can lead to water waste, inconsistent growth, and lower productivity.

This research introduces a cost-effective, crop-specific IoT-based monitoring system designed especially for cucumber cultivation. The system uses sensors to collect real-time data and automatically manages irrigation while allowing remote access through a simple interface. By focusing on the unique needs of cucumber plants and making the solution affordable and scalable, this project addresses a clear gap in current smart farming tools—offering a practical step forward for precision agriculture.

II. LITERATURE REVIEW

A. *Smart Plant Monitoring System using Aquaponics (2022)*

This project proposed a smart plant monitoring system based on aquaponics that utilizes an Atmega328p microcontroller, GSM communication, and environmental sensors. The system provides wireless automation, weatherproofing, and a notification mechanism for large-scale agricultural fields. While technically robust, the implementation is marked by high cost due to the number of actuators and communication modules required. Additionally, the complexity of setup and signal synchronization between multiple components presents operational challenges, making it less suitable for users with limited technical expertise.

B. *Greenhouse Smart Control and Monitoring System (2019)*

Using Arduino UNO and relays, this system supports water quality monitoring and provides real-time feedback via LCD. It is relatively easy to set up and moderately priced. Nonetheless, the use of separate microcontrollers increases cost and system complexity, while delayed data transmission limits real-time responsiveness. developed a prototype for a greenhouse monitoring and control system using Arduino UNO, relays, water pumps, and environmental sensors. This system is relatively easy to deploy and capable of monitoring water quality while displaying real-time data on an LCD screen. However, the use of separate microcontrollers increases the overall cost and design complexity. Furthermore, the system suffers from data transmission delays, limiting its responsiveness in time-critical scenarios such as irrigation control.

C. *Automatic Plant Watering System for Small Gardens (2021)*

The system uses Arduino UNO, a Wi-Fi module, and basic environmental sensors such as soil moisture detectors and sprinklers. It allows users to personalize watering schedules through a mobile application. Although the system is simple and budget-friendly, it lacks critical features such as pest detection, security mechanisms (e.g., motion detection), and adaptive automation. Its functionality is largely confined to sprinkler control, making it unsuitable for more complex agricultural or multi-sensor environments.

III. METHODOLOGY

This section outlines the structured approach taken to design, develop, and implement the Smart Plant Monitoring System for Cucumber. The methodology encompasses both hardware and software integration, sensor calibration, data transmission, and display logic. It was developed in iterative phases to ensure accuracy, reliability, and user-friendliness.

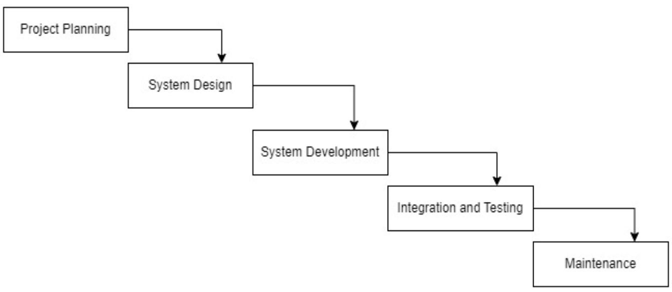


Figure 1: Graphical Illustration of Prototype Model

A. *Project Planning Phase*

The planning phase involves identifying project objectives, defining system requirements, and selecting appropriate tools and components. In the planning phase, the project objectives were identified, system requirements were defined, and appropriate tools and components were selected. The scope and timeline of the project were established, existing solutions were researched to identify gaps and IoT components such as environmental sensors, the ESP32 microcontroller, and communication protocols that were chosen.

B. *System Design Phase*

The design phase centered on developing a detailed plan for the system's architecture, outlining the interaction between hardware components and software functions. A block diagram was created to visualize the data flow from sensors to the cloud-based dashboard, a flowchart was constructed to represent the system's operational logic, and a schematic diagram was produced to assist with hardware assembly and integration.

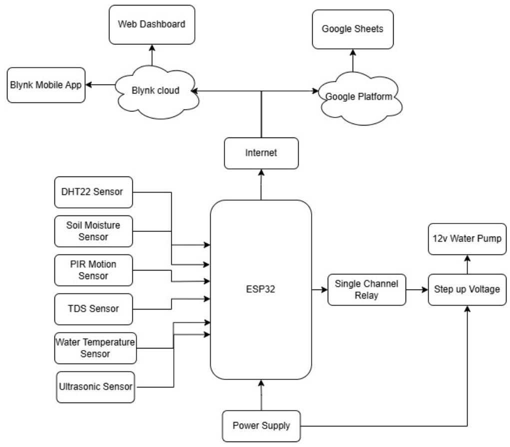


Figure 2: Block Diagram Smart Plant Monitoring System for Cucumber

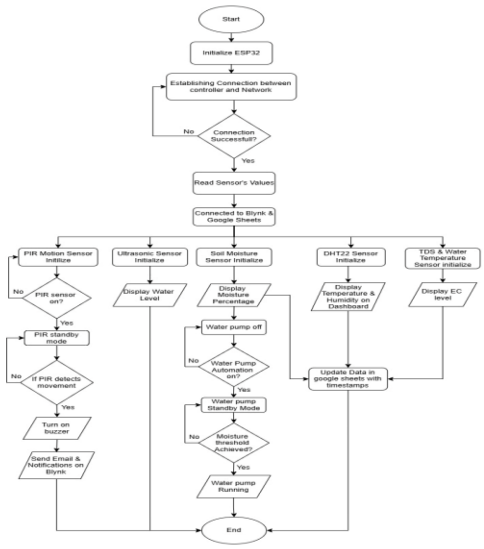


Figure 3: Flowchart Smart Plant Monitoring System for Cucumber

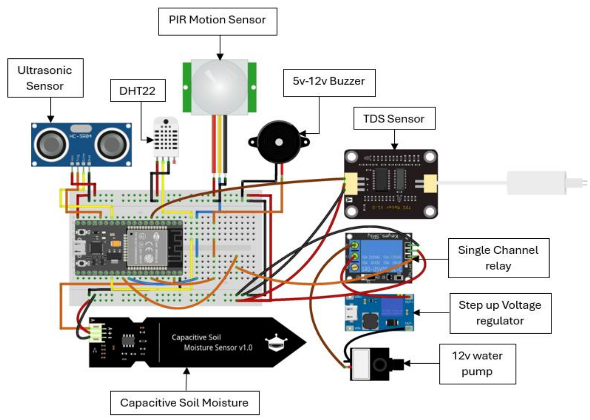


Figure 4: Circuit Diagram Smart Plant Monitoring System for Cucumber

C. System Development Phase

The development phase involved both hardware assembly and software programming. Hardware components such as current and voltage sensors, relays, and the ESP32 microcontroller were assembled and wired correctly. For software development, the ESP32 was programmed using the Arduino IDE, a cloud-based dashboard was set up using Blynk for data visualization, and the Blynk API protocol was implemented to enable real-time communication.

Figure 5: Hardware assembly Testing

D. Integration and Testing Phase

Integration and testing phase is an important phase to develop any IoT-based system, where individual components are combined and tested as an integrated system to ensure uniform communication, data flow, and overall functionality. Relative on this system, integration testing focuses on verifying the correct interaction between sensors, actuators, the ESP32 microcontroller, and cloud platforms such as Blynk and Google Cloud. Each of the sensors are tested with the microcontroller to achieve the expected result.

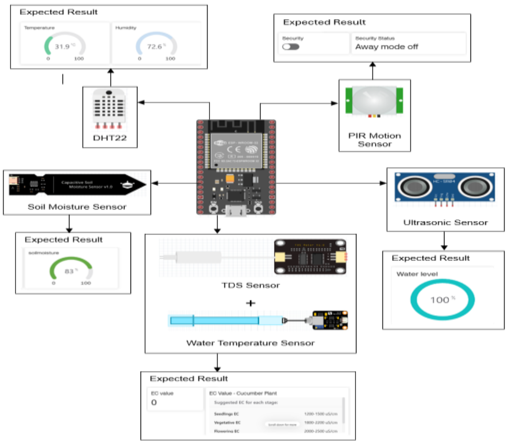


Figure 6: ESP32 Integration with expected results

By managing integration testing, I can validate that the system behaves as expected when all modules are working together, and any inconsistencies or failures can be identified and corrected before deployment. This ensures robustness, efficiency, and user-friendliness of the complete system. Following the successful simulation, the project proceeded to the physical testing phase using real components. The sensors were physically wired to the ESP32 microcontroller and connected to a power supply for practical validation.

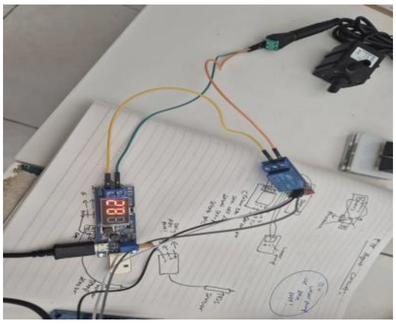


Figure 7: Physical testing phase using real components

E. Deployment Phase

The deployment phase involved transitioning the system from a simulated environment to a real-world setting to ensure its operational readiness. The hardware components, including sensors, relays and the ESP32 microcontroller, were installed at the designated location, ensuring secure and proper connections. The cloud-based dashboard was configured to provide end-users with seamless access to monitor energy consumption and receive real-time alerts. This phase also included fine-tuning the system to adapt to the physical environment, ensuring that the hardware and software worked cohesively. Initial tests were conducted post-installation to validate system functionality and performance under real-world conditions. These steps ensured a smooth transition from development to implementation.

F. Maintenance Phase

The maintenance phase focused on evaluating the system's performance and identifying areas for improvement over a defined monitoring period. Data collected from the system was analyzed to validate its accuracy, efficiency, and reliability under actual usage conditions. User feedback was actively gathered to understand end-user experiences, address any usability concerns, and refine the system for better performance. Additionally, the data trends were studied to detect any inconsistencies, verify energy consumption patterns, and ensure the system met the established project objectives. This phase also included a thorough assessment of the hardware and software components to identify potential upgrades or optimizations that could enhance system functionality and durability.

IV. PROJECT DEVELOPMENT

The development of the Smart Plant Monitoring System for Cucumber involved a structured and iterative process combining both software and hardware components to achieve accurate energy monitoring, real-time data transmission, and clear local display. The project was developed in phases—beginning with research and planning, followed by simulation, prototyping, hardware assembly, software integration, testing, and refinement. Each stage was guided by practical considerations such as cost-efficiency, component availability, and system reliability. Emphasis was placed on ensuring the system could measure Environment Sensors' value accurately while providing a user-friendly interface through Blynk. This chapter outlines the tools used, the configuration and connection of components, the code development process, and the final prototype integration.

A. Software Setup

The software setup plays a crucial role in enabling communication between the hardware components, processing data, and visualizing the output in the Smart Plant Monitoring System for Cucumber. The key tools used include the Arduino IDE for programming, Blynk for cloud monitoring, and google sheets for data logging history data.

1) Arduino IDE: The program development of project's system is developed using the Arduino IDE. The libraries and packages need to be installed. Once all the necessary libraries and packages are installed, the functions they provide can be called in the program. This allows the program to access the added functionality.

2) Blynk Platform: Blynk was used to create a real-time dashboard accessible via PC and mobile. A template was configured to display values for Environment Temperature, Ambient Humidity, water level and soil moisture level. Furthermore, the system allows users to turn on or off the security switch while displaying the status of the switch. If the switch is turned on and detects any motion within its range, it will turn on the buzzer and send an alert email to the user. An auto mode button was added to give flexibility to only turn it on depending on the situation. The water pump auto mode is synchronized with the water threshold value that allows users to maintain the soil moisture value. EC value measurement is added to keep user updated of the fertilizer mixture in the water tank.

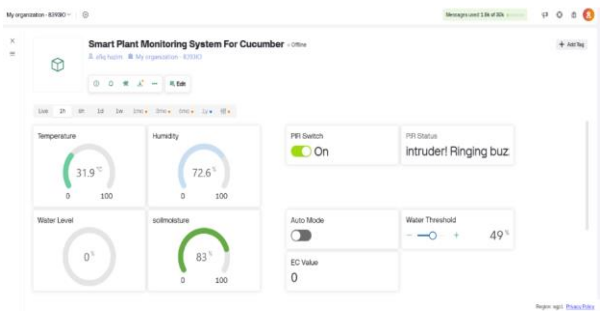


Figure 8: Blynk Dashboard

3) Google Sheets Platform: Google Sheets is utilized in this project as a cloud-based platform for real-time data logging, enabling remote access, automatic data backup, and easy visualization of sensor readings, while also offering a cost-effective and user-friendly solution for storing and analyzing environmental parameters essential to cucumber plant monitoring.

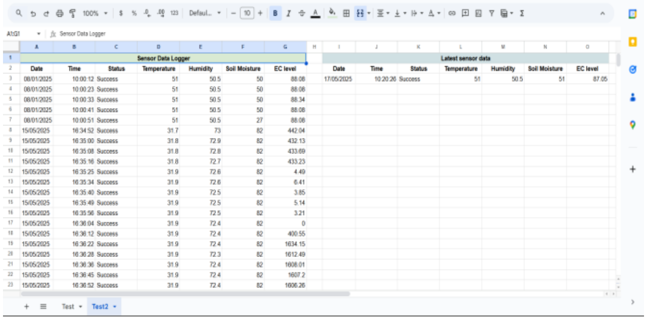


Figure 9: Google Sheets Data Logging

B. Hardware Setup

The hardware components used in this project were selected based on their compatibility, precision and suitability for Plant Monitoring Applications. Below is a detailed explanation of the specifications and roles of each key component in the system.

1) ESP32 Microcontroller: The ESP32 is a powerful 32-bit dual-core microcontroller equipped with built-in Wi-Fi and Bluetooth functionalities. It serves as the central processing unit of the project, handling the acquisition of sensor data, computation of power and energy usage and communication with the Blynk IoT platform. Its high-speed ADC, multiple GPIO pins, and energy efficiency make it ideal for real-time monitoring applications. The ESP32 receives analog signals from both the current and voltage sensors, processes the data internally and transmits the results to both the local LCD and cloud dashboard.

2) ESP32 shield and Jumper Wires: A standard ESP32 Shield to mount the ESP32 microcontroller is used for prototyping the circuit. All components, including the ESP32, sensors and relay, are mounted onto the ESP32 shield and connected using jumper wires. This setup provides flexibility for modifications and testing during the development phase. It is especially helpful during the initial design and debugging process, where frequent adjustments may be necessary.

3) DHT22: The DHT22 sensor was selected for this project due to its ability to accurately measure two crucial environmental parameters: temperature and humidity, both of which significantly influence cucumber plant growth and health. Compared to its predecessor, the DHT11, the DHT22 offers a wider sensing range and greater precision. The sensor communicates digitally, reducing the chances of signal degradation over longer distances and simplifying the interface with microcontrollers like the ESP32. Its low power consumption also supports energy-efficient system operation, which is important in field deployments or battery-powered applications. Additionally, the DHT22 is cost-effective, readily available, and easy to integrate into both prototype and production-level systems. These advantages collectively make it a reliable and practical choice for continuous, real-time environmental monitoring in smart agriculture systems focused on crops like cucumber.

4) Soil Moisture Sensor: This sensor is used to monitor the water content in the soil, which is a critical factor in cucumber cultivation. Cucumber plants require consistent and optimal soil moisture levels to avoid issues such as under-irrigation or over-irrigation. This sensor provides real-time data that enables the system to automate irrigation efficiently, ensuring that water is supplied only when needed. Its integration supports water conservation and enhances overall crop health.

5) PIR motion Sensor: The PIR (Passive Infrared) motion sensor is incorporated into the system for basic crop protection and monitoring. It detects movement near the plant area, which can be useful for identifying the presence of animals or unauthorized human activity. While not directly related to plant growth, it adds a layer of security to the smart agriculture setup. This can help prevent crop damage or theft, especially in open-field or minimally supervised environments.

6) TDS sensor + Water Temperature sensor: In this project, the TDS sensor, in conjunction with a water temperature sensor, is used to measure the Electrical Conductivity (EC) of the irrigation water. EC is a critical parameter in agriculture, as it reflects the concentration of dissolved salts and nutrients in the water, which directly influences plant nutrient availability and uptake efficiency. Cucumber plants are sensitive to nutrient imbalances; both deficiency and excess can impact growth, yield, and fruit quality. The TDS sensor estimates EC by detecting the conductivity of the solution, while the water temperature sensor compensates for temperature variations that can affect conductivity readings. Accurate EC measurement ensures that the nutrient solution or irrigation water maintains optimal salinity levels, supporting healthy root function and reducing the risk of osmotic stress.

7) Ultrasonic sensor: The ultrasonic sensor is employed in this project to accurately measure the water level in the irrigation reservoir. Maintaining an adequate water supply is essential for the continuous operation of the automated irrigation system. The ultrasonic sensor functions by emitting ultrasonic waves and measuring the time it takes for the echo to return after reflecting off the water surface. This time is then converted into a distance measurement, which allows the system to calculate the current water level.

8) Water pump + Relay: A relay module is integrated into the system to control the operation of the water pump responsible for irrigation. Since microcontrollers like the ESP32 operate at low voltage and current levels, they cannot directly power high-voltage devices such as water pumps. The relay acts as an electrically operated switch, allowing the microcontroller to safely turn the pump on or off by controlling a separate high-power circuit. Using a relay ensures safe isolation between the low-voltage control circuitry and the high-voltage pump, protecting the system and users from electrical hazards. It also enables automated irrigation based on sensor readings, improving water efficiency and crop health by delivering water precisely when needed without manual intervention.

V. RESULTS AND DISCUSSION

A. Prototype

The prototype of the smart cucumber plant monitoring system was developed using modular components connected to an ESP32 shield to facilitate flexible testing and iteration. The core hardware includes the ESP32 microcontroller, soil moisture sensor, DHT22 temperature and humidity sensor, TDS sensor paired with a water temperature sensor for EC measurement, an ultrasonic sensor for water level monitoring, and a relay module to control the water pump. ESP32 collects data from these sensors and transmits real-time information to the Blynk dashboard for remote monitoring while logging data to google sheets.

The relay enables automated irrigation based on soil moisture and water availability data. This modular setup allows for efficient integration, testing, and refinement of the system components to ensure reliable environmental monitoring and irrigation control tailored for cucumber cultivation.

Figure 10: Hardware Prototype

B. Final Prototype

After testing the functionality of the model, the components were securely mounted on a project board or enclosed for stability and reduced interference.

The microcontroller was loaded with the final firmware for smooth operation, and all sensor inputs were neatly arranged. This compact cucumber plant monitoring system provides real-time data, updates both locally and remotely, and runs autonomously.

Figure 11: Final Prototype

The TDS (Total Dissolved Solids) sensor and the DS18B20 water temperature sensor are used together to monitor the quality of the irrigation or nutrient solution. The TDS sensor measures the electrical conductivity (EC) of the water, which indicates the concentration of dissolved substances such as salts, minerals, and nutrients. Monitoring TDS levels helps ensure that the plants are receiving an appropriate amount of nutrients. The analog signal from the TDS sensor is sampled ten times and averaged. The resulting raw value is then converted to voltage using the formula:

$$Voltage = \frac{Analog\ Value\ \times VREF}{4095.0}$$

However, the accuracy of the TDS sensor is affected by water temperature, as electrical conductivity varies with temperature. To account for this, the DS18B20 digital temperature sensor is used to measure the current water temperature. This temperature value is then used to apply compensation to the EC reading, ensuring that the calculated TDS value reflects what it would be at the standard reference temperature of 25°C using this formula:

$$\begin{aligned} &CompensationCoefficient \\ &\quad = 1 + 0.02 \times (Temperature - 25.0) \end{aligned}$$

The measured voltage is then adjusted:

$$CompensatedVoltage = \frac{Voltage}{CompensationCoeficcient}$$

The compensated voltage is used to compute EC using a third-degree polynomial curve fitting derived from empirical calibration data:

$$\begin{aligned} EC = {} & 133.42 \times (CompensatedVoltage)^3 \\ & - 255.86 \times (CompensatedVoltage)^2 \\ & + 857.39 \times CompensatedVoltage \end{aligned}$$

Together, the TDS sensor and water temperature sensor provide reliable and accurate information about the nutrient content of the water, enabling better decision-making for irrigation and nutrient management in the plant monitoring system.

The water level in the tank is determined using an ultrasonic sensor, which operates by emitting a short ultrasonic pulse from the trigger pin and measuring the time it takes for the echo to return to the echo pin after bouncing off the surface of the water. This time duration is then used to calculate the distance between the sensor and the water surface.
The duration of the echo is first captured in microseconds. This value is then converted into distance in centimeters using the formula:

$$Distance(CM) = \frac{Duration\ \times 0.034}{2}$$

To determine the water level as a percentage, the measured distance is mapped within a predefined range, typically assuming the maximum tank depth is 20 cm. The map() function in the code translates the measured distance into a percentage scale. The result is then subtracted from 20 to convert the distance into a water level percentage using formula:

$$Water\ Level\ (\%) = 20 - map(Distance, 0,20,0,20)$$

The soil moisture sensor outputs an analog voltage that varies depending on the moisture content of the soil. This analog signal is read by the ESP32 using the analogRead() function, which returns a value between 0 and 4095. Referring to the formula:

$$Soil\ Moisture\ (\%) = 100 - map(analog\ value, 0,4095,0,100)$$

By subtracting the mapped value from 100, the system inverts the scale to express the result in terms of moisture content rather than dryness. This gives a more intuitive output where higher percentages represent wetter soil conditions.

These calculations are updated every second using the BlynkTimer function. The results are simultaneously:

- Display real-time values from the sensors to Blynk Dashboard Depending on the BlynkTimer function.
- Uploaded to the Blynk dashboard via virtual pins
- Stored in Google Sheets for data logging

Users can view real-time data via their mobile devices using the Blynk app, which makes the system practical for plant health management.

Figure 12: Readings from sensors sent to Blynk

Figure 12: Data on Blynk mobile

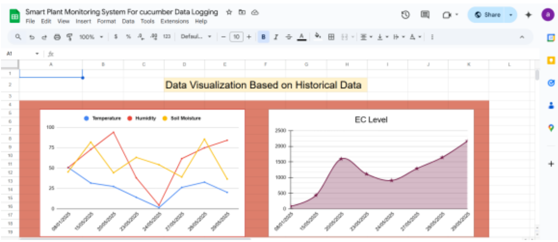


Figure 13: Data Visualization Google Sheets

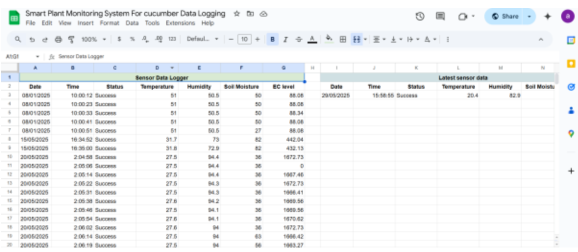


Figure 14: Data Logging Google Sheets

The system was subjected to various load conditions to test its performance and accuracy. The TDS and Water temperature sensor was needed to calibrate a few times to get its reading as accurate as possible. The ultrasonic sensor was also tested to ensure the correct tank levels. Soil moisture sensor is calibrated to measure the right amount of moisture of soil.

The Blynk dashboard proved to be responsive, reflecting real-time updates without noticeable lag. Users were able to monitor:

- The Environment Temperature & Ambient Humidity.
- Water Level in the water tank.
- EC quantities in the water tank.
- Refer to EC levels based on the plant growth stages.
- The soil moisture percentage.

These values were consistent across tests, suggesting reliable sensor calibration and effective data processing.

Strengths of the system include:

- Real-time monitoring of environmental sensors
- Local and cloud-based display for flexibility.
- Persistent data logging through Google sheets.
- Automation controlled water pump through Blynk.
- Control PIR motion sensor using Blynk.
- Notification-enabled Security system

Limitations:

- Absence of automated mixture of fertilizer.
- PIR motion sensor detection range is limited.
- Estimated battery level is not stated in the system.
-

Despite these limitations, the system fulfilled its core goal of providing users with visibility into their Cucumber plant monitoring system, which can support behavioural changes for improved plant growth and efficiency.

VI. CONCLUSION

In conclusion, this project focused on developing a Smart Plant Monitoring System for Cucumber using the ESP32 microcontroller, integrated with multiple sensors including DHT22 (temperature and humidity), PIR (motion), ultrasonic sensor (water level), soil moisture sensor, TDS (Total Dissolved Solids), and a water temperature sensor. The system not only gathers environmental data but also provides automation, such as triggering a water pump via a relay when soil moisture is low and transmits data in real time to both Blynk and Google Sheets for visualization and storage.

The successful integration of hardware and software demonstrated the potential of IoT systems in automating agricultural and environmental monitoring tasks. Sensor data collection, real-time alerts, data logging, and remote-control functionalities were all achieved as planned.

REFERENCES

[1] [1] K. Abouelmehdi, K. Elhattab, and A. El Moutaouakkil, "Smart Agriculture Monitoring System using Clean Energy," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 5, 2022. [Online]. Available: https://doi.org/10.14569/IJACSA.2022.0130544

[2] [2] Ada, *PIR Motion Sensor*, Adafruit Industries, 2023.

[3] [3] ANONIM, "5V Single-Channel Relay Module - Pin Diagram, Specifications, Applications, Working," *Components101*, 2020. [Online]. Available: https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet

[4] [4] A. Supriyanto and F. Fathurrahmani, "The prototype of the Greenhouse Smart Control and Monitoring System in Hydroponic Plants," *Digital Zone: J. Teknol. Inf. dan Komun.*, vol. 10, no. 2, 2019. [Online]. Available: https://doi.org/10.31849/digitalzone.v10i2.3265

[5] [5] M. B. I. Astutiningtyas, M. M. Nugraheni, and Suyoto, "Automatic Plants Watering System for Small Garden," *Int. J. Interact. Mobile Technol.*, vol. 15, no. 2, 2021. [Online]. Available: https://doi.org/10.3991/ijim.v15i02.12803

[6] [6] P. Kirci, E. Ozturk, and Y. Celik, "A Novel Approach for Monitoring of Smart Greenhouse and Flowerpot Parameters and Detection of Plant Growth with Sensors," *Agriculture (Switzerland)*, vol. 12, no. 10, 2022. [Online]. Available: https://doi.org/10.3390/agriculture12101705

[7] [7] Malvinos, *5V Active Buzzer*, Data Sheet, 91, 2006.

[8] [8] A. Q. Mohabuth and D. Nem, "An IoT-Based Model for Monitoring Plant Growth in Greenhouses," *J. Inf. Syst. Informatics*, vol. 5, no. 2, 2023. [Online]. Available: https://doi.org/10.51519/journalisi.v5i2.489

[9] [9] A. Mohanty and S. Roy, "Smart Plant Monitoring System Using BLYNK Application," in *Lect. Notes Netw. Syst.*, vol. 165, 2021. [Online]. Available: https://doi.org/10.1007/978-981-15-9433-5_19

[10] [10] V. R. Pathmudi, N. Khatri, S. Kumar, A. S. H. Abdul-Qawy, and A. K. Vyas, "A systematic review of IoT technologies and their constituents for smart and sustainable agriculture applications," *Sci. Afr.*, vol. 19, 2023. [Online]. Available: https://doi.org/10.1016/j.sciaf.2023.e01577

[11] [11] G. P. Pereira, M. Z. Chaari, and F. Daroge, "IoT-Enabled Smart Drip Irrigation System Using ESP32," *Internet Things*, vol. 4, no. 3, 2023. [Online]. Available: https://doi.org/10.3390/iot4030012

[12] [12] P. Rathod, S. Karambelkar, R. Khaire, M. Dhavale, and P. Raut, "Automated Water Pump System," *Int. J. Eng. Technol. Manag. Sci.*, vol. 7, no. 3, 2023. [Online]. Available: https://doi.org/10.46647/ijetms.2023.v07i03.032

[13] [13] S. D. Pawar and D. D. Rao, "Smart Plant Monitoring System for Plant Fitness Using IoT," *J. Pharm. Negat. Results*, 2022. [Online]. Available: https://doi.org/10.47750/pnr.2022.13.s06.102

[14] [14] "Smart Automated Plant Monitoring System Using Blynk Application," *Int. Res. J. Mod. Eng. Technol. Sci.*, 2023. [Online]. Available: https://doi.org/10.56726/irjmets43727

[15] [15] M. S. Novelan, "Monitoring System for Temperature and Humidity Measurement with DHT11 Sensor Using NodeMCU," *Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 10, 2020.

[16] [16] Tech Explorations, *What is Blynk?*, Tech Explorations, 2022.

[17] [17] Xukyo, "Overview of the NodeMCU ESP32 microcontroller," *AranaCorp*, Feb. 18, 2024. [Online]. Available: https://www.aranacorp.com/en/__trashed/

[18] [18] L. Yu, W. Gao, R. R. Shamshiri, S. Tao, Y. Ren, Y. Zhang, and G. Su, "Review of research progress on soil moisture sensor technology," *Int. J. Agric. Biol. Eng.*, vol. 14, no. 4, 2021. [Online]. Available: https://doi.org/10.25165/j.ijabe.20211404.6404

[19] [19] N. Zhang, F. He, F. Yuan, L. Li, Y. Li, and H. Li, "Development and Performance Evaluation of an Infrared Rain Sensor," *Procedia Comput. Sci.*, vol. 228, 2023. [Online]. Available: https://doi.org/10.1016/j.procs.2023.11.079