INTERNET OF THINGS (IoT)-BASED SMART GARDEN AND AUTOMATIC IRRIGATION SYSTEM

MOHAMAD HARITH AIZAT BIN SUHAILI

BACHELOR OF COMPUTER SCIENCE
(COMPUTER SYSTEM & NETWORKING
WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Mohamad Harith Aizat Bin Suhaili

Date of Birth

Title : Internet of Thing (IoT)-Based Smart Garden and Automatic Irrigation System

Academic Session : 2022 / 2023

I declare that this thesis is classified as:

- ☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- ☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- ☑ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_______________________ (Student's Signature)

_______________________ (Supervisor's Signature)

Mohamad Harith Aizat Bin Suhaili
Date: 10 July 2023

Ts Dr. Norsyahidatul Nadiah Binti Ismail
Date: 10 July 2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

____
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of Bachelor Computer Science (Computer System & Networking) with Honours.

(Supervisor's Signature)

Full Name : Ts. Dr. Nor Syahidatul Nadiah Binti Ismail

Position : Senior Lecturer

Date : 10 July 2023

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student's Signature)

Full Name : MOHAMAD HARITH AIZAT BIN SUHAILI

ID Number : CA20150

Date : 10 July 2023

INTERNET OF THING (IoT)-BASED
SMART GARDEN
AND AUTOMATIC IRRIGATION SYSTEM

MOHAMAD HARITH AIZAT BIN SUHAILI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor Of Computer Science
(Computer System & Networking) With Honours

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JULY 2023

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Ts. Dr. Nor Syahidatul Nadiah Binti Ismail for their guidance, support, and encouragement throughout the course of this project. Their expertise and knowledge have been invaluable in shaping the direction of this project and helping me to achieve my goals.

I would also like to thank anyone and PETAKOM who provided support or assistance. Their contributions have been greatly appreciated and have made this project a success.

I would also like to acknowledge my classmates and colleagues for their support, feedback, and contributions to the project. I would like to acknowledge my family and friends for their unwavering support and encouragement throughout this project and my academic journey.

Finally, I would like to show my gratitude to the project coordinator Dr. Danakorn Nincarean A/L Eh Phon for coordinate my project by giving guidance and encouragement for me to complete the proposal. Moreover, the guidance in proposal writing enables me to understand the correct format in writing the proposal.

Thank you all for your help and support.

ABSTRAK

Karbon dioksida, air, dan cahaya matahari adalah tiga elemen penting untuk tumbuhan. Air sangat penting untuk pengangkutan nutrien dan proses fotosintesis. Air yang tidak mencukupi boleh mengakibatkan kematian tumbuhan, manakala air yang berlebihan boleh menyebabkan akar kelemasan dan pembusukan. Sistem sedia ada tidak mempunyai sistem amaran untuk memberitahu pengguna, dan ia juga tidak menyediakan data masa nyata yang tepat mengenai taman. Oleh itu, disebabkan kelemahan sistem sedia ada, tesis mencadangkan Taman Pintar & Sistem Pengairan Automatik berasaskan Internet Of Things (IoT). Dalam sistem yang dicadangkan, ESP32 adalah papan utama bersama dengan tiga sensor iaitu sensor kelembapan tanah, sensor suhu & kelembapan, dan sensor pergerakan PIR. Objektif sistem yang dicadangkan ini adalah untuk menggunakan peranti yang disambungkan ke internet untuk memantau dan mengawal pelbagai aspek taman, termasuk kelembapan tanah, suhu & kelembapan, dan pergerakan. Matlamatnya adalah untuk mengautomasikan dan mengoptimumkan proses berkebun, menghasilkan tumbuhan yang lebih sihat dan penggunaan sumber yang lebih cekap. Selain itu, ia menyediakan keupayaan pemantauan dan kawalan jauh untuk pengguna. Pembangunan aplikasi pantas digunakan sebagai prinsip panduan untuk melengkapkan sistem yang dicadangkan, membolehkan pembangunan yang cepat dan cekap dengan keupayaan untuk membuat perubahan dan pelarasan berdasarkan maklum balas pengguna. Data dikumpulkan dan boleh diakses melalui aplikasi Blynk, dan jika kelembapan tanah jatuh di bawah paras tertentu, pam air akan diaktifkan secara automatik. Sistem ini juga memantau tahap kelembapan dan suhu dan menghidupkan kipas untuk menyejukkan mengikut keperluan. Juga, sistem akan mengesan gerakan dan akan memberi amaran kepada pengguna mengenai penceroboh. Berdasarkan Internet Of Things (IoT), sistem ini menggunakan aplikasi ESP32 dan Blynk untuk Pertukaran Data dan komunikasi dengan aplikasi telefon pintar. Kesimpulannya, sistem yang dicadangkan bertujuan untuk memenuhi objektifnya dan meningkatkan pengalaman berkebun dengan ketara dengan menjadikannya lebih cekap, mudah dan didorong oleh data.

ABSTRACT

Carbon dioxide, water, and sunlight are three essential elements for plants. Water is crucial for nutrient transport and the process of photosynthesis. Insufficient water can result in plant death, while excessive water can lead to root suffocation and decay. The existing system lacks an alert system to notify users, and it also doesn't provide accurate, real-time data on gardens. Thus, due to the weakness of the existing system, the thesis proposes Internet of Things (IoT) – Based Smart Garden & Automatic Irrigation System. In the proposed system, ESP32 is the main board along with three sensors which are soil moisture sensor, temperature & humidity sensor, and PIR motion sensor. The objective of this proposed system is to use internet-connected devices to monitor and control various aspects of a garden, including soil moisture, temperature & humidity, and motion. The goal is to automate and optimize the gardening process, resulting in healthier plants and more efficient resource utilization. Additionally, it provides remote monitoring and control capabilities for the user. Rapid Application Development is used as a guiding principle to complete the proposed system, enabling quick and efficient development with the ability to make changes and adjustments based on user feedback. Data is collected and can be accessed via a Blynk application, and if the soil moisture falls below a certain level, the water pump will activate automatically. The system also monitors humidity and temperature levels and turns on the fan to cool down as needed. Also, the system will detect motion and will warn the user about the intruder. Based on the Internet of Things (IoT), the system utilizes the ESP32 and Blynk application for data exchange and communication with a smartphone application. Conclusion, the proposed system aims to meet its objectives and significantly improve the gardening experience by making it more efficient, convenient, and data driven.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

^{O}C	Degree Celsius
%	Percentage

LIST OF ABBREVIATIONS

IoT	Internet of Things
IDE	Integrated Development Environment
OLED	Organic Light-Emitting Diode
LCD	Light Emitting Diode
RAD	Rapid Application Development
SMS	Short Message Services
B.C	Before Century
GDP	Gross Domestic Product
GSM	Global System for Mobile Communication
GPIO	General-Purpose Input/Output
API	Application Programming Interface
PIR	Passive Infrared
NO	Normally Open
DC	Direct Current
Wi-Fi	Wireless Fidelity
USB	Universal Serial Busy

CHAPTER 1

INTRODUCTION

1.1 Introduction

Carbon dioxide, water, and sunlight are three essential elements for plants (Bonnie, 2022). Water is essential for plants because it transports nutrients from the soil to the entire plant. Plants require water to survive, and if there is insufficient water, the leaves and stems will die. It is essential for plant growth because it strengthens the plant stem and prevents leaf drops. Without water, photosynthesis cannot occur in plants. To develop properly and produce abundant fruit, plants require adequate but not excessive water.

It is commonly known that insufficient water will kill plants, however much water will also harm plants (Heather Rhoades, 2022).Plants will perish if there is an excess of water since they require oxygen to survive. Plants breathe through their roots, and if there is an excessive amount of water, the roots will not absorb gas to breathe. The leaves will become yellow, the roots will decay, and the soil will become green owing to the growth of algae if a plant receives an excessive amount of water.

The irrigation system was developed to solve the problem of insufficient water for plants and excess water for plants. Archaeologists discovered evidence of the first irrigation system in the Jordan Valley of the Middle East around 6000 B.C (Sojka et al., 2022). This demonstrates that plants have utilized the irrigation system for a very long time. Irrigation systems provide sufficient water to agricultural land directly and efficiently. Increased crop yields and healthy crop yields are among the advantages of using an irrigation system. Consistent distribution of nutrients, hydrogen, and oxygen to the tree's roots can be ensured by utilizing an irrigation system (Woofter, 2022).

1.2 Problem Statement

The purpose of an irrigation system is to provide water to plants. It is typically used in dry and rainless regions. Malaysia's economy depends heavily on agriculture. It contributes 7 to 12 percent to the Gross Domestic Product (GDP) and employs approximately 16 percent of the labour force (Raju Chellam, 2020).The following is a list of irrigation system issues:

i. Humans are unaware of the actual watering schedule for plants and the amount of water required to maintain soil moisture.

ii. When the user is on holiday or spends more time outdoors, no one will tend to the plant properly.

iii. It causes the soil to become infertile and harmful to plants, as well as a waste of water to plants.

iv. For poor people, excessive water consumption becomes a cost.

v. Manual irrigation system provides irregular water to the plants, which might result in excessive plant stress.

To tackle these issues, an IoT Smart Garden and Automatic Irrigation System will be implemented. The soil moisture level will be monitored and detected by a soil moisture sensor. It is a resistor that can function in wet environments. When the soil is dry, the moisture sensor will detect and the water pump will begin to dispense water to the plants and when the soil moisture has reached the specified level, the water pump will cease to dispense water to the plants.

Utilizing an IoT Smart Agricultural and Automatic Irrigation System allows plants to save water efficiently. Water used for plants will be used to prevent water waste (Obaideen et al., 2022). In addition to enhancing the yield of plants in seasonal agriculture, the farmer's income will also grow. Most significantly, it will conserve water for future generations. Even though 70 percent of the earth is water, there is a severe lack of clean water in most regions. It is crucial to conserve water to preserve clean water and preserve the environment (h2ouse, 2022).

1.3 Objective

To develop a solution that will increase the dependability of the current irrigation system while needing less manpower and human contact. The fundamental objective of this project is to provide farmers with autonomous irrigation systems that work based on the moisture content of the soil.

The objectives of this project are:

i. To study the existing work and current issue on smart garden and irrigation system.

ii. To develop an Internet of Things (IoT) Smart Garden system.

iii. To evaluate the effectiveness of the proposed system using User Acceptance Test.

1.4 Scope

The scope of this project:

i. Platform

- Smartphone apps, where smart gardens can be controlled and monitored using a smartphone app.
- Web-based interfaces, which smart garden can be controlled and monitored through web-based interface, which can be accessed from any device with internet access.

ii. Development / Functionality

- Plants are watered based on their specific needs, as determined by sensors that measure soil moisture.
- Gardeners can monitor and control their smart garden remotely, using a smartphone or other device.

- Smart gardens can collect soil moisture data and humidity & temperature help gardeners understand their plan's needs and optimize their gardening techniques.

iii. User

- Home gardeners, which these individuals who want to grow plants in their own home.

- Urban dwellers, where people living in cities or other urban areas with limited space for gardening can benefit from IoT smart gardens, as they allow for gardening in small areas or indoors.

- Educational institutions, where smart gardens can be used in schools, universities, and research institutions to teach students about gardening and lant sciences.

1.5 Thesis Organization

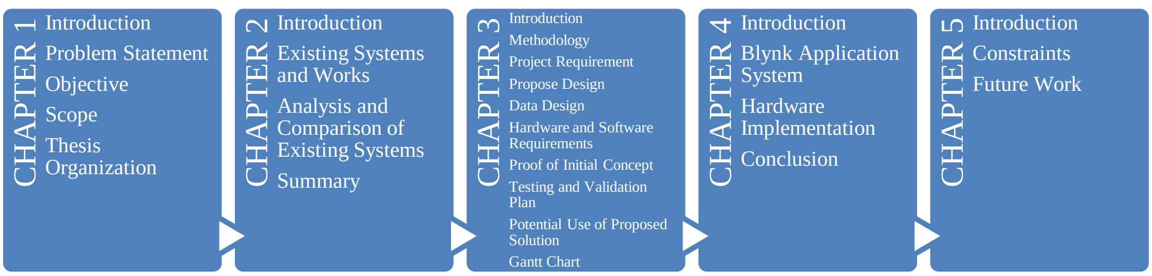


Figure 1.1 Thesis organization of the chapter

Based on the Figure 1.1 it shows about thesis organization of the chapter. Chapter 1 contains both the project's introduction and a discussion of the current issue. This chapter also addresses the problem description, project objectives, scope, and relevance. Chapter 2 contains a discussion topic about the literature view of an existing problem with an old type of irrigation system, as well as a comparison of the current project's solutions with those of a previous related project. Lastly, Chapter 3 demonstrates the discussion about the methodology is used during this study. In this topic, the study discusses the project requirement, design of the project, data design and testing from client. Chapter 4 will discuss the development stages, which include implementation and testing. Lastly, in Chapter 5, the project and its future potential will be discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Chapter 2 cover into a thorough examination of the relevant literature and background information related to the IoT Smart Garden project. Given the importance of research in creating a successful outcome, this chapter provides an overview of current systems, a comparison of those systems to the one proposed, details about the proposed system itself, and an exploration of the technology and tools used in its development.

2.2 Existing Systems and Works

2.2.1 Arduino Based Automatic Plant Irrigation System (System A)

The Arduino-based Automatic Plant Irrigation System, built with Arduino, ensures that plants receive the appropriate amount of water by using soil moisture sensors. It also sends updates to the user via text message. The system uses the sensor to determine the moisture level in the soil and activates the water pump when necessary to hydrate the plant. Once the optimal moisture level is reached, the pump will turn off.

The Automatic Plant Irrigation System notifies the user of the water pump status and soil moisture level through text message, sent via a GSM module, every time the pump turns on or off. This technology is ideal for agricultural, gardening, and home use, as it is completely automated and eliminates the need for human intervention (Gargee Sarkar et al., 2018). Based on the Figure 2.1, it shows the hardware setup for Arduino based Automatic Plant Irrigation System.

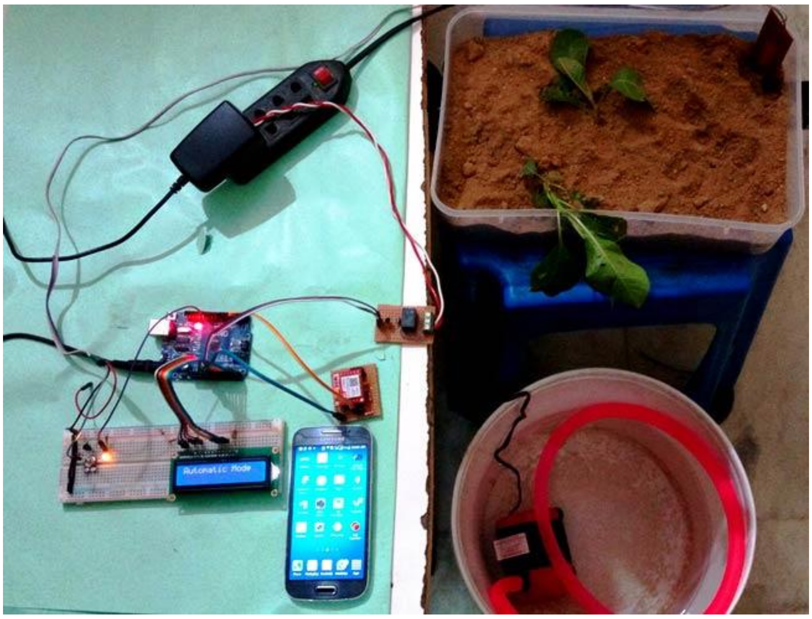


Figure 2.1 Hardware Setup for System A

The Arduino controls all aspects of the Smart Plant Irrigation System, as illustrated in the circuit diagram in Figure 2.2. The output of the soil moisture sensor is connected to digital pin D7 on the Arduino board. The sensor circuit includes an LED which acts as an indicator of soil moisture level: when the LED is on, it means there is enough moisture in the soil, and when it's off, it means the soil is dry. The system uses a TTL SIM800 GSM Module for sending SMS messages to the user.

The Smart Plant Irrigation System uses an LM317 Voltage Regulator to power the GSM Module. The ideal voltage range for the regulator is between 3.8v and 4.2v, and 3.8v is recommended. A 12V Relay is used to control a 220VAC water pump and is activated by a BC547 Transistor connected to digital pin 11 on the Arduino board. Additionally, an optional LCD display can be utilized to show status updates and messages. The control pins RS and EN of the LCD are connected to Arduino pins 14 and 15, respectively, and the data pins D4-D7 are connected to Arduino pins 16, 17, 18, and 19. The LCD is used in 4-bit mode and is controlled through the built-in LCD library in Arduino.

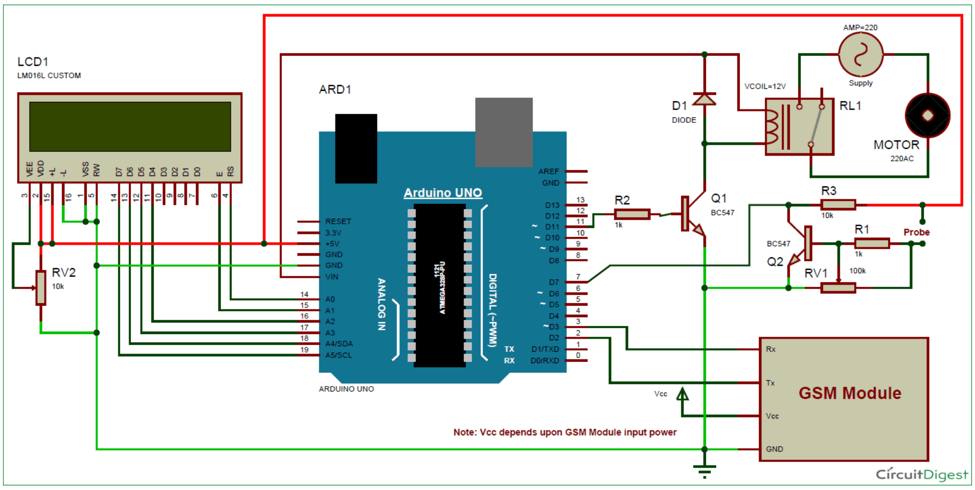


Figure 2.2 Circuit Diagram for System A

2.2.2 Smart Drip Irrigation System using Raspberry Pi and Arduino (System B)

The Smart Drip Irrigation System combines the use of Raspberry Pi and Arduino to conserve water by slowly dripping it onto the roots of plants through tubes and valves. To activate the irrigation, an email with the subject "run irrigation system for X minutes" is sent to a pre-configured email account. The email triggers one of the General-Purpose Input/Output (GPIO) connection ports on the Raspberry Pi to turn high. A Phyton software is used to receive the email and set the specified GPIO pin high for the given duration. The same software also sends status updates to the user through email.

The Smart Drip Irrigation System also utilizes Zigbee Modules to send and receive commands between the Raspberry Pi and the Arduino microcontrollers. When a low water level is detected by the sensor, it sends a signal from the Arduino to the Raspberry Pi, which then retransmits it to the Arduino to activate a relay-controlled solenoid valve. This system operates autonomously, regulating watering without the need for human intervention. The hardware setup for this system is illustrated in Figure 2.3, which shows the use of Raspberry Pi and Arduino (Swaroop et al., n.d.)

Figure 2.3 Hardware Setup for System B

Relay and ultrasonic distance sensor are controlled by Arduino. Based on the Figure 4 it shows circuit diagram of Smart Drip Irrigation System using Raspberry Pi and Arduino. One-channel relay board with 5-6V operation. On the relay board, there are three terminals labelled normally open (NO), usually closed (NC), and common (C). The common pin is linked to the NC pin while the relay is off and the NO pin when it is on. The input pin "INP" receives a logic high from the raspberry pi, which turns on the relay; therefore, common is connected to NO, which turns the device on until the relay is turned on. The relay's "VCC" and "GND" pins are respectively linked to a 5V power source and ground.

One end of the water pump is linked to the 240V AC supply, and the other end is attached to the relay board's NO pin. Then, two 30-liter water storage tanks and two 220V/50Hz water pumps are utilized for the water tank and pump. It requires 0.23A and 18W of current and power. Each water tank is equipped with an ultrasonic distance sensor for measuring water depth. Arduino receives a signal to open a solenoid valve as soon as the water level falls below a predetermined threshold. At that point, the water can be replaced.

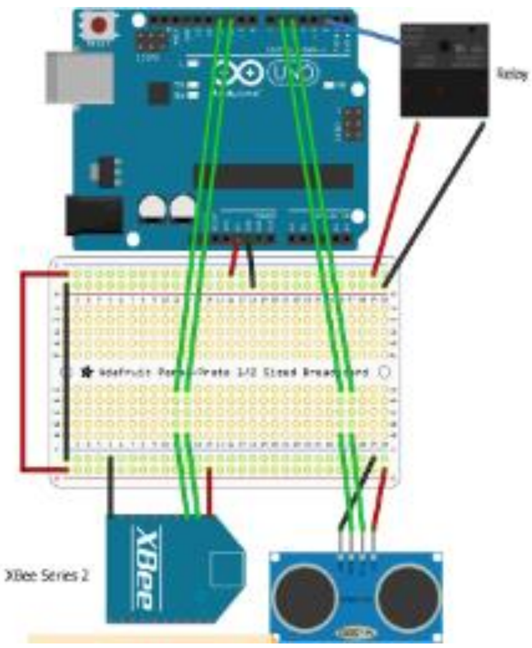


Figure 2.4 Circuit Diagram for System B

2.2.3 Internet of Thing (IoT) Enabled Smart Irrigation System for Mini Gardens (System C)

A wireless sensor node has been integrated into the irrigation system for greenhouse monitoring by using the Sensinode sensor platform. This system can be controlled using a GSM module, and all data can be accessed through a web-based interface. The system uses a combination of local weather report and soil moisture sensor to monitor outdoor plants. By inserting a soil moisture sensor into the soil, the system can monitor the soil moisture level and determine the likelihood of rain by utilizing a weather API.

The system will automatically water the plants as per the requirements when it's turned on, after that it will go into sleep mode and stop sensing. The system will wake up and perform sensing once every 24 hours. The system also uses cloud services to gather meteorological information and uses sensors to detect environmental factors like temperature and humidity.(Maheswari & Bhuvaneswari, 2021). Based on Figure 5, it shows the hardware setup for IoT Enabled Smart Irrigation System for Mini Gardens.

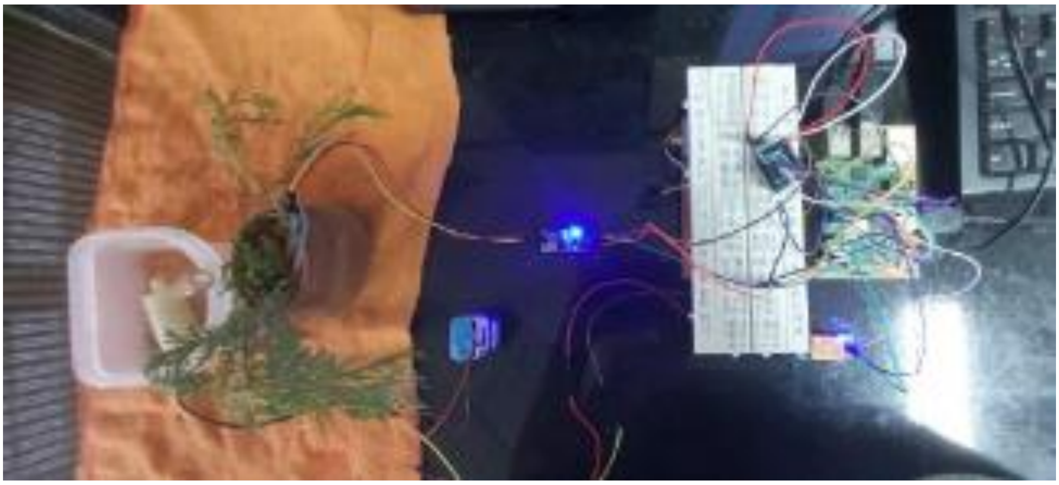


Figure 2.5 Hardware Setup for System C

This system is powered by a Raspberry Pi, which is used to control the sensor and gather weather data. The soil moisture sensor is crucial for obtaining information about the water content of the soil. It can determine the volumetric water content of the soil based on the soil's dielectric constant. The sensor supports both analog and digital outputs. The VCC pin is used to power the device, while the Analog Output (AO) pin provides access to the sensor's output voltage, which is proportional to the probe's resistance. The same signal can also be accessed via the Digital Output (DO) pin.

To gather the analog data from the soil moisture sensor, this project employs the use of the MCP3008 Analog-to-Digital Converter (ADC), which provides excellent performance and low power consumption in a compact form. This is necessary because the Raspberry Pi's General-Purpose Input Output (GPIO) does not have the capability to read analog signals. The project also uses OpenWeather, an online service that provides global meteorological data via an API. This includes current weather data, nowcasts, and historical weather data for any location. It also provides minute-by-minute forecasts for hyperlocal nowcasts for any place.

2.3 Analysis and Comparison of Existing Systems

2.3.1 Analysis of Comparison on Existing Systems

Table 2.1 presents an assessment of the chosen existing systems. The table provides a summary of the comparison between the three existing systems selected for the literature study and the proposed system. System A refer to ‘Arduino based Automatic Plant Irrigation

System', system B refer to 'Smart Drip Irrigation System using Raspberry Pi and Arduino', system C refer to 'IoT Enabled Smart Irrigation System for Mini Gardens' and lastly is system D which is the proposed system refer to 'IoT Smart Garden & Automatic Irrigation System'.

Table 2.1 Comparison on Existing Systems

FEATURES	SYSTEMS			
	System A	System B	System C	System D (Proposed System)
Sensor	- Soil Moisture Sensor	- Ultrasound Distance Sensor	- Soil Moisture Sensor	- Soil Moisture Sensor - DHT 22 Temperature & Humidity Sensor - PIR Motion Sensor
Hardware	- Arduino - GSM Module - Transistor BC547 - LCD Display - DC Water Pump	- Raspberry Pi - Arduino - Relay Board - Power Supply - Solenoid Valve - DC Water Pump	- Raspberry Pi - MCP3008 Analog-to-Digital Converter - DC Water Pump - GSM Module	- ESP32 - DC Water Pump - 2 Channel Relay Module - DC 5V Brushless Fan
Criteria	- Automatic watering system to the plant and updating the	- Smart drip irrigation system that can operate	- Watering system that can be controlled by GSM	- Smart watering system that can be monitor on

	user the soil moisture level and water pump status by sending massage to cell phone via GSM Module.	automatically using commands that send through email and depending on the current weather conditions.	Module and can be monitored on web platform.	the phone and automatically irrigate the plant according to the soil moisture level.
Plant	- No Plants	- Chili Tree - Flowers	- Curry Tree	- Chili Tree
Interfaces	Mobile Platform	Web Platform	Web Platform	Mobile Platform and Web Platform
Cloud-based	No	No	- Open WeatherMap API	- Blynk
Internet Dependency	No	Yes	Yes	Yes
Advantages of the system	- Low Power Consumption - Global Range - Easy To Operate	- Automates and controls the watering process without the need for human involvement.	- Simple smart watering system. - Using weather API which can predict the sign of rain. - Automates watering plants	- Run with the latest equipment with the project can operate smoothly. - It will ease the user to manage and monitor the plant.

			without man work	
Disadvantages of the system	- Not work in remote areas - Flexible To Run at Specific Intervals - Undetectable internal problems in motor - Communication delays	- Limitation of project design is that the failure of any part or device is not informed and must be tested manually.	- Using broad location access which it is not accurate rather than local location. - Did not have alert system to make user who is away get notified.	- Did not have weather prediction for the plant

In summary, all the systems using soil moisture sensor except System B only using ultrasound distance sensor. For the proposed system it added three more different sensors which soil moisture sensor, humidity and temperature sensor and PIR motion sensor. For the hardware equipment, only System A and B use Arduino but on proposed system it uses ESP32.

All the system equipped of DC water pump for watering the plant. Each system has different criteria which on system A the criteria is it automatically watering system to the plant and updating the user the soil moisture level and water pump status by sending massage to cell phone via GSM Module. For system B is smart drip irrigation system that can operate automatically using commands that send through email and depending on the current weather conditions.

Then, the criteria for system C are watering system that can be controlled by GSM Module and can be monitored on web platform. For the proposed system, the criteria are smart watering system that can be monitor on the phone and automatically irrigate the plant according to the soil moisture level.

Next, there is no plant that have been used in system A. For the system B the plant that used in the project are chili tree and some flowers. System C using curry tree for their project and proposed system using three types of plant which is parsley, chili tree and salad. There are two different types of interfaces that used were system A and system D use mobile platform. For system B and C using web platform. Moving to cloud-based that used for system C using Open Weather and proposed system using Blynk. All the system relies on internet except for system A.

2.3.2 Relevance of Comparison with Project Title

To compare the proposed system to the three current systems, the moisture sensor will be utilized to measure the soil moisture level of plant in the proposed system. On the proposed system there are three added parameter which is soil moisture sensor, humidity & temperature sensor, and PIR motion sensor. The present soil moisture level, temperature and humidity level of the plant can be monitored through the user's mobile device. If soil moisture falls below a specific level, the water pump will activate automatically.

Next, the temperature in the mini greenhouse can be controlled by the fan based on the readings from the humidity and temperature sensor. The ESP32 is compatible with the Arduino IDE software, which is used to connect the board to the Blynk application over a Wi-Fi connection. The proposed system will be built on the concept of the Internet of Things (IoT), making it a cloud-based system with internet-based communication networks.

2.4 Summary

In conclusion, a smart garden is well-suited for home usage, particularly in high-rise buildings with balconies, as it reduces the effort required for watering and monitoring the garden. Furthermore, the system can automatically detect the soil moisture level of plants and water them accordingly. Lastly, the smart garden incorporates IoT technology, enabling the transmission of real-time data over the internet. This significantly reduces the time needed for plant monitoring, as the data is displayed on a smartphone. Users can simply verify the application's real-time data to stay updated on the information.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will cover the project's methodology and describe the project's development workflow. The system development technique is a framework for structuring, planning, and developing the system in a methodical manner. The methodology that suitable for this project is Rapid Application Development (RAD) model which is often used for small to medium projects that have a time constraint.

3.2 Methodology

Rapid Application Development (RAD) is a condensed development process that emphasizes prototyping and enables the developer to rapidly adapt to changing needs and conduct several iterations without starting from scratch each time. This ensures that the final product delivered within the time (*SDLC - RAD Model*, n.d.).

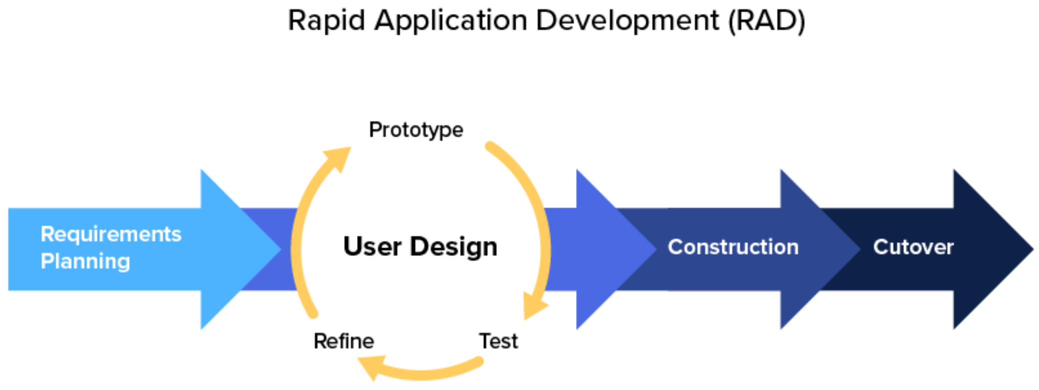


Figure 3.1 Rapid Application Development

The RAD model comprises of four stages: Defining project requirements, Planning, Design, Construction, and Cutover.

3.2.1 Requirement Planning Phase

During this step of requirement planning, the project's objectives, expectations, deadlines, and potential problems must be specified and identified. Before entering the design phase, for instance, this project's system purpose, functional, and non-functional requirements will be established. Requirements gathering is the initial phase before the development phase, and it ensures a deeper understanding of the system.

The proposed project is to create an IoT-based Smart Garden & Automatic Irrigation System using ESP32. The system will monitor various parameters of the soil such as soil moisture, humidity & temperature and motion sensor then will be sending the results in real-time to the cloud through a network. The target users for this system are housewives or anyone who has an interest in gardening, as it allows them to monitor the status of their garden in real-time.

The requirement phase is an important step in the development of an IoT smart garden. During this phase, data is collected to understand the needs, goals, and constraints of the users and the environment. One way to collect this data is by using a Google Form. A Google Form is a tool that allows users to create a survey or questionnaire and share it with others. The responses can be collected and analysed in a spreadsheet, making it easy to organize and analyse the data. A survey has been conducted for this project to study time spent on garden maintenance. This survey covers only one section, consisting of five questions. The questionnaire is on Appendix A.

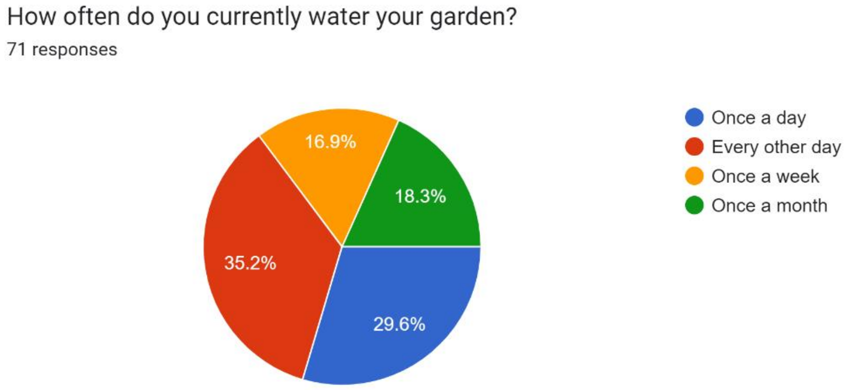


Figure 3.2 Survey Chart 1

From the Figure 3.2 "How often do you currently water your garden?", we can conclude that most of the respondent's water their garden every other day (35.2%), followed by once a day (29.6%). A smaller portion of respondents water their garden once a month (18.3%) and once a week (16.9%). This data suggests that most respondents water their garden regularly, with a significant portion watering it daily. This information can be useful for understanding the watering habits of the respondents and how often they need to maintain their garden. It can also provide insight into how much time the respondents currently spend on maintaining their garden, which can be useful for determining if an IoT-based smart garden and automatic irrigation system would be beneficial for them.

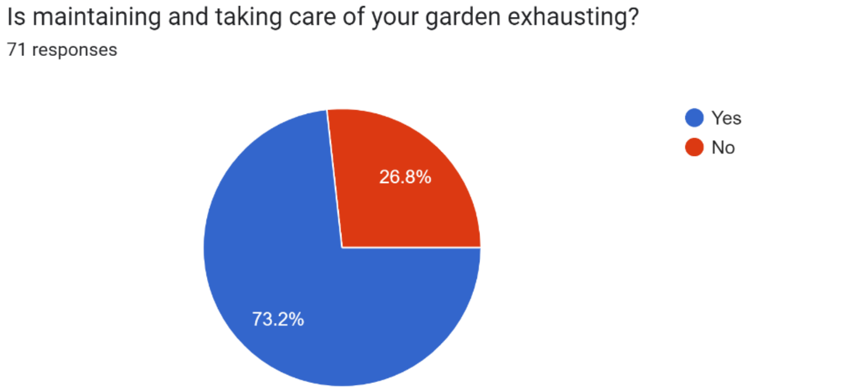


Figure 3.3 Survey Chart 2

From the Figure 3.3 "Is maintaining and taking care of your garden exhausting?", we can conclude that most of the respondents find maintaining and taking care of their garden to be exhausting (73.2%), while a smaller portion of respondents (26.8%) do not find it to be exhausting. This data suggests that for most respondents, maintaining their garden is an arduous task. This information can be useful for understanding the level of effort the respondents feel they need to put into maintaining their garden. It can also provide insight into how much of a burden maintaining their garden currently is for the respondents, and whether an IoT-based smart garden and automatic irrigation system would be beneficial for them in terms of reducing the time and effort they need to spend on maintaining their garden.

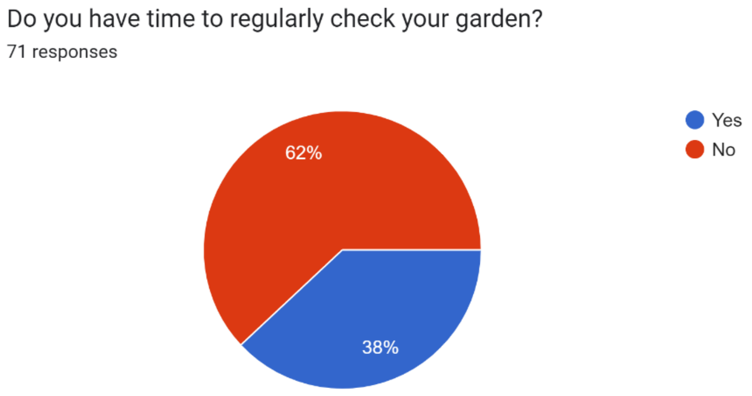


Figure 3.4 Survey Chart 3

From the Figure 3.4 "Do you have time to regularly check your garden?", we can conclude that most of the respondents have time to regularly check their garden (62%), while a smaller portion of respondents (38%) do not have time to regularly check their garden. This data suggests that for most respondents, they have time to monitor and maintain their garden, but for some of them, it might be a problem, which can lead to neglecting their garden. This information can be useful for understanding the availability and schedule of the respondents to monitor their garden. It can also provide insight into how much of a burden maintaining their garden currently is for the respondents, and whether an IoT-based smart garden and automatic irrigation system would be beneficial for them in terms of reducing the time they need to spend on monitoring and maintaining their garden.

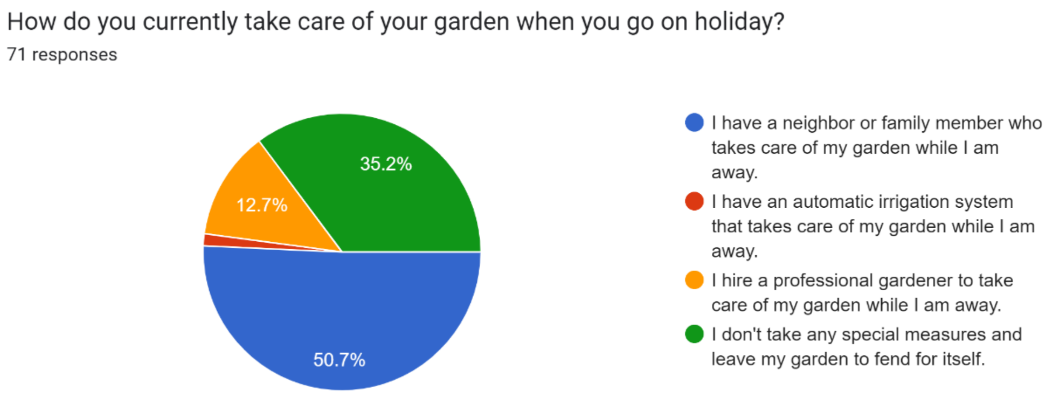


Figure 3.5 Survey Chart 4

From the Figure 3.5 "How do you currently take care of your garden when you go on holiday?", we can conclude that most of the respondents rely on a neighbour or family member to take care of their garden while they are away (50.7%), while a significant proportion of respondents don't take any special measures and leave their garden to fend for itself (32.5%). A smaller portion of respondents hire a professional gardener to take care of their garden while they are away (12.7%), and only a very small portion of respondents have an automatic irrigation system that takes care of their garden while they are away (1.4%). This data suggests that for most respondents, they do not have a proper solution or system in place to take care of their garden while they are away, this leads to neglecting their garden. This information can be useful for understanding the level of effort the respondents feel they need to put into maintaining their garden and how much of a burden maintaining their garden currently is for the respondents, and whether an IoT-based smart garden and automatic irrigation system would be beneficial for them in terms of reducing the time and effort they need to spend on monitoring and maintaining their garden when they are away.

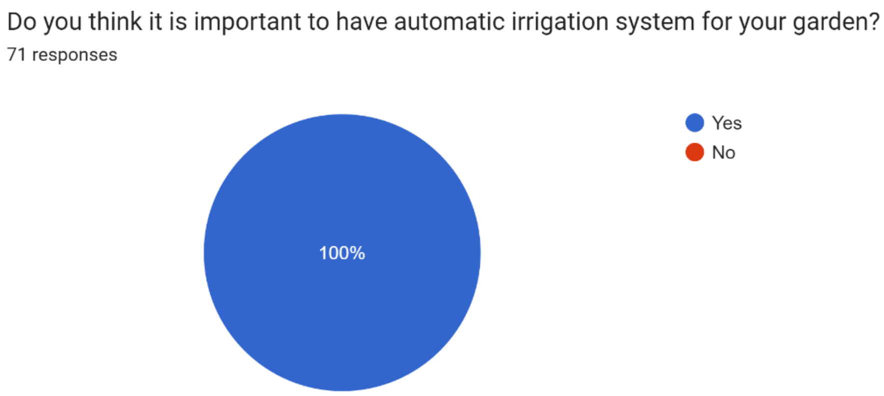


Figure 3.6 Survey Chart 5

From the Figure 3.6 "Do you think it is important to have automatic irrigation system for your garden?", we can conclude that all the respondents (100%) think it is important to have an automatic irrigation system for their garden. This data suggests that all the respondents believe that having an automatic irrigation system for their garden is important and it can help them to save time, effort, and water. This information can be useful for understanding the level of importance the respondents give to irrigation systems and the benefits they are expecting from it.

3.2.2 Design Phase

After identifying the system's requirements, the design phase will begin, during which system models will be created. This phase will include tasks such as system design and interface design. In system design, techniques such as flowchart, use case diagram, and context diagram are used to explain how the system interacts with the end user. The flowchart is used to show the process of reading water quality data, analysing the threshold level, sending alerts, and storing data. The use case diagram illustrates the user's interaction with the system, including the client and the administrator.

In the use case diagram, the users or actors can perform actions such as viewing reports, managing databases, assigning tasks, etc. The context diagram is intended to show the flow of data between the ESP32 microcontroller, the system, and the user, or vice versa. After that, a draft of the interface design will be created, primarily using storyboards to illustrate the navigation in the interface of the water monitoring system. During this step, the appropriate design strategy will also be chosen.

In addition to conducting a literature review, other existing related systems will be analysed to assist in the design phase of the proposed system and to generate new ideas. This is also an effective technique to improve the capabilities of the system. The current systems and the proposed system will be compared to enhance the proposed system.

3.2.3 Construction Phase

In the building phase, emphasis will be placed on system development. On the suggested system, stages such as development and testing would be implemented. Writing code for the intelligent garden is a technique for implementing the proposed system. Arduino IDE will be used to write the code for the microcontroller's operation. In this phase, any necessary changes or improvements to the system will also be made. The activities to be completed include system development, coding, and system testing.

3.2.4 Cutover Phase

This is the deployment phase, in which the development of the system is completed, and it is ready for launch. The smart garden will undergo testing to ensure its performance and functionality, this guarantee that the system meets the standards and aligns with the goals and scope of the smart garden. In this phase, the system will be evaluated using a soil sample containing different types of plants to determine if it meets the needs and objectives of the system.

In the phase focused on minimizing the error rate of the proposed system, actions such as code repair and enhancement will be performed. After testing is completed, the system is ready for launch, however, in this project, testing will be carried out on the system prototype.

3.3 Project Requirement

3.3.1 Functional Requirements

i. The system should be able to measure soil moisture and temperature of the garden.

ii. The system should be able to automatically water the plants based on the soil moisture level.

iii. The system should be able to be accessed and controlled remotely through a smartphone or computer.

iv. The system should be able to send alerts and notifications to the user regarding the status of the garden and any potential issues.

3.3.2 Non-Functional Requirements

i. The system should be easy to use and understand for the average user, with clear and intuitive interfaces and controls.

ii. The system should be able to perform its functions in a timely and efficient manner, without undue delays or errors.

iii. The system should be flexible and able to adapt to changing requirements and new technologies as they emerge.

3.3.3 Limitation

The limitation of the proposed system may not be able to support a wide variety of plant types and may only work with certain types of plants. In term of technical limitations, the system may have limitations on its ability to accurately monitor and control different environmental conditions, such as temperature and light. Besides, As the number of plants in the garden increases, the system may become less effective or require additional hardware to function properly.

3.3.4 Constraints

This project's constraint is that it will be impossible to collect information about the plant if the network connection becomes unreliable or is interrupted. Then, if any component or device failed, it would not be notified and would have to be manually tested. The project may be constrained by the availability of certain resources, such as sensors or software development kits.

3.4 Propose Design

3.4.1 Overall System Design

An IoT smart garden system is a network of interconnected devices and sensors that work together to monitor and control various aspects of a garden such as soil moisture, temperature humidity and motion. The overall system design includes several key components such as sensors, microcontroller, and actuators. Based on Figure 3.7 it shows the overall system design for IoT Smart Garden

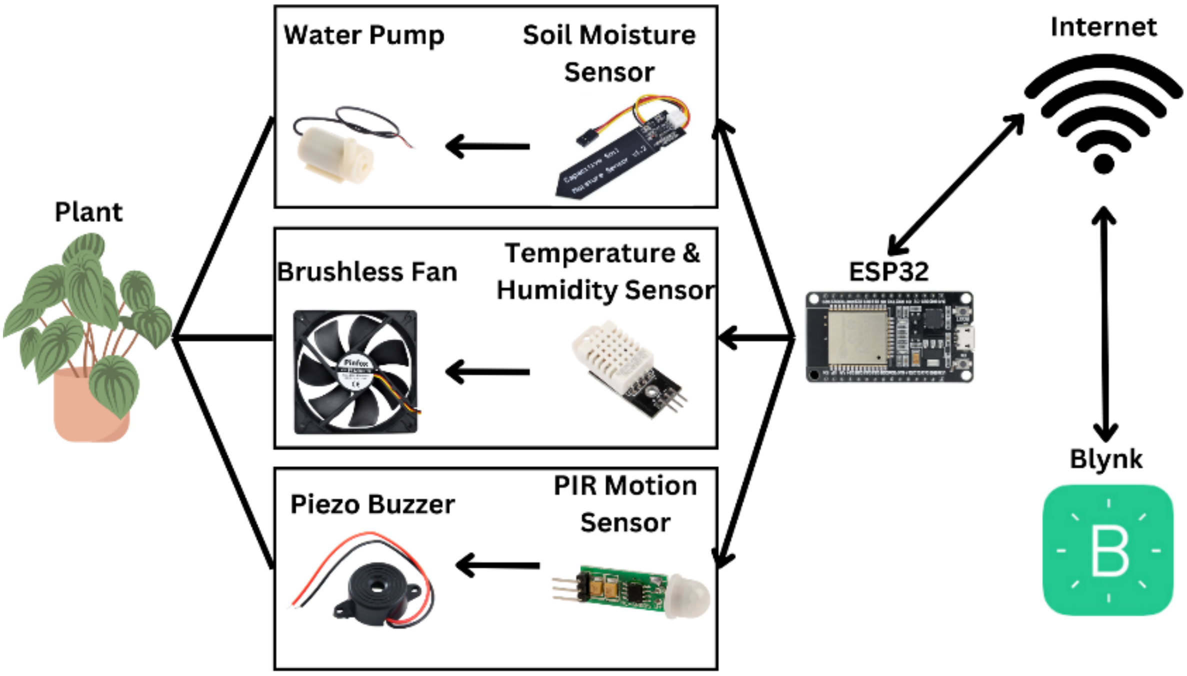


Figure 3.7 Overall System Design

3.4.2 Architecture Design for IoT Smart Garden

In this proposed system, the microcontroller used is the ESP32. As it is a Smart Garden and Automatic Irrigation System, it includes sensors such as the soil moisture sensor, the temperature & humidity sensor and PIR motion sensor. These sensors are connected to the ESP32 microcontroller. The microcontroller is connected to the Wi-Fi which enables the data to be sent and received to the cloud, and then to devices such as computers and smartphones. The user can view the real-time data by accessing it through a website or a smartphone application. The architecture design of this proposed system is shown in the Figure 3.8.

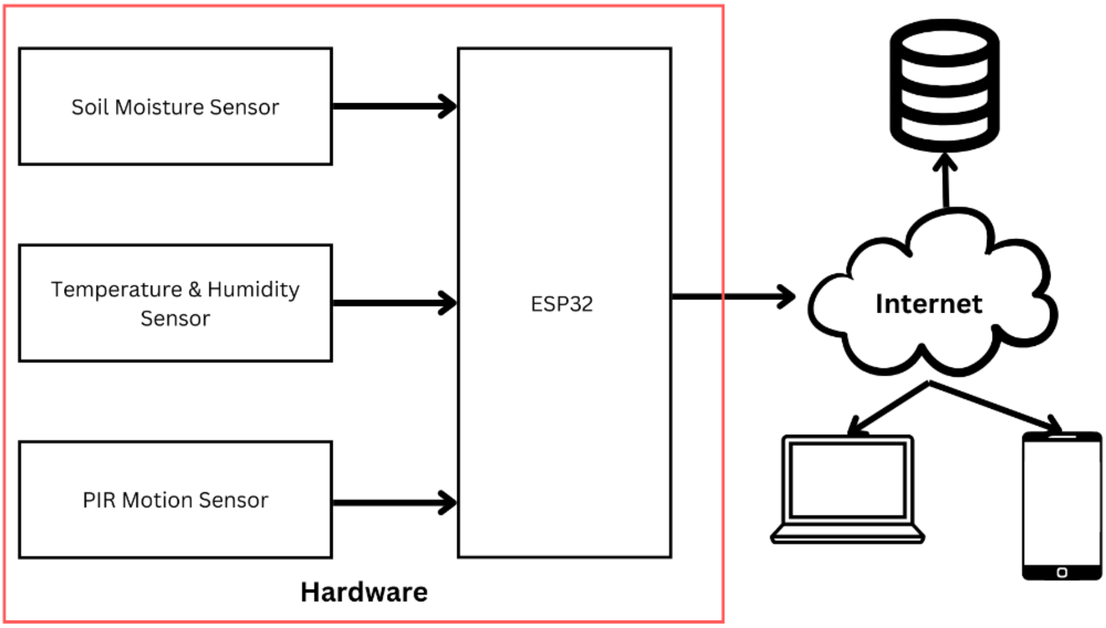


Figure 3.8 Architecture Design

3.4.3 Circuit Diagram for IoT Smart Garden

Figure 3.9 depicted circuit diagram is a representation of how various components of an IoT smart garden system can be connected using a ESP32 microcontroller as the central control unit.

- ESP32: The ESP32 is the microcontroller that controls and manages the entire system. It reads sensor data, controls the relay and water pump, and sends data to the Blynk application.
- Relay Module: The relay module is used to control the water pump. It acts as an interface between the ESP32 and the water pump, allowing the microcontroller to turn the pump on and off as needed.
- DHT 22 Temperature and Humidity Sensor: This sensor is used to measure the temperature and humidity in the garden. The ESP32 reads the sensor data and sends it to the Blynk application for display.
- Soil Moisture Sensor: This sensor is used to measure the moisture content in the soil. The ESP32 reads the sensor data and sends it to the Blynk application for display.

- Water Pump: The water pump is used to water the plants in the garden. It is controlled by the relay module, which is in turn controlled by the ESP32.

- PIR Motion Sensor: This sensor used for detecting motion or unauthorized access to a protected area.

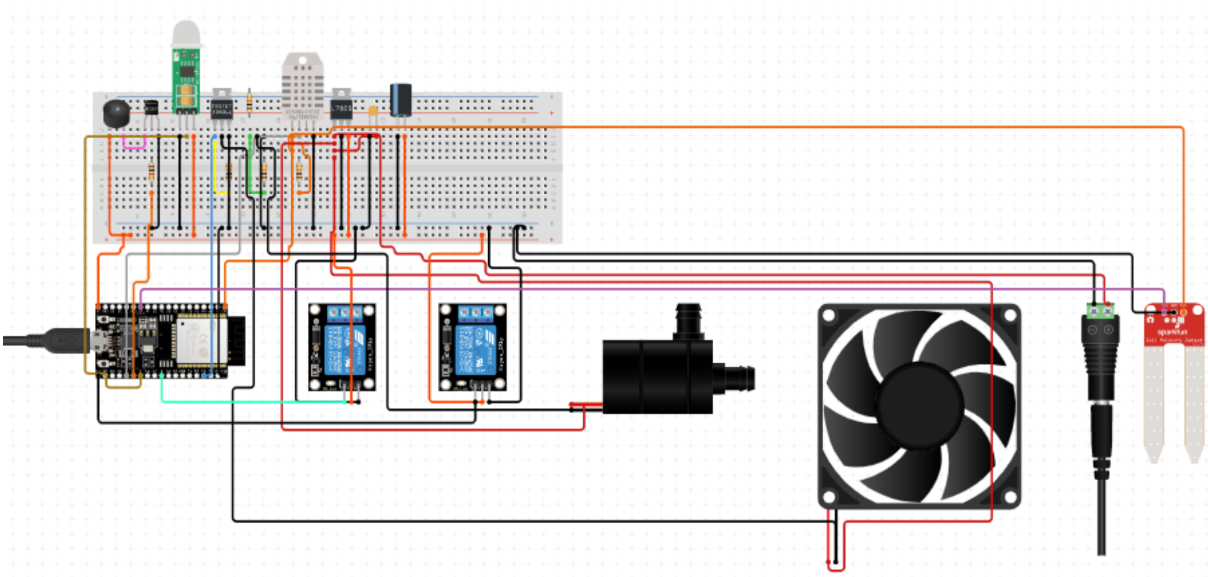


Figure 3.9 Circuit Diagram

3.4.4 Mint Plant

The mint plant, scientifically known as Mentha, is a herbaceous perennial plant that belongs to the Lamiaceae family. It is native to Europe, Asia, Africa, and North America but is widely cultivated around the world for its aromatic leaves and various culinary and medicinal uses. Mint plants prefer a slightly acidic to neutral soil pH ranging from 6.0 to 7.0. Mint plants thrive in loamy or sandy soil that is well-draining. Mint plants thrive in cool to moderately warm temperatures. The ideal temperature range for mint plants is typically between 15°C and 21°C. They can tolerate temperatures slightly outside this range, but prolonged exposure to high temperatures above 27°C can cause stress and affect the plant's growth.

Mint plants prefer moderate humidity levels. They generally thrive in humidity ranges between 40% and 60%. Higher humidity levels can promote better growth and help prevent the leaves from drying out. However, mint plants can still tolerate lower humidity levels, especially if provided with adequate watering and regular misting to maintain moisture around the plant according to the Table 3.1 is the data of Mint Plant.

Table 3.1 Mint Plant Data

Parameter	Threshold	
	Minimum	Maximum
Soil Moisture	60%	80%
Temperature	15^{o}C	21^{o}C
Humidity	40%	60%
Soil Type	Garden soil, perlite or coarse sand and compost or well-rotted manure	

3.4.5 Flowchart

3.4.5.1 General Flowchart

The proposed system is an IoT smart garden with an automatic irrigation system is a type of gardening system that uses internet-connected devices to monitor and care for plants. Figure 3.10 shows about the general flowchart for the proposed system.

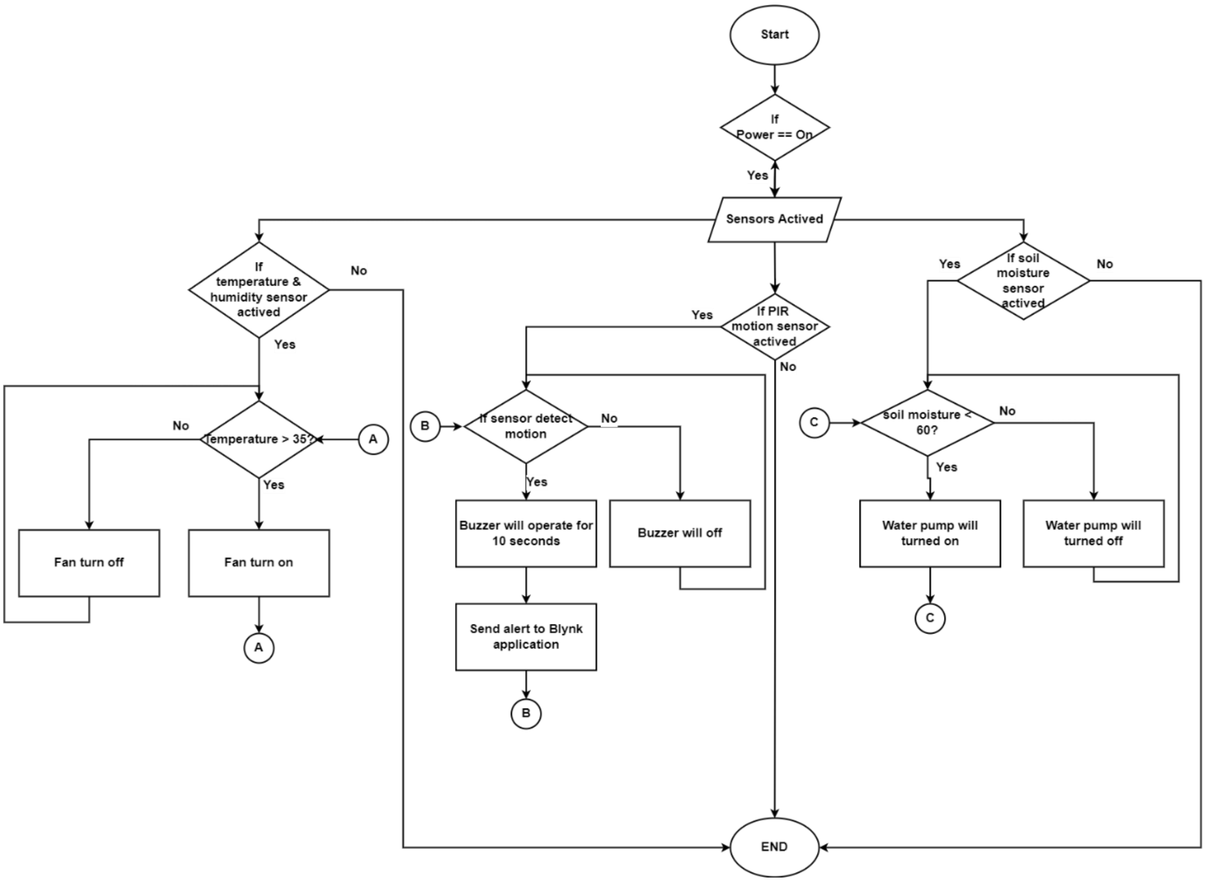


Figure 3.10 General Flowchart

3.4.5.2 Hardware Flowchart

The hardware used in this system includes ESP32, Soil Moisture Sensor, Temperature & Humidity Sensor, and Water Pump. The hardware is needed to irrigate the plant using the water pump, in which the soil moisture sensor will detect the moisture of soil, analyse the soil condition. The sensors will be sending the signal to ESP32 for data processing, resulting in automatic irrigation and real time data for plant.

The data will be processed, and the data transmitted from ESP32 to Blynk application. Water pump will start operated when the soil moisture detects the soil moisture level is below 60% and will be stop when it over 79%. To allow data transmission, the ESP32 must be turned on, and the power supply must be available, as the sensor require power to function. The flowchart for the hardware is shown in Figure 3.11 below.

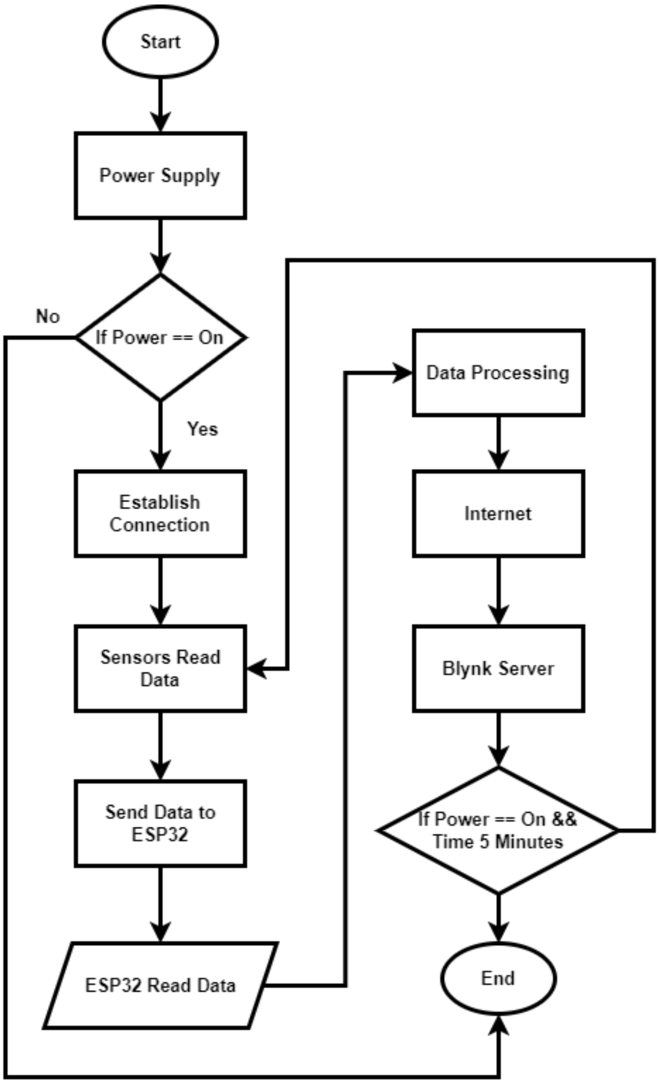


Figure 3.11 Hardware Flowchart

3.4.5.3 System Flowchart

The users need to login before accessed to the system. The valid email and password need to be entered. User can monitor the garden status. The email and password will be used for verification. If the email and password match with the database, then user is being authorized to access the system. Based on the Figure 3.12 it shows the flowchart for login interface.

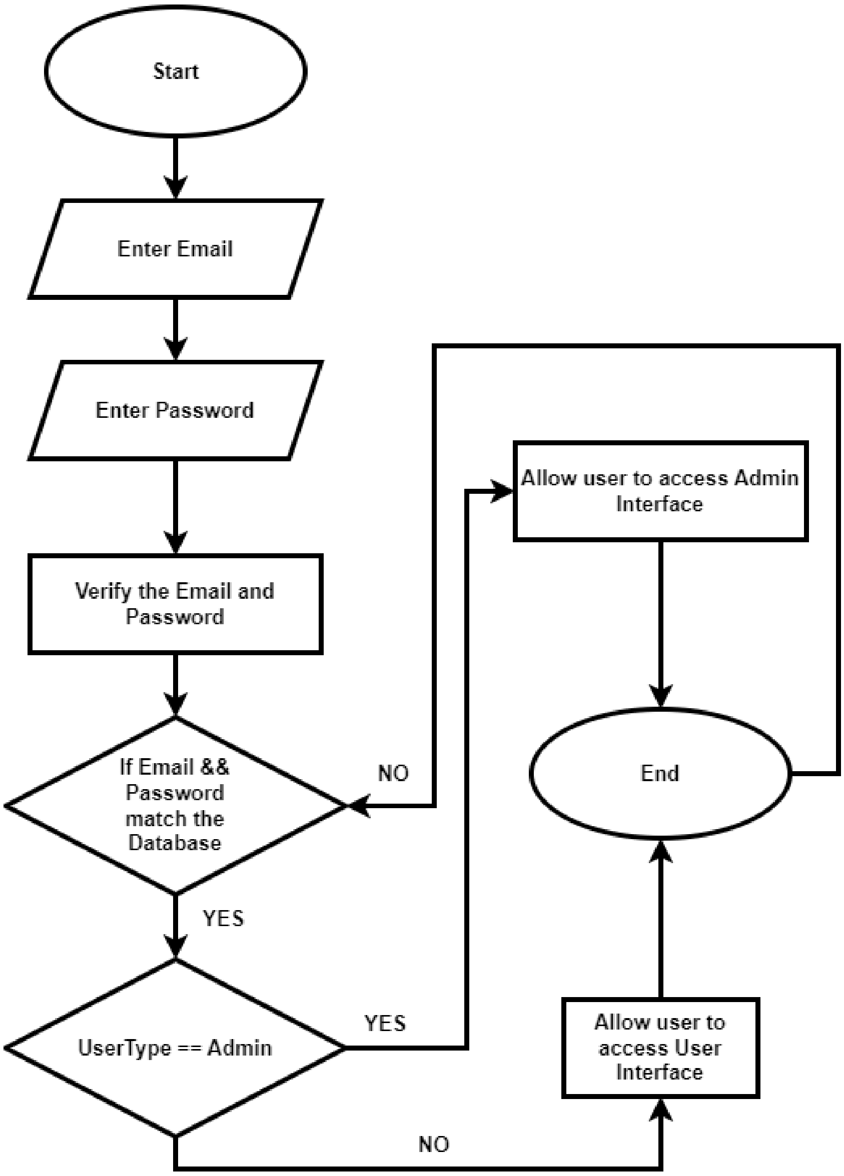


Figure 3.12 System Flowchart

3.4.5.4 Subprocess Soil Moisture Flowchart

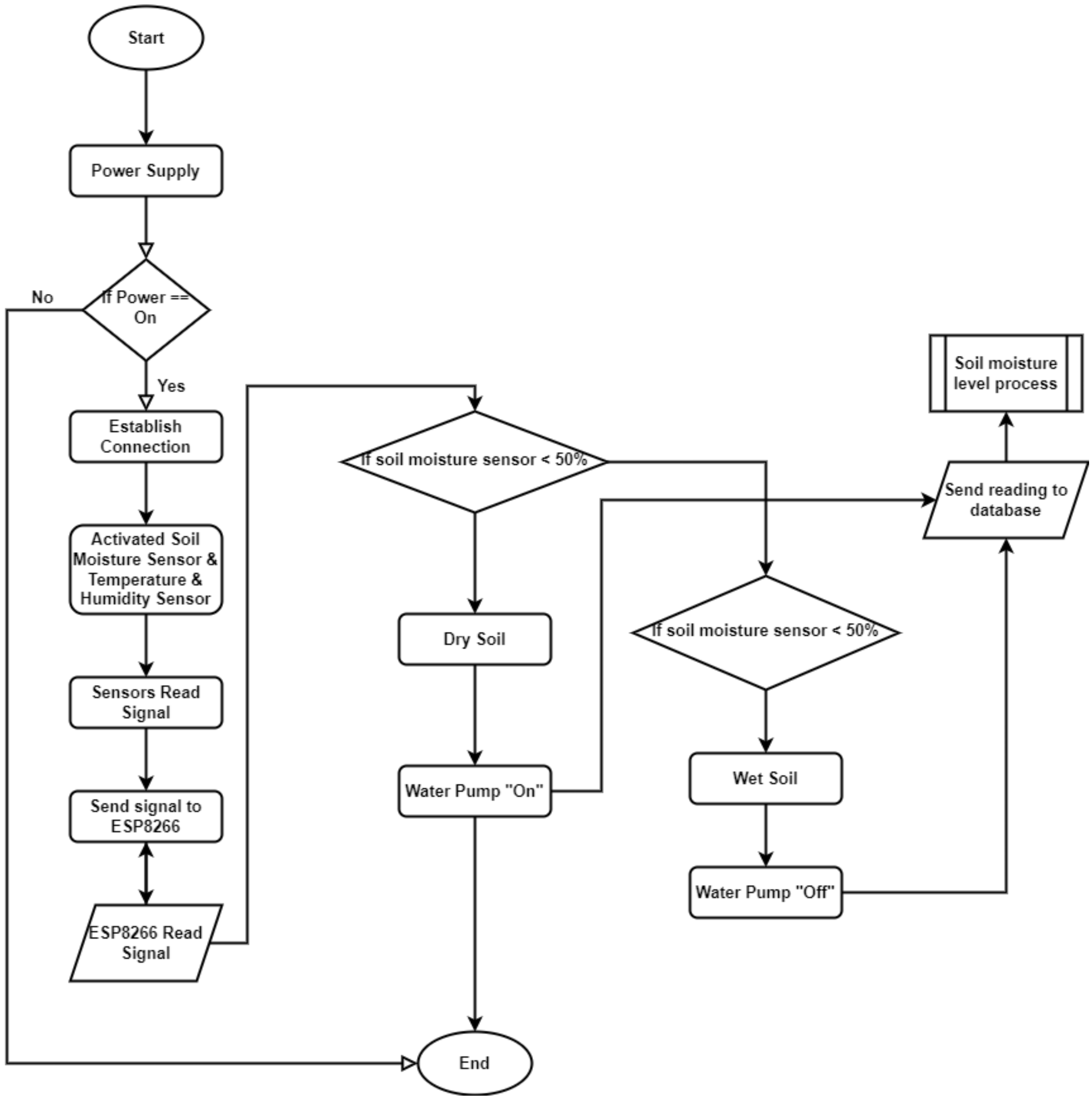


Figure 3.13 The Subproccess of Soil Moisture Flowchart

3.4.5.5 Subprocess Temperature and Humidity Flowchart

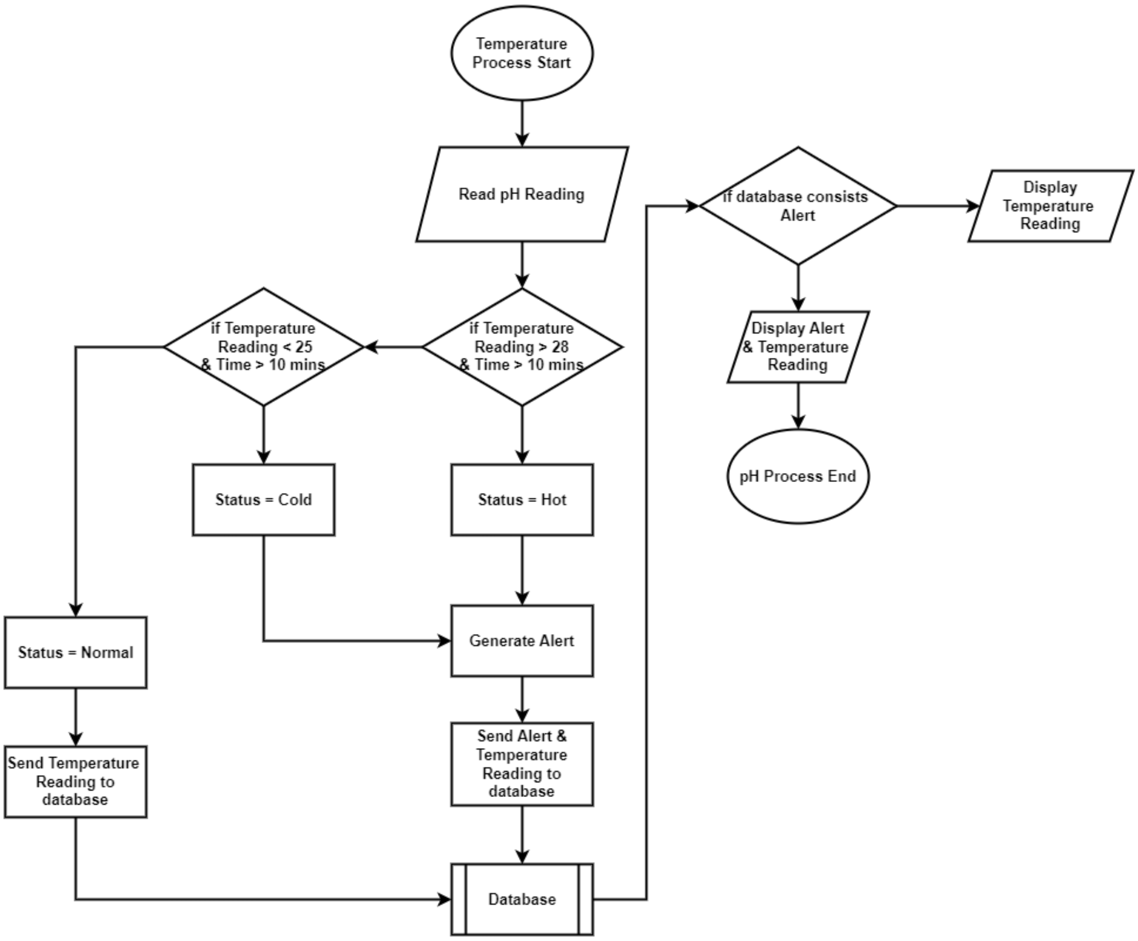


Figure 3.14 The Subprocess of Temperature & Humidity Flowchart

3.4.6 Context Diagram

A context diagram is a high-level visual representation of a system that shows the relationships between that system and its environment. It typically includes the system itself, along with the external entities that interact with it, such as people, other systems, or physical devices. The diagram is often used to provide a broad overview of a system and its components, and can help to identify the inputs, outputs, and processes involved in the system's operation. Based on the Figure 3.15 the user and the Arduino are examples of external entities in this system. The user will request real time data and the system will send the generated report to the user. The Arduino will send temperature, humidity, and soil moisture data to the system.

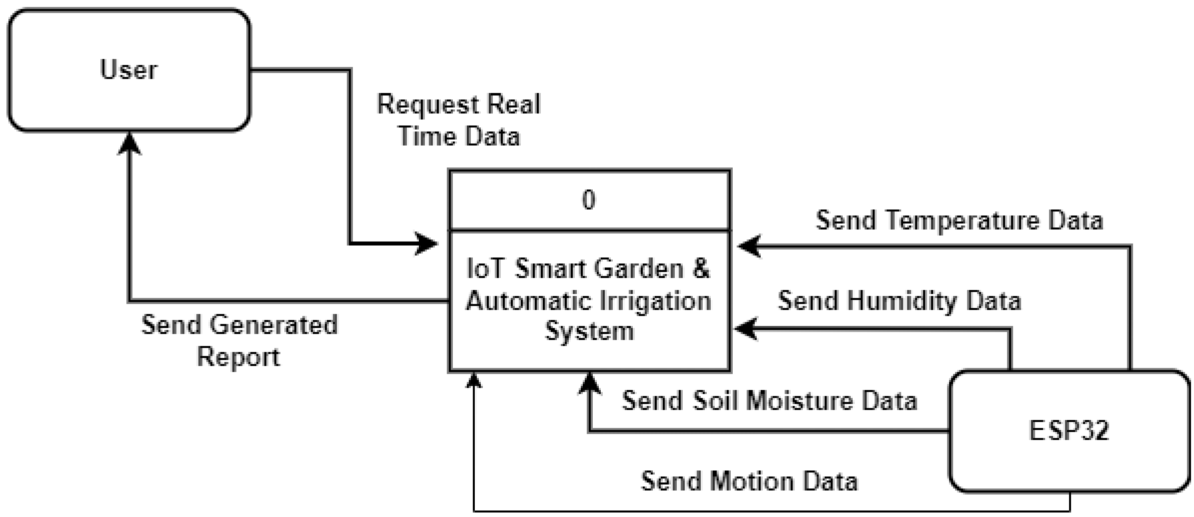


Figure 3.15 Context Diagram

3.4.7 Use Case Diagram

The use case diagram is a visual illustration of the system's behaviour where the relationships between the system, actors, and the use case are depicted in Figure 3.16 In use case describes the activities which the actors can login to the system, view garden data, view garden graph and get alert. In this system, the user will be able to manage plant and show garden data. While the developer will be managing the garden in term of setup soil moisture threshold for garden, setup temperature and humidity threshold of surrounding and setup notification for motion detection.

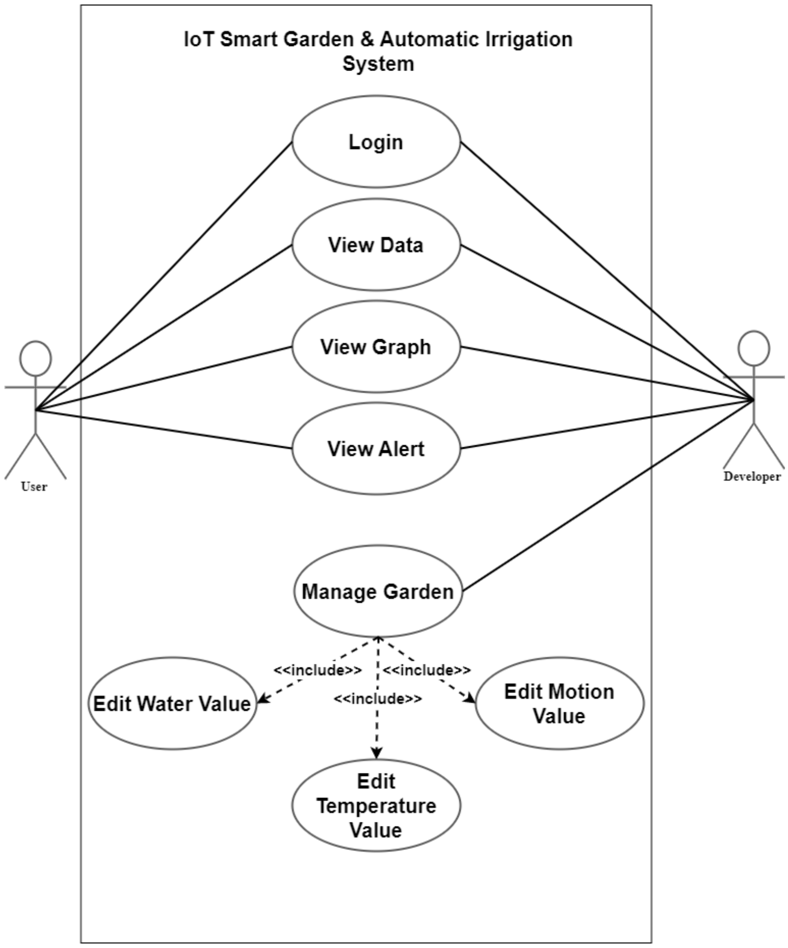


Figure 3.16 Use Case Diagram

3.4.8 Activity Diagram

Figure 3.17 shows the activity diagram for IoT Smart Garden & Automatic Irrigation System. ESP32 receives sensor data from the garden, including information on soil moisture, temperature, humidity, and motion detection. The data is analysed to determine the current needs of the plants in the garden. Based on the analysis, the system activates or deactivates the irrigation system to ensure that the plants are receiving the appropriate amount of water. The sensor send soil moisture, temperature & humidity and motion value at Blynk application through Wi-Fi connection.

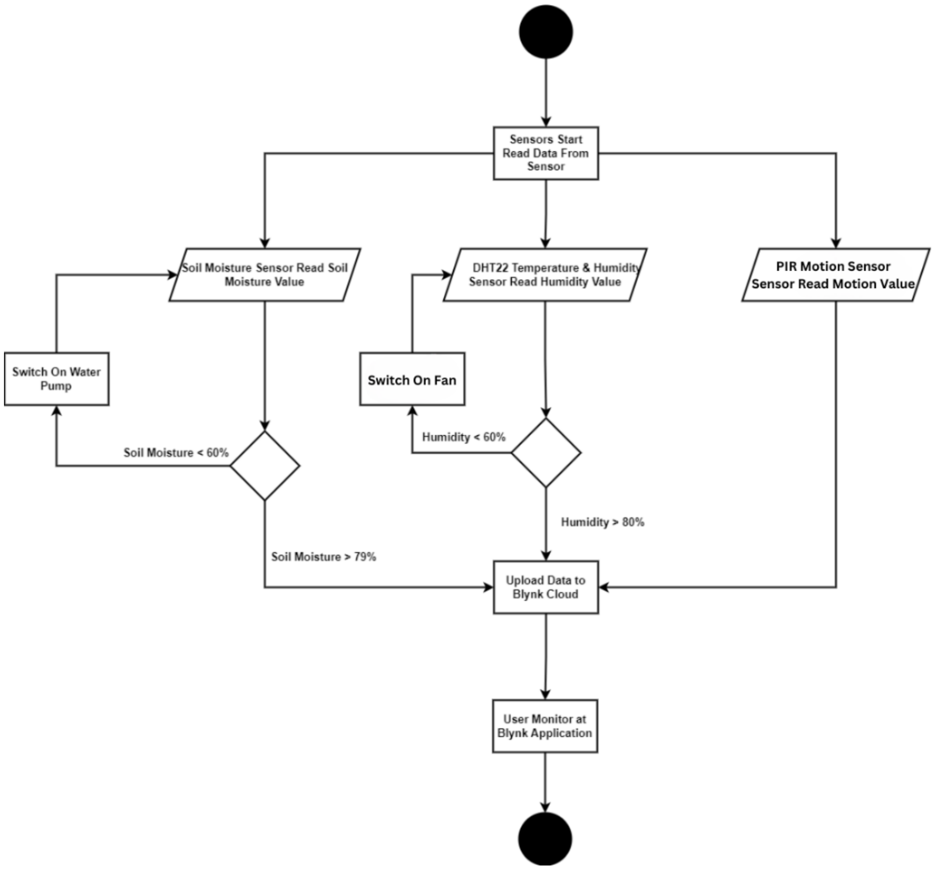


Figure 3.17 Activity Diagram

3.5 Data Design

3.5.1 Entity Relationship Diagram (ERD)

Figure 3.18 shows the proposed system's Entity Relationship Diagram consists of three tables: User, Logs, and Sensor. The primary key for the User table is Account_ID, for the Logs table is Log_ID, and for the Sensor table is Data_ID.

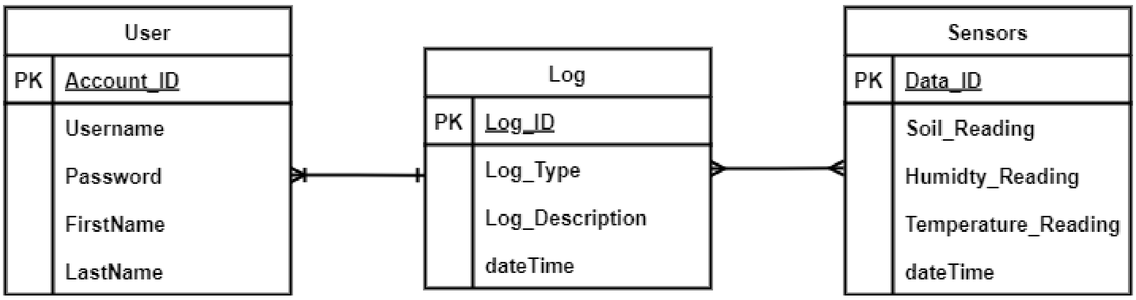


Figure 3.18 Entity Relationship Diagram

3.5.2 Database Design (PK, FK) and Data Model

3.5.2.1 User Table

Table 3.2 User Table

Attribute	Data Type	Length	Description	Key
Account_ID	int	11	Account_ID	(PK)
Username	nvarchar	30	Username	
Password	nvarchar	-	Password	
FirstName	nvarchar	50	First_Name	
LastName	nvarchar	50	Last_Name	

3.5.2.2 Sensors Table

Table 3.3 Sensors Table

Attribute	Data Type	Length	Description	Key
Data_ID	int	11	Data_ID	(PK)
Soil_Reading	float	10	Soil_reading	
Humidity_Reading	float	10	Humidity_reading	
Temperature_Reading	float	10	Temp_reading	
dateTime	timestamp		Date_Time	

3.5.2.3 Logs Table

Table 3.4 Logs Table

Attribute	Data Type	Length	Description	Key
Log_ID	int	11	Log ID	(PK)
Log_Type	varchar	10	Log type	
Log_Description	char	10	Log description	
dateTime	datetime	10	Time and date	

3.6 Hardware and Software Requirements

3.6.1 Hardware Requirements

3.6.1.1 ESP32

The ESP32 is a popular low-cost, low-power system-on-a-chip (SoC) microcontroller module developed by Espressif Systems. It is widely used in various Internet of Things (IoT) applications and projects due to its versatility and features. The ESP32 is equipped with a dual-core Tensilica LX6 microprocessor, which operates at up to 240 MHz. It has built-in Wi-Fi and Bluetooth connectivity, making it easy to connect to networks and other devices. The module also offers a variety of peripheral interfaces such as GPIO, I2C, SPI, and UART. One of the main advantages of the ESP32 is its built-in Wi-Fi and Bluetooth capabilities. It supports both 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth 4.2 and 5.0. This allows the ESP32 to connect to the internet, communicate with other devices, and interact with cloud services.

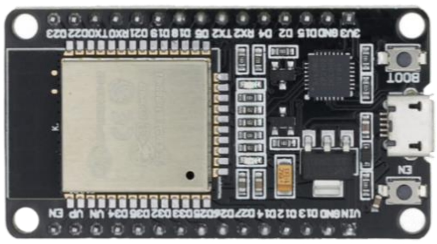


Figure 3.19 ESP32 Microcontroller

3.6.1.2 Soil Moisture Sensor

A soil moisture sensor is a device that measures the amount of water in the soil. It typically consists of a probe or sensor that is inserted into the soil and a reading device that displays the moisture level. Soil moisture sensors can be used to monitor the water content of soil in agricultural, gardening, and landscaping applications, as well as in environmental monitoring and research. They can help farmers, gardeners, and landscapers optimize irrigation and watering schedules, reduce water usage, and improve crop yields. Some soil moisture sensors can also be used to monitor soil moisture levels in areas prone to drought or flooding, or to track the effectiveness of soil conservation efforts. There are many different types of soil moisture sensors available, including capacitive, resistive, and time domain reflectometry (TDR) sensors.

Figure 3.20 Soil Moisture Sensor

3.6.1.3 Temperature and Humidity Sensor

A temperature and humidity sensor are a device that measures the temperature and humidity (moisture content) of the air. It typically consists of two sensors, one for temperature and one for humidity, and a reading device that displays the measured values. Temperature and humidity sensors are used in a variety of applications, including indoor and outdoor temperature and humidity monitoring, weather forecasting, HVAC (heating, ventilation, and air conditioning) system control, and industrial process control. They are often used in residential and commercial buildings to maintain comfortable and healthy indoor conditions, and in outdoor environments to monitor weather patterns and climate changes. Temperature and humidity sensors can be stand-alone devices or part of a larger system, and are available in a range of form factors, including handheld, wall-mounted, and wireless sensors.

Figure 3.21 Temperature & Humidity Sensor

3.6.1.4 PIR Motion Sensor

A PIR (Passive Infrared) motion sensor is a type of electronic sensor that detects changes in infrared radiation emitted by objects in its field of view. PIR motion sensors detect motion by sensing changes in the infrared energy emitted by objects. When a person or warm object moves within the sensor's range, it detects the variation in infrared radiation. PIR sensors consist of a pyroelectric sensor, which generates a voltage when exposed to infrared radiation, and a Fresnel lens that focuses the infrared radiation onto the sensor. The sensor measures changes in the received infrared energy to determine motion. IR motion sensors are commonly used in security systems, lighting control systems, and energy-saving applications. They can automatically turn on lights, activate alarms, or trigger other actions in response to detected motion.

Figure 3.22 PIR motion Sensor

3.6.1.5 5V Water Pump

A water pump is a device that is used to move water from one place to another. It can be powered by electricity, a gasoline engine, or a diesel engine, and is commonly used to transfer water from a well, a pond, or a river into a home or irrigation system. Water pumps can also be used to drain water from flooded areas or to remove excess water from a swimming pool. There are many different types of water pumps available, each designed to meet specific needs and requirements. Some common types include centrifugal pumps, jet pumps, submersible pumps, and diaphragm pumps.

Figure 3.23 5V Water Pump

3.6.2 Hardware Support

Table 3.5 Hardware Support

Hardware	Specification	Purpose
Laptop	Brand: Lenovo Processor: Intel Core i7 Windows OS RAM: 8GB System Type: 64-bit OS	To develop the system, a device is required to write the code, install software, and design the system. This device can be a computer, laptop or any other similar device that has the capability to run the necessary software and tools to develop the system.

ESP32	The ESP32 is equipped with a dual-core Tensilica LX6 microprocessor, which operates at up to 240 MHz. Operating Voltage: 5V Power Supply: USB Only	The device also acts as a microcontroller, which connects the sensor to the cloud and facilitates communication between the two. It receives data from the sensor, processes it, and sends it to the cloud for further analysis, storage, and decision-making.
Sensors	Soil Moisture Sensor Temperature & Humidity Sensor PIR Motion Sensor	Sensor that essential component for operating a Smart Garden.
Breadboard	Null	It is construction base for electronic circuit prototype

3.6.3 Software Requirements

3.6.3.1 Arduino Integrated Development Environment (IDE)

The Arduino IDE is based on the Processing language and includes a code editor, a built-in compiler, and a debugger. It also includes a library of example sketches and a serial monitor for interacting with your Arduino board. To use the Arduino IDE, you will need to connect your Arduino board to your computer with a USB cable and select the correct port and board type in the IDE. You can then write your code in the editor, upload it to the board, and use the serial monitor to see the output of your program.

Figure 3.24 Arduino IDE

3.6.3.2 Blynk Application

Blynk is a platform that allows you to control and monitor your connected devices using a smartphone app. It can be used with a variety of hardware platforms, including Arduino, Raspberry Pi, and ESP8266, and is designed to make it easy to build Internet of Things (IoT) projects.

Figure 3.25 Blynk Application

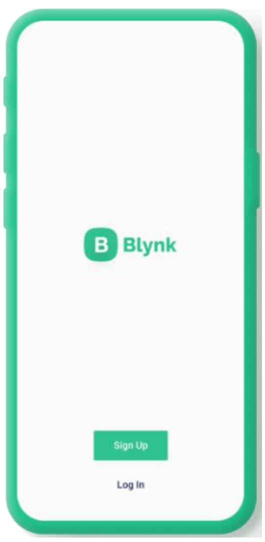
3.7 Proof Of Initial Concept

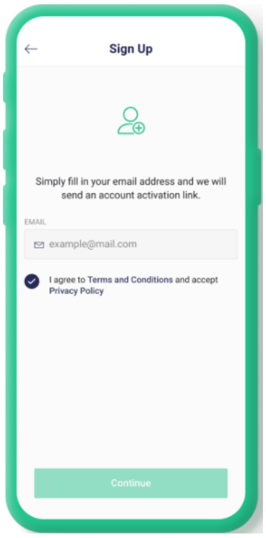
3.7.1 Prototype

Based on the Figure 3.26 it shows the storyboard for IoT Smart Garden & Automatic Irrigation System.

Figure 3.26 Prototype for Proposed System

Table 3.6 Interface Design

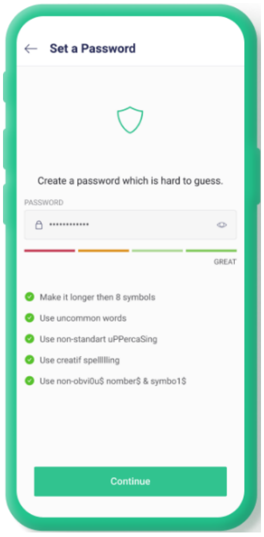
Interface	**Description**
	**Home Page** This is the Home Page for Blynk application. This is the first screen after application launch.

<table>
<tr>
<td>


</td>
<td>

Sign Up Page

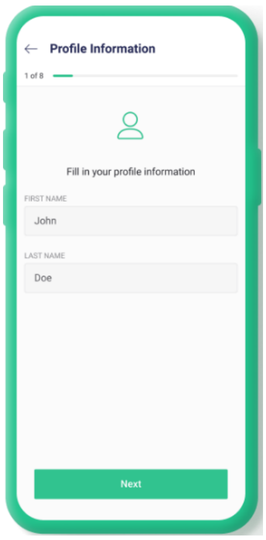
On this Sign-Up page, user need to enter E-mail, read, and agree to the Terms and Conditions. Tap Continue and user need to check their inbox for an Email with instructions.

</td>
</tr>
<tr>
<td>


</td>
<td>

Set Password Page

In the user’s email, user need to tap Create Password button and the link will return user to the application.

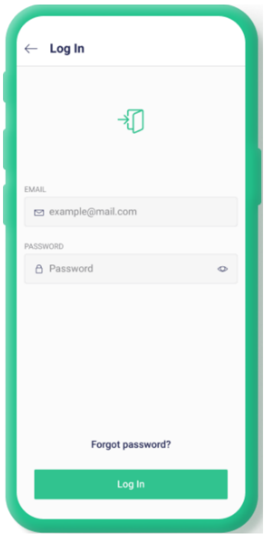
</td>
</tr>
<tr>
<td>


</td>
<td>

Profile Information Page

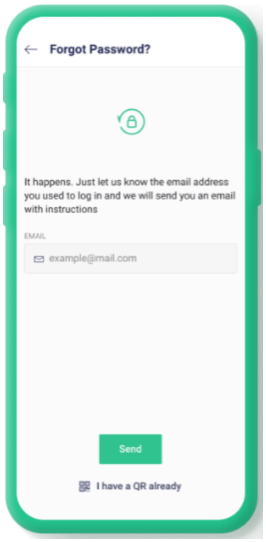
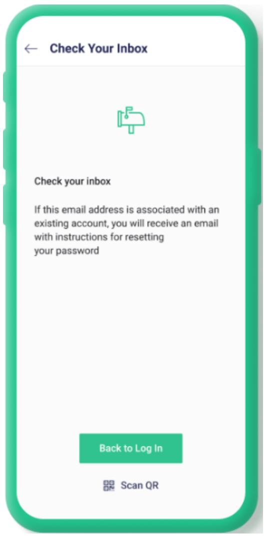
Users need to enter the password according to the hint on the field and need to tap Continue and fill in user profile required and optional information. Once user have finished with Profile setup, user can add and manage Devices.

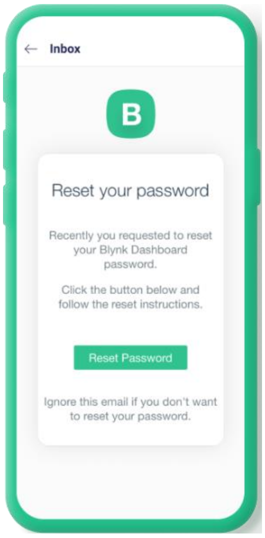
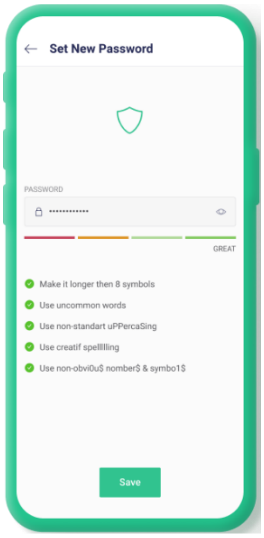
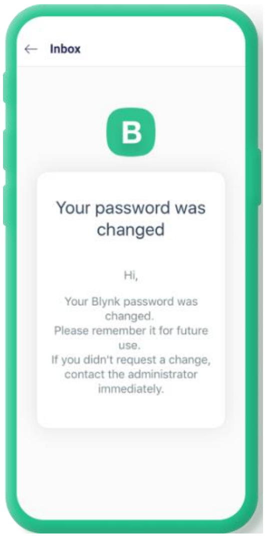
</td>
</tr>
</table>

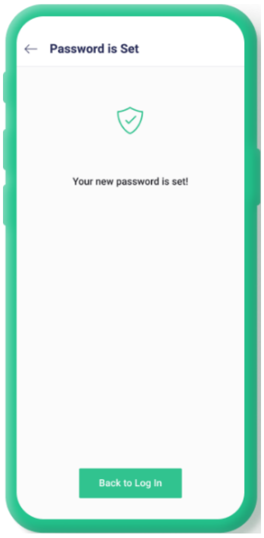
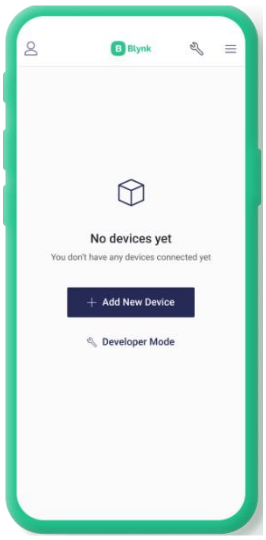
<table>
<tr>
<td>
</td>
<td>

Password Reset Page

Users need to tap Forgot password?

</td>
</tr>
<tr>
<td>
</td>
<td>Users need to confirm reset password link sending by tap on the Send button.</td>
</tr>
<tr>
<td>
</td>
<td>After the link is sent, users need to follow the instructions stated on the screen and then return to Login button. The link will be sent to the user's Email that was linked to user's Blynk account during Sign Up.</td>
</tr>
</table>

	Users need to tap Reset Password button got in the Email. Once user have tapped Reset Password button, user be redirected back to Blynk application and asked to type a new password.
	Tap Save button to submit new password
	Your account password is successfully changed. Users get UI an Email confirmation about that.

<table>
<tr>
<td>
</td>
<td>Tap Back to Login button and sign into account.</td>
</tr>
<tr>
<td>
</td>
<td>Add Device Page

Once users logged in, Devices tab will be opened. Here users can add and view and manage devices or switch to Developer Mode.</td>
</tr>
</table>

3.7.2 Software Setup

An IoT smart garden typically includes a web dashboard and a mobile app dashboard that allow user to monitor and control various aspects of the garden remotely. The Figure 3.27 is the web dashboard and Figure 3.28 is mobile app dashboard are typically accessed through a browser or an app on a smartphone or tablet, and they may provide a variety of features, including:

- Real-time monitoring of temperature, humidity, and soil moisture.

- Control of irrigation systems.
- Monitoring of plant growth and health.
- Historical data and analytics to track the performance of the garden over time.

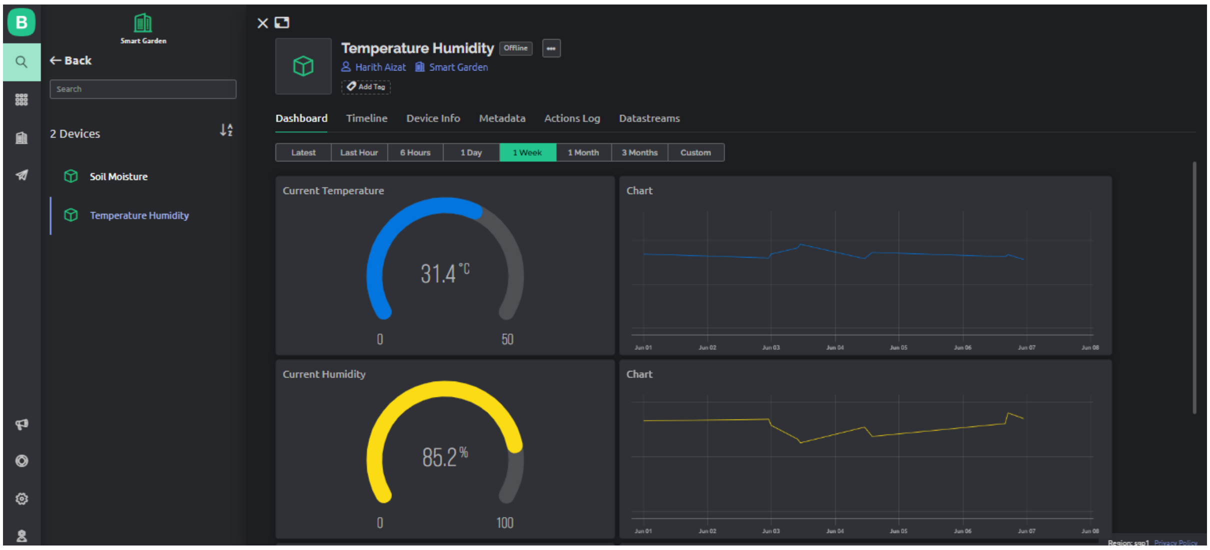


Figure 3.27 Web Dashboard

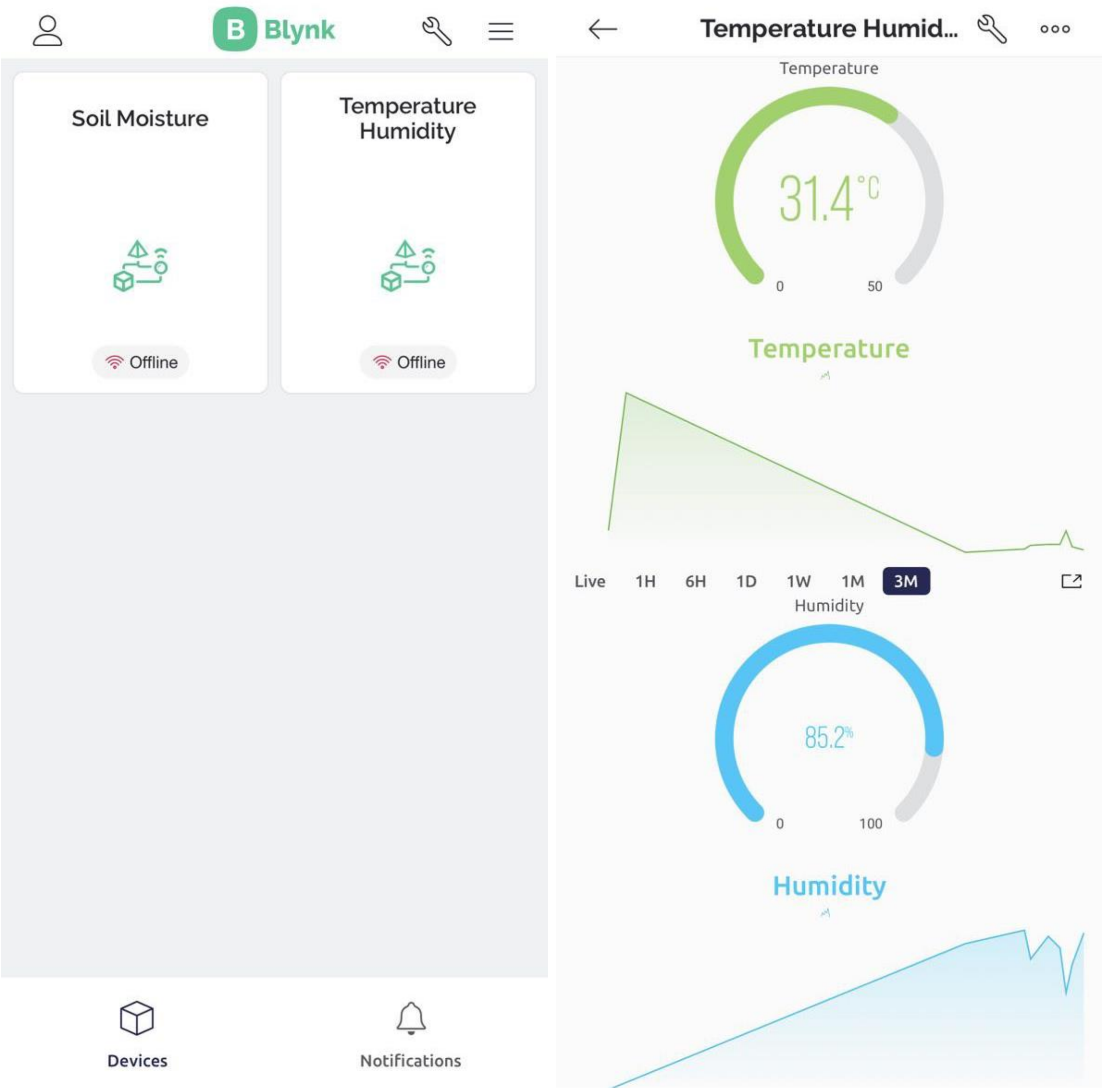


Figure 3.28 Mobile App Dashboard

3.7.3 Blynk Connection

As illustrated in Figure 3.29, the connection between the Blynk project and the ESP32 microcontroller is established by utilizing the Blynk library and the appropriate hardware connections. Once the connections have been made, the microcontroller's internet connection is verified to ensure proper communication between the Blynk project and the microcontroller.

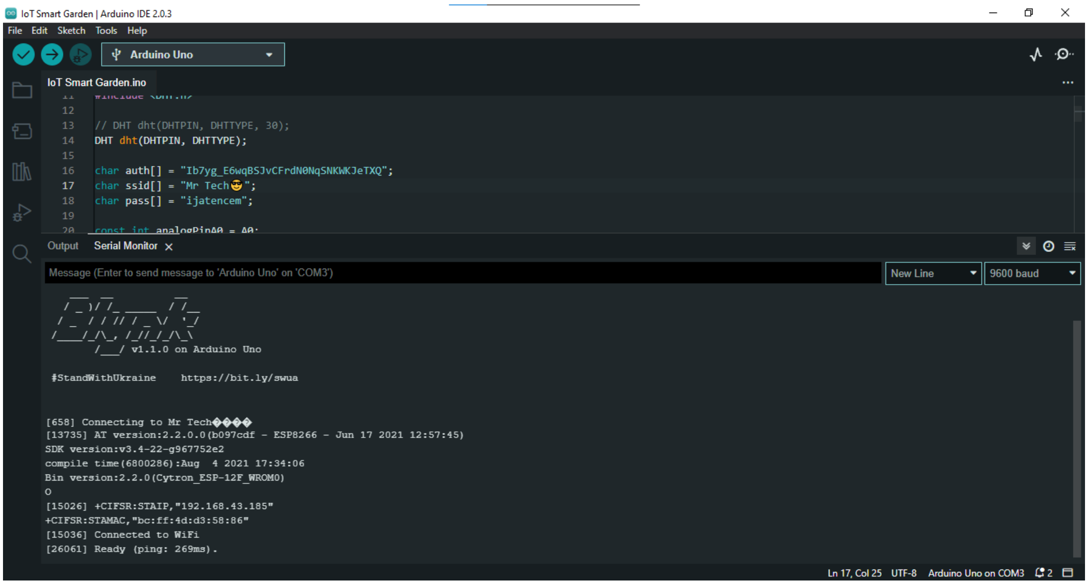


Figure 3.29 Blynk connection to Wi-Fi

3.7.4 Hardware Setup

The hardware setup for an IoT Smart Garden, as depicted in Figure 3.30, serves as proof of an initial concept. At this stage, the Smart Garden includes two types of sensors: a soil moisture sensor and a temperature and humidity sensor. These sensors collect data that is sent to a microcontroller for processing and then to the Blynk application. The data is then made available to users through a web and mobile interface, where they can view sensor data and keep track of the garden's conditions.

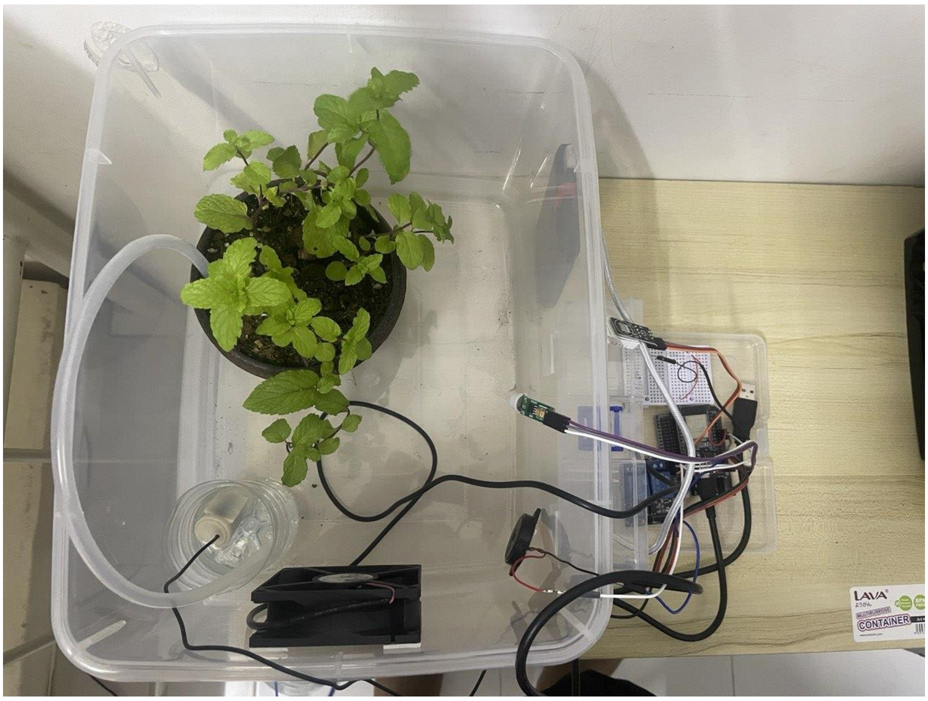


Figure 3.30 Hardware Setup

3.8 Testing and Validation Plan

3.8.1 Blynk Application Testing Plan

Table 3.7 Blynk Application Testing Plan

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Login with correct email and password	Email: m.harithaizat@gmail.com Password: 123456	Login successful and redirect to homepage			
Login with incorrect username	Email: adlansyahin@gmail.com Password:	Login unsuccessful and error message pops-			

and password	11223344	up and returns login page			
When My Devices is clicked	Null	The interface should consist of My Devices.			

3.8.2 Sensors, Hardware, and Database Connection Testing Plan

Table 3.8 Sensors, Hardware, and Database Connection Testing Plan

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When power “On”	Null	Proposed system is active and send reading			
Soil moisture level less than 50%	Null	Active the water pump			
Soil moisture level more than 79%	Null	Stop the water pump			
Temperature level is less 33	Null	Stop the Fan			
Temperature level more than 35	Null	Active the Fan			
ESP32 when power is “On”	Null	LED light is on			

Database when power "On"	Null	Connection is successful with ESP32 board			

3.8.3 Graph Section Testing Plan

Table 3.9 Graph Section Testing Plan

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When soil moisture graph button is clicked	Null	The soil moisture graph will show on the interface.			
When temperature graph button is clicked	Null	The temperature graph will show on the interface.			
When humidity graph button is clicked	Null	The humidity graph will show on the interface.			

3.9 Potential Use of Proposed Solution

An IoT smart garden is a type of automated gardening system that uses Internet of Things (IoT) technology to monitor and control various aspects of a garden, such as irrigation, temperature, humidity, and plant health. Some smart gardens may also have features such as pest control and lighting. Smart gardens can be used in a variety of settings, including residential gardens, commercial farms, and public parks. They can be especially useful for people who are unable to physically maintain their gardens due to physical limitations or a busy schedule, or for those who want to optimize the growth of their plants by providing the optimal conditions.

Some potential uses for an IoT smart garden include:

i. Home gardens: A smart garden can help homeowners maintain a healthy, productive garden with minimal effort. With sensors and automated irrigation systems, a smart garden can ensure that plants receive the right amount of water and nutrients and alert the gardener if there are any issues.

ii. Commercial farms: Smart gardens can be used to optimize the growth of crops on a larger scale, helping farmers increase yield and reduce waste.

iii. Public parks and community gardens: Smart gardens can be used to maintain and manage public green spaces, ensuring that plants are healthy and well-maintained for the enjoyment of the community.

iv. Indoor gardens: Some smart gardens are designed for indoor use, allowing people to grow plants indoors even if they don't have a traditional outdoor garden space.

3.10 Gantt Chart

The Gantt Chart shows the progress of the project with the timescale. The project will be following the proposed methodology. The Gantt Chart of the IoT Smart Garden & Automatic Irrigation System will be show at Appendix B.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter provides a detailed explanation of the system implementation process, testing, and results obtained using the RAD methodology discussed in the previous chapter. It covers the system implementation method, testing techniques used to achieve the project objectives, programming languages employed, hardware and software setup, and project constraints. The discussion and results of this project aid in future system planning and direction.

4.1.1 Development Environment

The system was built using a combination of software, including Arduino IDE and Blynk Application. Arduino IDE served as the primary interface for coding, uploading, and monitoring sensor activity. The programming language used for sketching the code was the Arduino programming language, enabling the microcontroller to interact with the sensors. The Arduino IDE's serial monitors facilitated monitoring and debugging during the project. Since the smart garden system employed an Arduino microcontroller board, the open-source Arduino Software IDE was the ideal tool for writing code. This platform comprises a text editor, a message area, toolbars, and menus with compiling and uploading features. The language used is standard C++, as implemented by the GNU C++ compiler. With the Arduino IDE, the user can connect the Arduino microcontroller to the sensors, upload sketches to the microcontroller, and define the hardware functions. The Figure 4.1 shows the Arduino IDE interface.

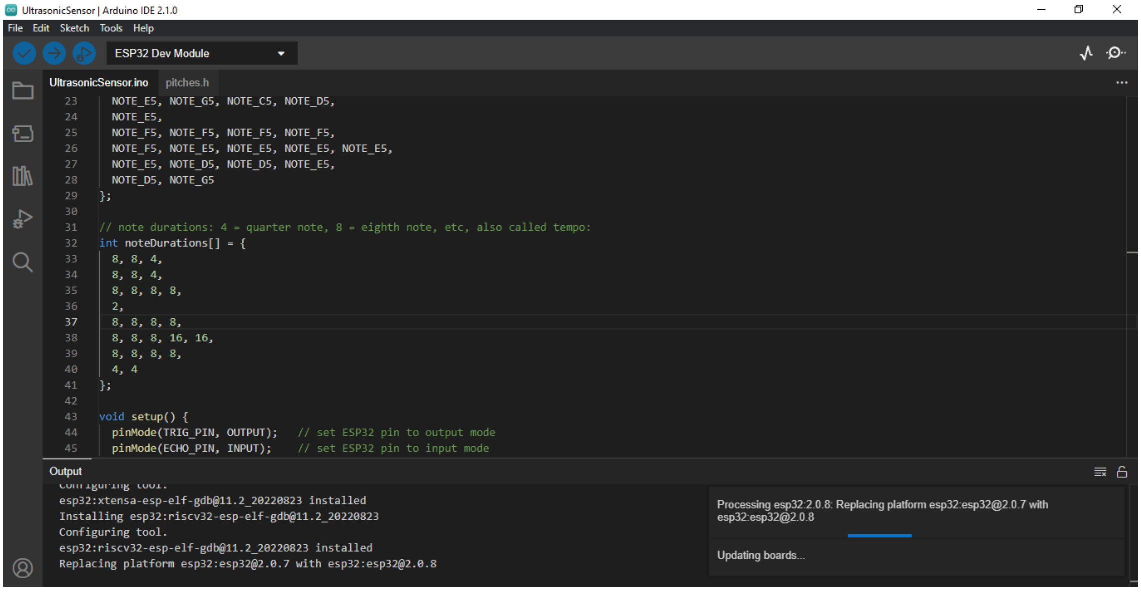


Figure 4.1 Arduino IDE

4.2 Blynk Application System

The primary target audience for the IoT smart garden system are individuals who lead busy lives and cannot dedicate much time to maintaining their gardens. The system's main objective is to alleviate their worries by providing an automated solution to garden maintenance. The sensors installed in the garden collect data on soil moisture, temperature, and humidity, which is then processed to identify issues. The system responds by activating a water pump to irrigate the soil when moisture levels are low and turning on a fan when temperatures are high. Additionally, the system provides users with graphical representations and statistics on their garden's condition.

4.2.1 Login Interface

The login page in Blynk apps serves as the initial point of entry for users to access their Blynk accounts and control their connected devices. It requires users to provide their unique username and password credentials, which are verified by the app's server before granting access to the account. The login page is a security feature that helps protect user data and prevent unauthorized access to their Blynk devices and services. The Figure 4.2 shows the login interface of Blynk applications.

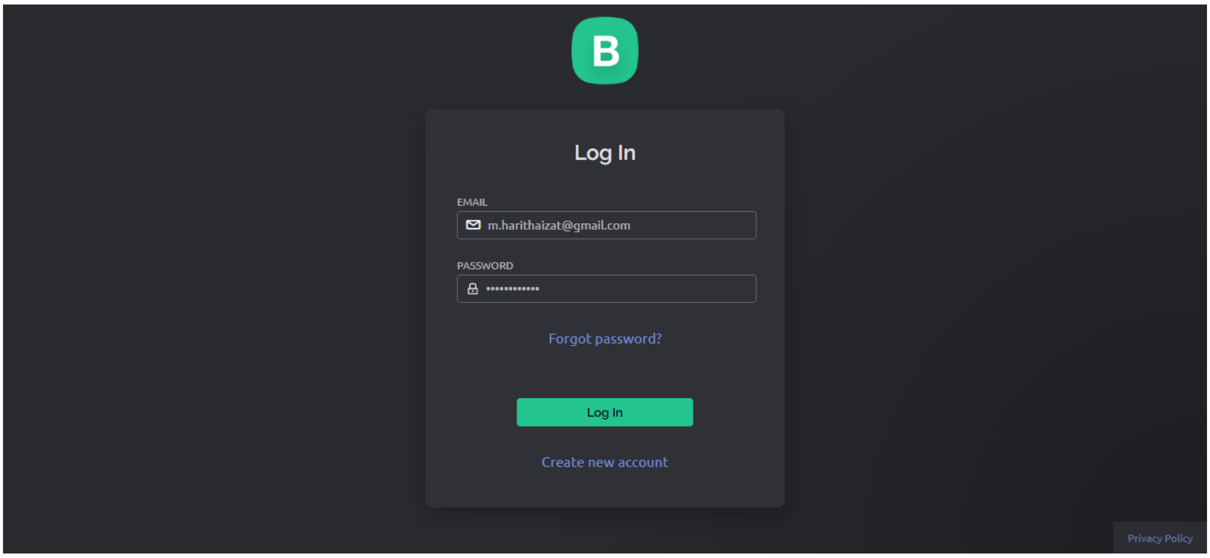


Figure 4.2 The Login Interface

4.2.2 My Devices Page

Based on the **Figure 4.3** it shows My devices page where this page will be redirected after user login to Blynk application. This section displays a list of users connected devices. Each device is usually represented by a name or an icon. User can see the status of each device, such as whether it's online or offline. So, in this case the devices name is Soil Moisture and Temperature Humidity. User can click on the devices name and can control the function for IoT Smart Garden.

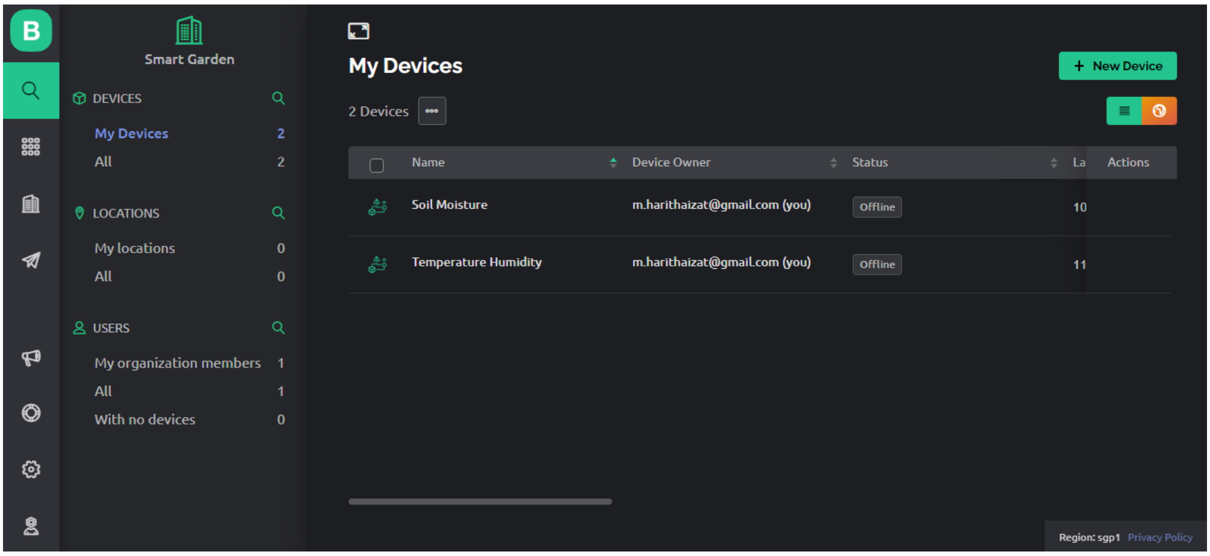


Figure 4.3 The My Devices Interface

4.2.3 Soil Moisture Dashboard

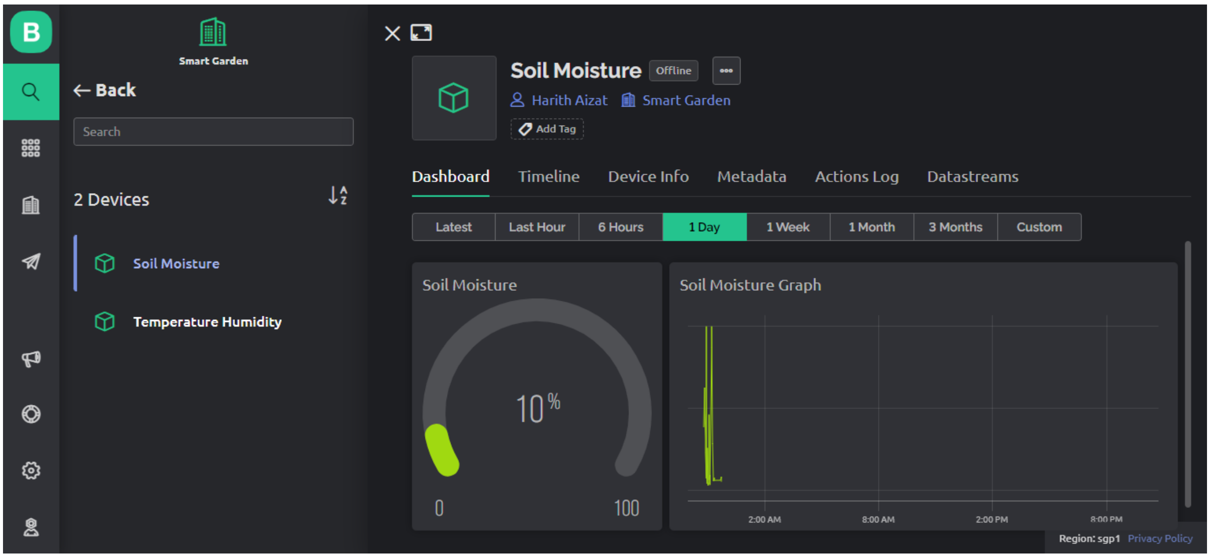


Figure 4.4 Soil Moisture Dashboard

Based on the Figure 4.4 shows the gauge meter for soil moisture and the graph for soil moisture value. The soil moisture sensor collects the data from the soil and send to Blynk application using Wi-Fi. User can monitor from Blynk web-based interface and mobile interface. The report for graph is stored in Blynk database and user can download the report.

4.2.4 Temperature & Humidity Dashboard

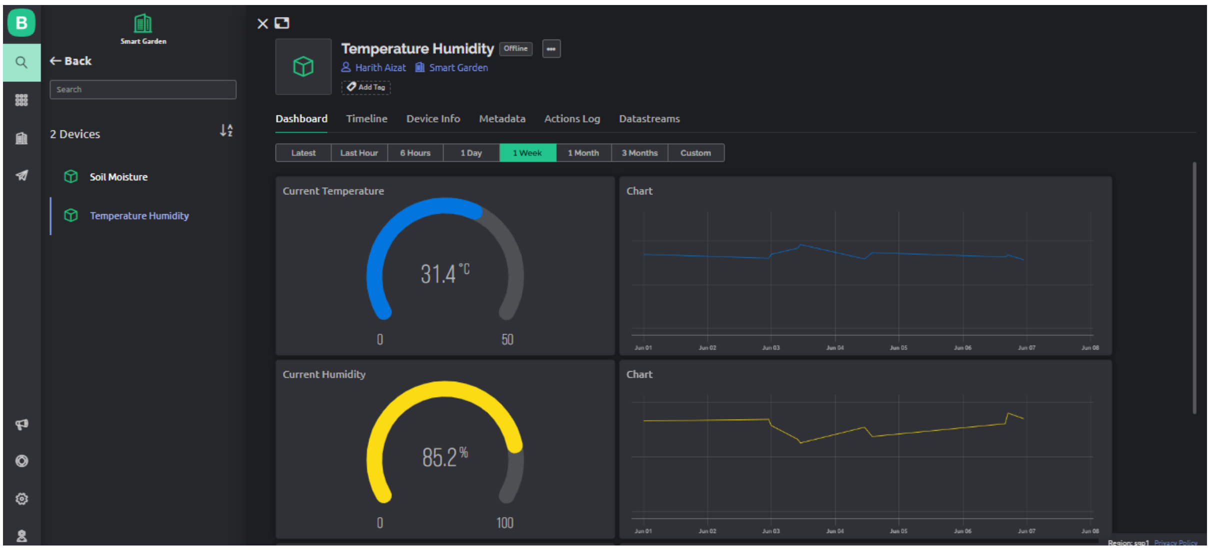


Figure 4.5 Temperature & Humidity Dashboard

Based on the Figure 4.5 it shows the dashboard for temperature and humidity sensor. The DHT22 sensor collect the data from the greenhouse and send the data to Blynk application through Wi-Fi connection. When the temperature is more than the threshold the fan will be turned on to reduce the greenhouse temperature. The report for temperature and humidity can be downloaded for analysis purposes.

4.2.5 Motion Detected

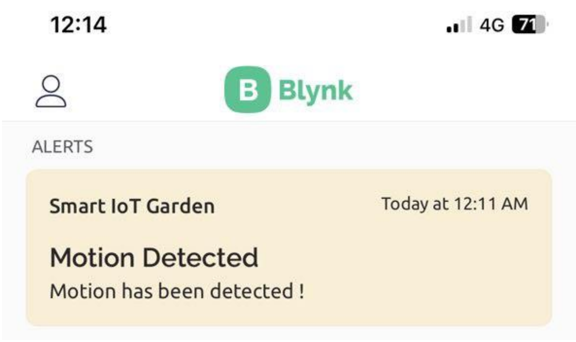


Figure 4.6 Motion Detected

Based on the Figure 4.6 shows the alarm and notification when there is motion and the garden. PIR motion sensor has been setup with Piezo Buzzer at the garden to avoid the intruder. User will be received the notification and alert about the motion detected.

4.2.6 Graph Result

- Soil Moisture Graph

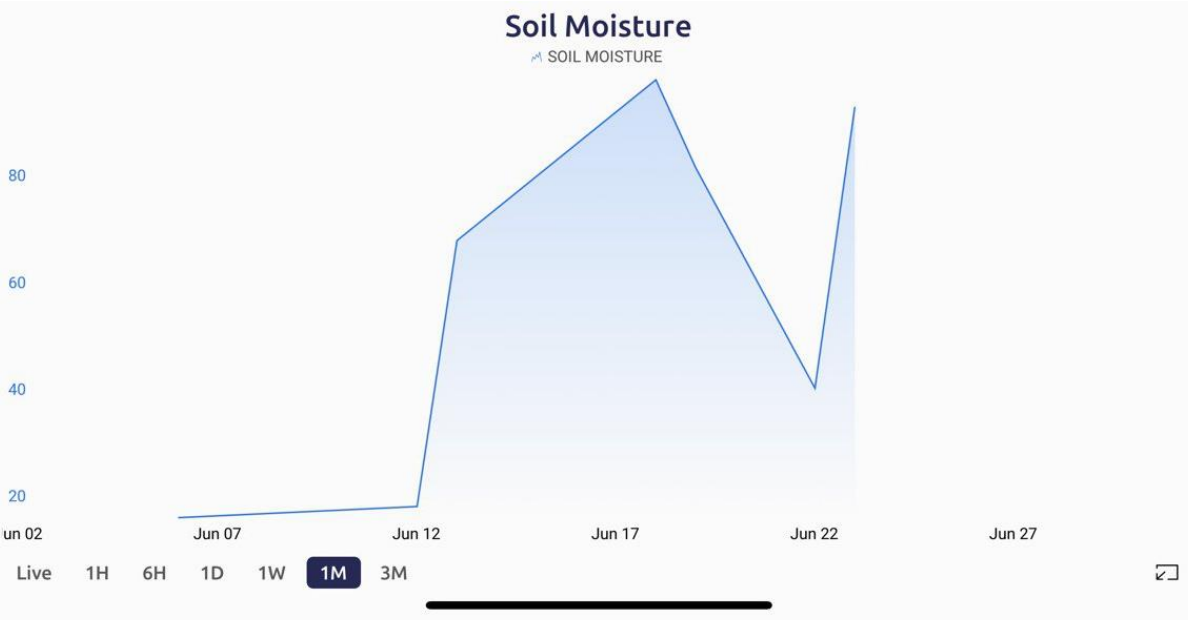


Figure 4.7 Soil Moisture Graph

Figure 4.7 shows the soil moisture graph and it display the data in real-time based on the update interval. The Blynk app receives the data from the microcontroller and displays it on the Graph widget in real-time. The x-axis represents time, and the y-axis represents the moisture values. As new data points arrive, the graph updates accordingly, showing a continuous line or curve. By observing the graph, user can track the moisture levels in soil over time. This can help user monitor the watering needs of the plants or determine if additional irrigation or watering is required.

- Temperature and Humidity Graph

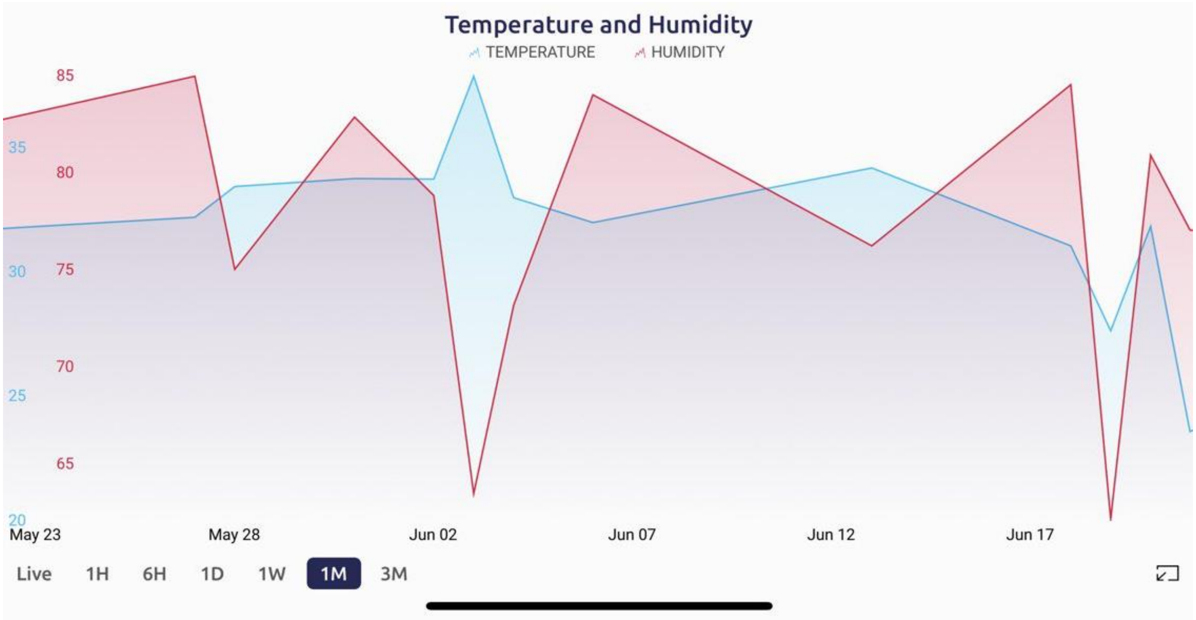


Figure 4.8 Temperature & Humidity Graph

Figure 4.8 shows the temperature and humidity graph and it display the data in real-time based on the update interval. The Blynk app receives the temperature and humidity data from the microcontroller and displays it on the Graph widget in real-time. The x-axis represents time, while the y-axis represents the temperature or humidity values. As new data points arrive, the graph updates accordingly, displaying a continuous line or curve. By observing the temperature and humidity graph, use can gain insights into the variations in greenhouse conditions over time. This information can be helpful for monitoring greenhouse environments, assessing the effectiveness of ventilation system, or analysing patterns and trends.

4.3 Hardware Implementation

A microcontroller is a small computer that can receive input from sensors and control output to actuators based on programmed logic. In an IoT smart garden, a microcontroller ESP32 can serve as the brain of the system. The ESP32 is a powerful and versatile microcontroller designed for use in Internet of Things (IoT) applications. It is a successor to the popular ESP32 microcontroller and features more processing power, built-in Wi-Fi and Bluetooth connectivity, and a wider range of input/output options. These are devices that can measure various parameters of the garden environment, such as soil moisture, temperature, humidity, and ultrasonic sensor. These sensors will detect the soil moisture sensor and send data via the Internet.

To establish connections between the sensors and the Arduino microcontroller board, hardware components such as breadboards, male-to-male jumper wires, and male-to-female jumper wires were utilized. The initial step involved setting up the hardware, with male-to-male jumper wires connected from the 5 volt and GND ports to the breadboard to resolve the limited pin number issue on the Arduino microcontroller board and expand the number of pins for the sensors. Following this, the sensors were connected to their respective ports on the Arduino using jumper wires. Once the connections were made, the Arduino microcontroller board was connected to the laptop via a USB cable, and the code was uploaded to the microcontroller using the Arduino IDE software. The serial monitor was then employed to monitor the sensors' activity. The Figure 4.9 shows the IoT Smart Garden hardware prototype.

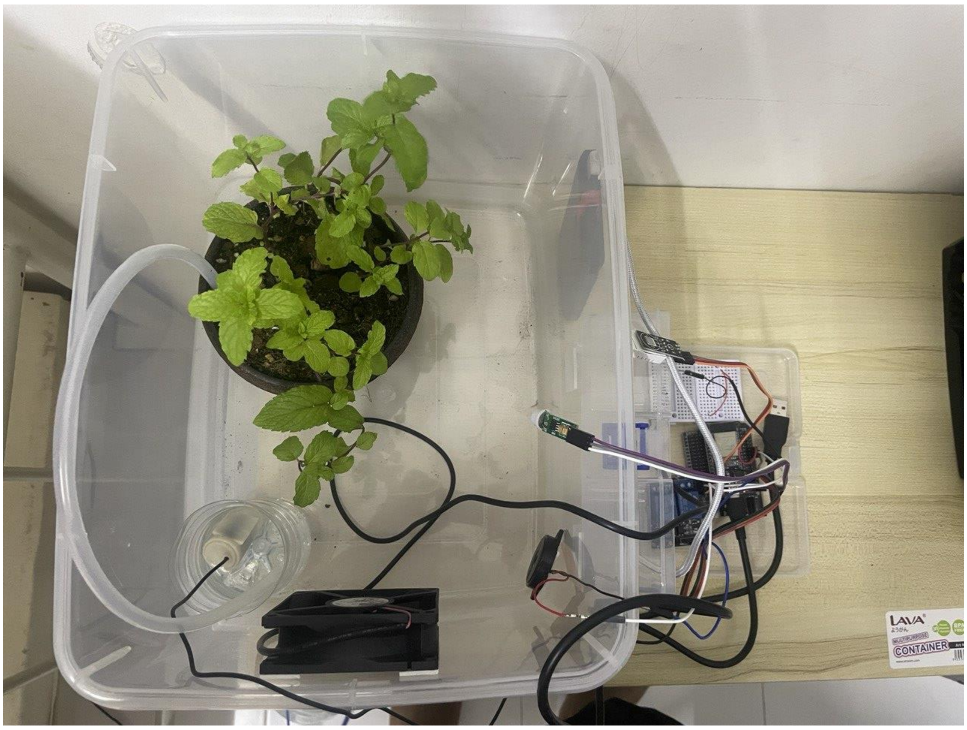


Figure 4.9 The IoT Smart Garden hardware prototype

Soil Moisture Sensor with Water Pump

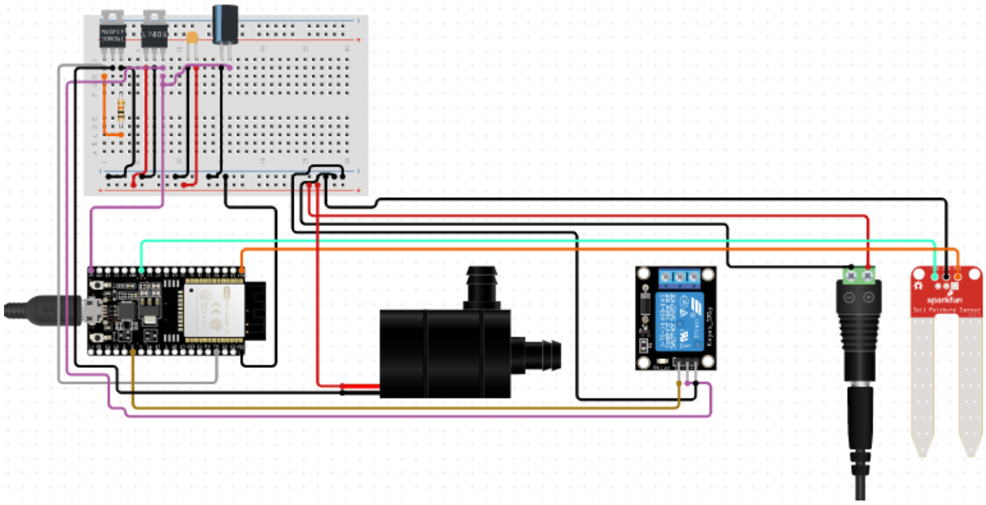


Figure 4.10 Soil Moisture Sensor Diagram

Figure 4.10 shows the diagram for soil moisture sensor and water pump. This Arduino code utilizes an ESP32 microcontroller with WiFi capabilities to connect to the Blynk IoT platform. It includes libraries for WiFi and Blynk. Macro values are defined for Blynk configuration, and constants are set for dry and wet soil moisture values, as well as pin assignments for the soil moisture sensor and relay pump. The setup function initializes serial communication, connects to Blynk, and sets pin modes for the sensor and pump. The loop function repeatedly reads the soil moisture sensor and runs Blynk communication. The soilMoistureSensor function reads the analog value from the sensor, converts it to a percentage based on the dry and wet values, and controls the relay pump accordingly. It also logs soil moisture events to Blynk. Overall, this code enables the ESP32 to connect to Blynk, read soil moisture levels from a sensor, and control a relay pump based on the moisture readings.

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPL6E64TWjWw"
#define BLYNK_TEMPLATE_NAME "Soil Moisture"
#define BLYNK_AUTH_TOKEN "SV3pt869e8yk4aTRG-jY-sNfoS_lMrAG"

const int DryValue = 4095;
```

```
const int WetValue = 800;
int soilMoistureValue = 0;
int soilmoisturepercent = 0;
int SensorPin = 36;
int RelayPump = 22;
char ssid[] = "Harith Aizat";
char pass[] = "smartgarden";

void setup() {
  Serial.begin(9600);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  pinMode(SensorPin, INPUT);
  pinMode(RelayPump, OUTPUT);
}

void loop() {
  soilMoistureSensor();
  Blynk.run();
}

void soilMoistureSensor() {
  soilMoistureValue = analogRead(SensorPin;
  Serial.println(soilMoistureValue);
  soilmoisturepercent = map(soilMoistureValue, DryValue, WetValue, 0, 100);
  if (soilmoisturepercent < 60) {
    digitalWrite(RelayPump, LOW);
    Blynk.logEvent("soil_moist");
  } else {
    digitalWrite(RelayPump, HIGH);
}
  Blynk.virtualWrite(V1, soilmoisturepercent);
}
```

Temperature & Humidity Sensor with Fan

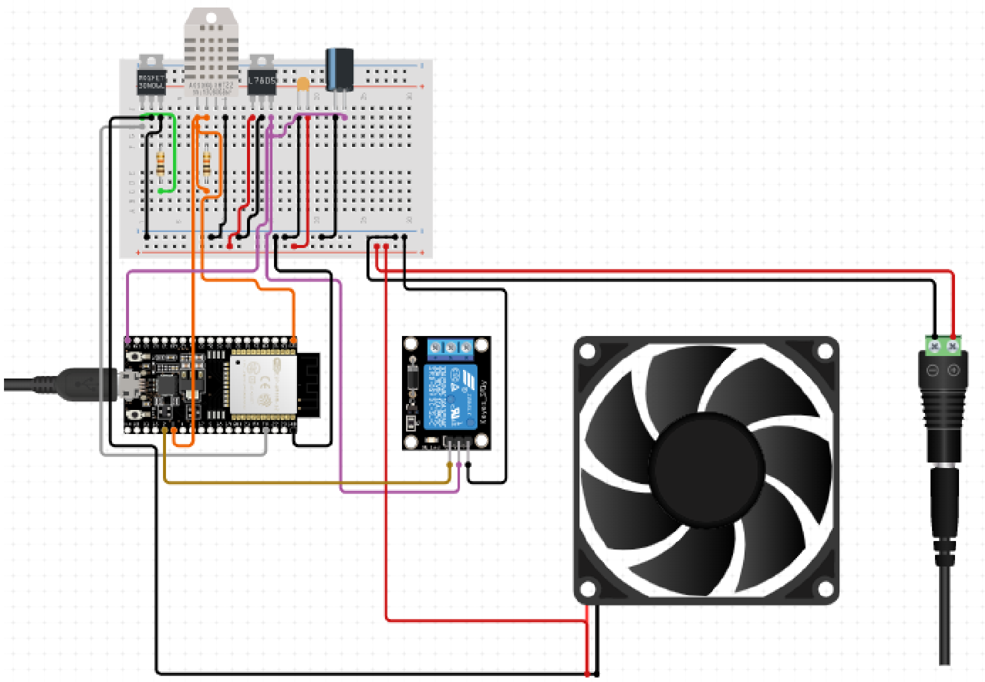


Figure 4.11 Temperature & Humidity Sensor Diagram

Figure 4.11 shows the diagram for temperature & humidity sensor. The provided Arduino sketch is for an IoT Smart Garden project. It uses an ESP32 board along with a DHT22 sensor to monitor temperature and humidity levels in a garden. The sketch connects to the Blynk platform using Wi-Fi and sends data to the Blynk app for visualization. This Arduino code utilizes an ESP32 microcontroller with WiFi capabilities to connect to the Blynk IoT platform. It includes libraries for DHT temperature and humidity sensor, WiFi, and Blynk. Macro values are defined for Blynk configuration, and pin assignments are made for the relay fan, DHT sensor, and threshold values for temperature control. The setup function initializes pins, serial communication, connects to Blynk, and starts the DHT sensor. The loop function repeatedly reads temperature and humidity values from the DHT sensor and runs Blynk communication. The temperatureHumiditySensor function reads the sensor values, checks if the temperature is above a predefined upper threshold, and controls the relay fan accordingly. It also logs temperature events to Blynk. Overall, this code enables the ESP32 to connect to Blynk, read temperature and humidity values from a sensor, and control a relay fan based on the temperature readings.

```
#include <DHT.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPLCr7cIo9X"
#define BLYNK_TEMPLATE_NAME "Temperature Humidity"
#define BLYNK_AUTH_TOKEN "Ib7yg_E6wqBSJvCFrdN0NqSNKWKJeTXQ"
#define RelayFan 22
#define DHT_SENSOR_PIN 19
#define DHT_SENSOR_TYPE DHT22
#define TEMP_UPPER_THRESHOLD 26
#define TEMP_LOWER_THRESHOLD 24
DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);

char ssid[] = "Harith Aizat";
char pass[] = "smartgarden";

void setup() {
  pinMode(RelayFan, OUTPUT);
  pinMode(DHT_SENSOR_PIN, INPUT);
  Serial.begin(9600);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  dht_sensor.begin();
}

void loop() {
  temperatureHumiditySensor();
  Blynk.run();
}

void temperatureHumiditySensor() {
  float humidity = dht_sensor.readHumidity();
  float temperature = dht_sensor.readTemperature();

  if (isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
    if (temperature > TEMP_UPPER_THRESHOLD) {
      digitalWrite(RelayFan, LOW);
      Blynk.logEvent("temp");
      } else if (temperature < TEMP_LOWER_THRESHOLD) {
        digitalWrite(RelayFan, HIGH);
      }
  }
  Blynk.virtualWrite(V2, temperature);
  Blynk.virtualWrite(V3, humidity);
}
```

Ultrasonic Sensor with Piezo Buzzer

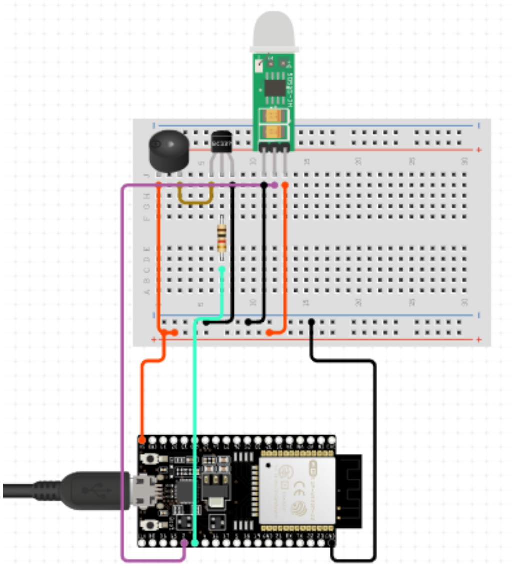


Figure 4.12 Ultrasonic Sensor with Piezo Buzzer

Figure 4.12 shows the diagram for ultrasonic sensor with piezo buzzer. This Arduino code utilizes an ESP32 microcontroller with WiFi capabilities to connect to the Blynk IoT platform. It includes libraries for WiFi and Blynk and defines macro values for Blynk configuration. The setup function sets up the pins, initializes serial communication, and connects to Blynk. The loop function continuously checks the motion sensor state and runs the Blynk communication. The motionSensor function checks if the motion sensor detects movement and triggers the buzzer accordingly, while logging motion events to Blynk. Overall, this code enables the ESP32 to connect to Blynk, monitor motion with a sensor, and control a buzzer based on the detected motion.

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPLCr7cIo9X"
#define BLYNK_TEMPLATE_NAME "Temperature Humidity"
#define BLYNK_AUTH_TOKEN "Ib7yg_E6wqBSJvCFrdN0NqSNKWKJeTXQ"
#define PIR 5
int motionStateCurrent = LOW;
int motionStatePrevious = LOW;
```

```
#define BUZZER_PIN 18

char ssid[] = "Harith Aizat";
char pass[] = "smartgarden";

void setup() {
pinMode(PIR, INPUT);
digitalWrite(PIR, LOW);
pinMode(BUZZER_PIN, OUTPUT);
Serial.begin(9600);
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop() {
  motionSensor();
  Blynk.run();
   }

void motionSensor() {
  if (digitalRead(5) == HIGH) {
    digitalWrite(BUZZER_PIN, HIGH);
    Serial.println("Someone Here");
    Blynk.logEvent("motion_detected");
  } else {
    Serial.println("Nobody Here");
    digitalWrite(BUZZER_PIN, LOW);
   }
}
```

4.4 Conclusion

In conclusion, the IoT smart garden and automatic irrigation system can greatly benefit gardeners by providing an efficient and convenient way to manage their plants. The sensors and ESP32 microcontrollers were connected using jumper wires. The sketches for each sensor were uploaded to the Arduino IDE to enable their functionality. With sensors to monitor temperature, humidity, and soil moisture, as well as the ability to automatically water the plants when needed, this system can help ensure that plants receive the optimal conditions for growth.

The system can also send alerts or notifications to gardeners when certain conditions are met or when the water level is low. Additionally, the ability to remotely control the system through a mobile application or web interface provides even more convenience and flexibility for users. While the initial setup and cost of the system may require some investment, the long-term benefits in terms of plant health and efficiency may make it a worthwhile investment for serious gardeners or commercial farms. Overall, the IoT smart garden and automatic irrigation system has the potential to revolutionize the way we grow plants and manage our gardens.

CHAPTER 5

CONCLUSION

5.1 Introduction

This project aimed to address the challenges associated with water management in agriculture through the implementation of an Internet of Things (IoT) smart garden and automatic irrigation system. The system was designed to provide an efficient and autonomous solution for watering plants based on their specific moisture needs, ultimately reducing water waste, and improving crop yields. Through the study of existing work and current issues in smart gardens and irrigation systems, the project successfully developed an Internet of Things (IoT) smart garden system. The system incorporated various platforms, including smartphone apps and web-based interfaces, allowing users to control and monitor their smart gardens remotely from any device with internet access.

By utilizing sensors to measure soil moisture, humidity & temperature and a PIR motion sensor, the smart garden system provided accurate data to gardeners, enabling them to make informed decisions and optimize their gardening techniques. This not only improved the efficiency of water usage but also contributed to better plant growth and the overall gardening experience. The project's objectives were achieved through extensive research, learning, and the implementation of IoT concepts and technologies. The system underwent user acceptance tests to evaluate its effectiveness, ensuring that it met the requirements and expectations of the users.

5.2 Constraints

During system implementation, various constraints need to be considered, such as hardware limitations, time restrictions, and technical requirements.

5.2.1 Hardware Constraints

Difficulties were encountered during the setup of multiple sensors, including a soil moisture sensor, a temperature and humidity sensor, and a PIR motion sensor. The sensors' readings were found to be inaccurate during testing. The ESP32 microcontroller experienced a short circuit caused by an imbalance in voltage supply during the programming process. As a result, the ESP32 became unusable for programming purposes. Despite encountering various hardware constraints, all the affected hardware components were successfully replaced, and the voltage supply issue was resolved before the proposal submission deadline.

5.2.2 Time Constraints

While the system was implemented on time, there is still room for improvement in its functionality and performance. The interface in Blynk application was designed with basic essential functions, but there are additional features that can be implemented in the future. Due to time constraints during the system implementation, these improvements have been included in the discussion of future work for the system.

5.2.3 Cost Constraints

The required hardware for this system proved to be costly for students, especially with the need for repeated purchases of components such as soil moisture sensors, humidity sensors, and temperature sensors due to replacements. This increased the overall expenses associated with completing the system.

5.2.4 Technical Constraints

Implementing this system involved learning new concepts such as the Internet of Things (IoT) and Blynk applications. The lack of prior knowledge in these areas added to the challenge of researching and learning programming languages for Arduino, understanding Blynk applications, and setting up the ESP32 with the sensors. There were numerous initial failures, but through extensive research and learning about the concept and implementation process of an IoT Smart Garden, the system was eventually completed.

5.3 Future Work

Future work on the system is crucial for enhancing its robustness and efficiency. The system currently lacks several functionalities, leaving significant room for improvement. In this section, the focus will be on outlining the future work that can be done to enhance the system. This includes adding additional functionalities to further enhance its capabilities. Future recommendations for enhancing the system's functionality and performance include:

i. Integration of additional sensors: Explore the incorporation of more sensors, such as light intensity sensors or nutrient level sensors, to provide comprehensive monitoring and control capabilities for optimal plant growth.

ii. Enhanced user interface: Continuously improve the user interface of the smartphone apps and web-based interfaces to enhance usability, accessibility, and user experience.

iii. Data analytics and insights: Implement advanced data analysis techniques to derive meaningful insights from the collected data, enabling gardeners to make data-driven decisions and optimize their gardening practices further.

iv. Smart irrigation scheduling: Develop intelligent algorithms and machine learning models to create personalized and adaptive irrigation schedules based on factors like plant type, weather conditions, and seasonal variations.

v. Integration with smart home systems: Explore integration possibilities with existing smart home systems to enable seamless control and automation of the smart garden along with other household devices.

By implementing these future recommendations, the IoT Smart Garden and Automatic Irrigation System can be further enhanced, leading to increased efficiency, sustainability, and convenience for users.

REFERENCES

Bonnie. (2022, October 14). *The Importance Of Water Soil And Sunlight In Plant Growth And Development – SC Garden Guru*. https://www.scgardenguru.com/2022/10/14/the-importance-of-water-soil-and-sunlight-in-plant-growth-and-development/

Gargee Sarkar, Shubrangsu Chaki, Suja Uddin Mistry, & Sourish Roy. (2018). *Arduino Based Automatic Plant Irrigation System With Message Alert*.

h2ouse. (2022). *Why Save Water? - H2OUSE*. https://www.h2ouse.net/why-save-water/

Heather Rhoades. (2022). *Gardening Know How Signs Of Plants Affected By Too Much Water-Gardening Know How If You Overwater a Plant, Will It Still Grow?* https://www.gardeningknowhow.com/plant-problems/environmental/plant-leaves-turn-

Maheswari, S. U., & Bhuvaneswari, P. T. V. (2021). IoT Enabled Smart Irrigation System for Mini Gardens. *3rd IEEE International Virtual Conference on Innovations in Power and Advanced Computing Technologies, i-PACT 2021*. https://doi.org/10.1109/i-PACT52855.2021.9696598

Obaideen, K., Yousef, B. A. A., AlMallahi, M. N., Tan, Y. C., Mahmoud, M., Jaber, H., & Ramadan, M. (2022). An overview of smart irrigation systems using IoT. *Energy Nexus*, *7*, 100124. https://doi.org/10.1016/j.nexus.2022.100124

Raju Chellam. (2020, September 29). *Econ 4.0: Why focus on farming? | The Edge Markets*. https://www.theedgemarkets.com/article/econ-40-why-focus-farming

SDLC - RAD Model. (n.d.). Retrieved December 2, 2022, from https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm

Sojka, R. E., Bjorneberg, D. L., & Entry, J. A. (2022). *0-7 0 IRRIGATION: AN HISTORICAL PERSPECTIVE*.

Swaroop, A., Sharma, V., Galgotias University. School of Computing Science and Engineering, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, & Institute of Electrical and Electronics Engineers. (n.d.). *International Conference on Computing, Communication and Automation : on 15-16 May, 2015*.

Woofter. (2022). *The Importance of a Proper Irrigation System | Woofter Construction & Irrigation | Woofter Construction & Irrigation*. https://www.woofter.com/updates/importance-of-proper-irrigation/

APPENDIX A
GOOGLE FORM

Time Spent on Garden Maintenance

Good day everyone,
I am a final year students of Bachelor of Network Engineering from University Malaysia Pahang. I am conducting a survey to gather information about time spent on garden maintenance

I hope you may spend some of your precious time to answer my questionnaires. This survey will be treated as anonymous, confidential and purely for educational purposes. Your kind cooperation in helping us is highly appreciated. Thank you!

m.harithaizat@gmail.com (not shared) Switch account

How often do you currently water your garden?

- ○ Once a day
- ○ Every other day
- ○ Once a week
- ○ Once a month

Is maintaining and taking care of your garden exhausting?

- ○ Yes
- ○ No

Do you have time to regularly check your garden?

- ○ Yes
- ○ No

How do you currently take care of your garden when you go on holiday?

- ○ I have a neighbor or family member who takes care of my garden while I am away.
- ○ I have an automatic irrigation system that takes care of my garden while I am away.
- ○ I hire a professional gardener to take care of my garden while I am away.
- ○ I don't take any special measures and leave my garden to fend for itself.

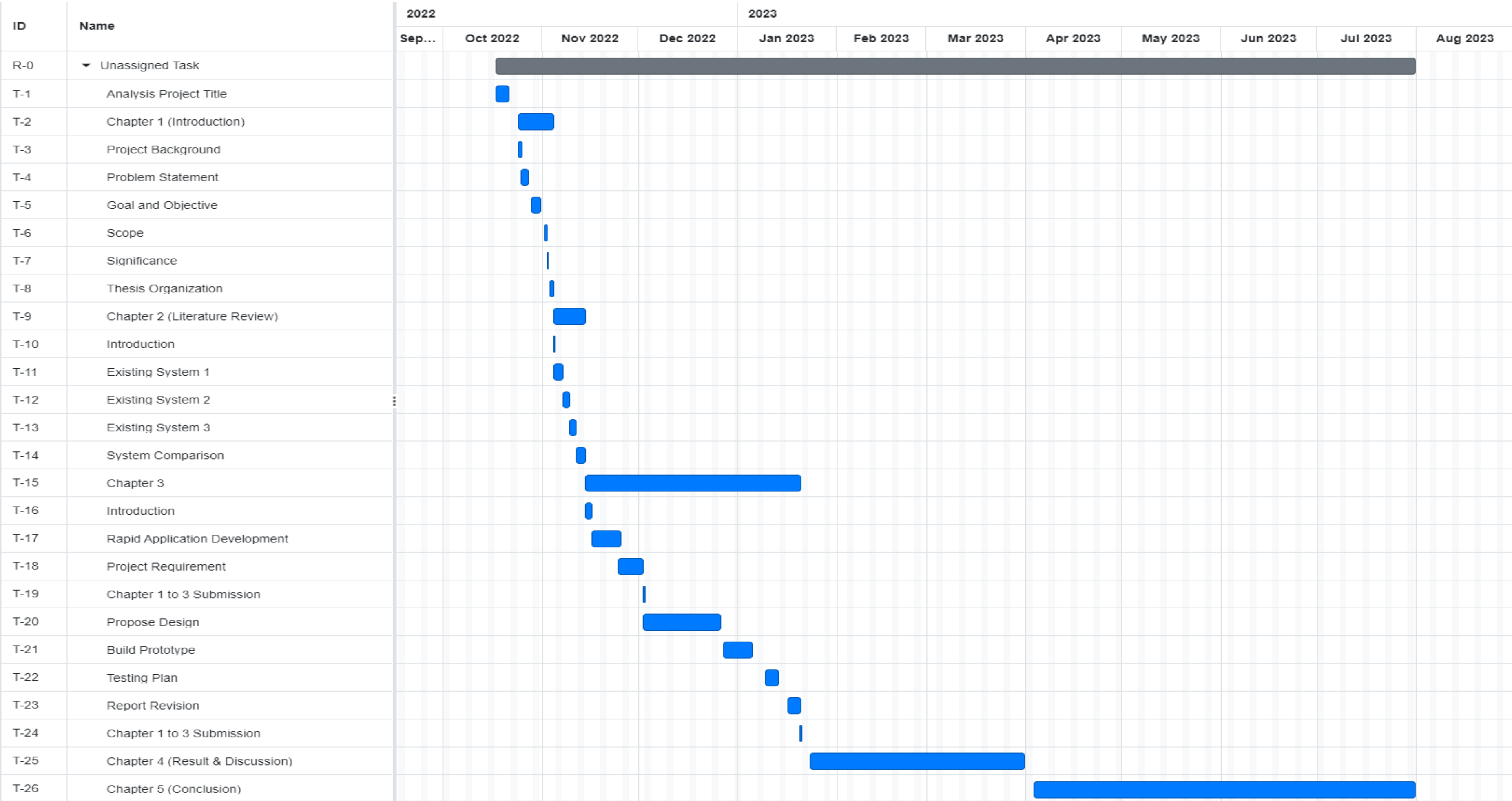
Do you think it is important to have automatic irrigation system for your garden?

- ○ Yes
- ○ No

Submit

Clear form

APPENDIX B
GANTT CHART



APPENDIX C
USER ACCEPTANCE TEST (UAT)

1.0 Blynk Application Testing

Table 10 Blynk Application Testing

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass / Fail**	**Comment**
Login with correct email and password	Email: m.harithaizat@gmail.com Password: 123456	Login successful and redirect to homepage	Able to login successfully and redirect to the My Devices Page	Pass	
Login with incorrect username and password	Email: adlansyahin@gmail.com Password: 11223344	Login unsuccessful and error message pops-up and returns login page	The error message is popping out when incorrect username and password is key in	Pass	
When My Devices is clicked	Null	The interface should consist of My Devices.	The My Devices show Soil Moisture and Temperature Humidity	Pass	

			devices name		

2.0 Sensors, Hardware, and Database Connection

Table 11 Sensors, Hardware, and Database Connection Testing

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When power "On"	Null	Proposed system is active and send reading	The system is active and send the reading	Pass	
Soil moisture level less than 60%	Null	Active the water pump	Water Pump is on	Pass	
Soil moisture level more than 80%	Null	Stop the water pump	Water Pump is off	Pass	
Temperature level is less 33	Null	Stop the Fan	Fan is off	Pass	
Temperature level more than 35	Null	Active the Fan	Fan is on	Pass	
ESP32 when power is "On"	Null	LED light is on	LED is on	Pass	
Database when power "On"	Null	Connection is successful with ESP32 board	Connection is successful	Pass	

3.0 Graph Section Testing

Table 12 Graph Section Testing

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When soil moisture graph button is clicked	Null	The soil moisture graph will show on the interface.	The soil moisture graph is show	Pass	
When temperature graph button is clicked	Null	The temperature graph will show on the interface.	The soil temperature graph is show	Pass	
When humidity graph button is clicked	Null	The humidity graph will show on the interface.	The soil humidity graph is show	Pass	

4.0 System Approval

Table 13 System Approval

Signature	Name	Date
Verified by: 	MOHAMAD HARITH AIZAT BIN SUHAILI	10 JULY 2023
Verified by 	AHMAD HISYAM BIN SURYANTO SUGIAN	10 JULY 2023