SMART AGRICULTURE'S IOT APPLICATION FOR SUSTAINABLE PLANT GROWTH

CHONG ZI JIAN

A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Electronics Engineering with Honours

Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

May 2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : Chong

Name : Chong Zi Jian

ID No. : 1904784

Date : 15 April 2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"SMART AGRICULTURE'S IOT APPLICATION FOR SUSTAINABLE PLANT GROWTH"** was prepared by **Chong Zi Jian** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Electronics Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature :

Supervisor : Dr. Toh Pek Lan

Date : 22 April 2024

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Toh Pek Lan for her invaluable advice, guidance, and his enormous patience throughout the development of the research. When I encountered a problem, she was patient, friendly, and effective in giving me appropriate help to complete my project. My beloved parents and my friends, who have supported and encouraged me, also deserve my appreciation.

SMART AGRICULTURE’S IOT APPLICATION FOR SUSTAINABLE PLANT GROWTH

ABSTRACT

Since the pace of life is becoming faster, many people may not remember or lazy to take care of their plants. Furthermore, one of the elements affecting the plant's growth process is the weather. It may lead to the process of photosynthesis not able to progress and affect the temperature and humidity of the environment of plants. To overcome this problem, the smart plant monitoring system is promoted in this project. It can help the user to monitor the condition of the plant automatically. Then, this can prevent the case of overwatering or no watering even if the soil is dry. Next, the monitoring also includes temperature control which will decrease the temperature of the environment of the plant if the temperature of the plant environment is higher than the pre-setting temperature. To ensure the process of photosynthesis can be conducted, the monitoring system also will determine the intensity of light in the surrounding of the plant and the sunlight will be replace by full spectrum of light if the intensity of light in the plant environment is low or dark. Through this monitoring system, it can help the user to take care of their plants in a healthy growing environment.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

SSR	Self-sufficiency ratio

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Nowadays, driven by technological progress, the agricultural field is undergoing a transformative and pressing need to solve issues caused by resource scarcity, whether change, and the rising demand for food worldwide. Due to the COVID-19 pandemic, Malaysia's self-sufficiency in food items has been impacted. In 2020, only 19 food items were produced domestically, compared to 25 in 2019. Consequently, the self-sufficiency ratio (SSR) of food items has fallen below 100%, making it necessary to import from other countries (Lim, 2021). This may result in an increase in food prices. This is because agriculture is essential to preserving ecological equilibrium and maintaining the supply of food, new methods are being tested to improve plant development, improve yields, and maximize resource use. The smart plant monitoring system is one of the methods that is changing plant growth and maintenance through the combination of automation, data analytics, and advanced sensors. A common problem faced by a plant owner is forgetting to water their plants on time or accidentally overwatering them. To overcome this problem, a solution is provided by the installation of an automated watering system. This device uses advanced technologies to determine the percentage of moisture in the soil and only sends water when needed. It will help plant owner save the trouble of doing it by hand. The automation reduces the work for water plants on time, and then reduces the chance of overwatering, a common mistake that will damage the health of the plants. Providing a correct amount of moisture to the plants, will reduce the waste of water and save the time of the plant owner. In certain periods, the temperature of the environment will be slightly higher than the normal temperature. If the plant condition is dry, it may cause the plant to burn or have possible damage to the plants. When the plant is exposed to dry conditions such as a lack of water in a dry environment. It is unable to manage high temperatures and more subject to burning or other damage. The high temperatures can cause wilting, burning of the leaves, or even lifelong tissue damage, all of which may decrease a plant’s vitality and general health. Furthermore, plants require sunlight as one of the main sources of energy for photosynthesis, which is needed for their life and growth. However, the amount of sunlight available may reduce during bad weather, such as cloudy or rainy days. It will affect the ability of plants to perform photosynthesis. This is because the process of photosynthesis is proportional to the intensity of light (Reckitt Benckiser, 2021). Plant owners can maintain the health of their plants throughout changes in the outside weather by understanding the importance of light intensity in photosynthesis and putting a suitable method into place.

1.2 Problem Statements

The challenge of maintaining ideal circumstances for plant growth, such as routine watering, temperature regulation, and intensity of light in the external environment, is currently the issue faced by plant owners. This frequently results in plants being either overwatered or underwatered, exposed to severe soil conditions, or both. These difficulties lead to poor plant growth and health. The use of a Raspberry Pi Pico W-based smart plant monitoring system is suggested as a solution to this issue. The system's goals include automating the watering of plants depending on soil moisture levels, controlling temperature to minimize the risk of plants burning, and monitoring the intensity of light in the environment to ensure the plant’s growing condition. In addition, users will also have access to data analysis tools through the Blynk software, enabling them to monitor and study environmental conditions that have an impact on plant health. Users will be able to conserve time and water by putting this smart plant monitoring system in place, protect plants from damage caused by severe temperatures, and keep the intensity of light to make sure the process of photosynthesis will not be affected by weather such as cloudy or rainy days.

1.3 Aim And Objectives

The aim of implementing the smart plant monitoring system with the Raspberry Pi Pico W is to simplify the plant watering process through automation. Based on soil moisture levels, provide a cooling system to maintain a safe environment for the plants during high temperatures, detect and adjust the intensity of light to ensure the process of photosynthesis can be conducted, and allow users to analyze environmental data such as temperature, humidity, intensity of light, and moisture levels using the Blynk software. Belows are the objectives of this project:

a) To design a smart plant monitoring system using Raspberry Pi Pico W.
b) To ensure the monitoring system able to read data by Blynk.
c) To ensure the plant can growth in healthy conditions.

CHAPTER 2

LITERATURE REVIEW

2.1 Raspberry Pi Pico W

A single-board computer called the Raspberry Pi was first developed in 2012 by the British nonprofit Raspberry Pi Foundation. Operating systems other than Linux can be run on the Raspberry Pi boards. Various uses include media centers, automated homes, robots, and more (Whitmore, 2022). The Raspberry Pi 4 Model B, Raspberry Pi 3 Model B+, Raspberry Pi Zero W, Raspberry Pi Zero 2W, and Raspberry Pi Pico are the most widely used Raspberry Pi models on the market (Obias, 2023). The Raspberry Pi Pico W and Raspberry Pi Pico are the same model, except this one has built-in WiFi. It is so suitable for the IoT project. In June 2022, Raspberry Pi introduced the Pico W, a powerful and low-cost microcontroller developer board. With built-in wireless networking, it is a better version of the initial Pico board. The Pico W comes with 40 pins, three analog inputs, and a Programmable I/O subsystem. It is powered by the Raspberry Pi's RP2040 system-on-chip (SoC) with dual-core Arm Cortex-M0+ processors. The onboard Wi-Fi functionality is its key benefit, enabling users to join their Wi-Fi network, communicate with other devices, and browse the internet. The Pico W is appropriate for a range of Internet of Things (IoT) projects since it supports common digital communication protocols and has programmable I/O state machines. It may be developed in C/C++ or MicroPython, and due to its small size, it is perfect for projects that require little room. Both developers and hobbyists will find the Pico W to be a flexible and cost-effective alternative (King, 2022).

2.2 Compare between other types of microcontrollers.

2.2.1 Raspberry Pi Pico W Compare with Arduino

The Raspberry Pi Pico W triumphs over Arduino in several aspects when the two devices are compared. It is adaptable for a variety of projects because of the vast range of functionality and GPIO possibilities it provides, including programmable IO pins. In comparison to Arduino boards, the Pico W's dual-core Arm Cortex-M0+ processor offers faster processing times and more storage. Additionally, the Pico W offers flexibility in coding options by supporting both C/C++ and MicroPython. Its usability, particularly with MicroPython, and decreased power usage are additional draws (Pounder, 2021).

2.2.2 Raspberry Pi Pico W Compare with ESP32

The Raspberry Pi Pico W is a microcontroller board built around the RP2040 chip, while the ESP32 microcontroller replaces the ESP8266 microcontroller. The Raspberry Pi Pico W has 26 programmable GPIO pins, while the ESP32 has 34. Both boards have dual-core CPUs. The Arm Cortex-M0+ dual-core processor powers the Raspberry Pi Pico W, whereas the Tensilica Xtensa LX6 32-bit dual-core processor powers the ESP32. The ESP32 has a CPU that is somewhat more powerful and operates at a faster clock speed (80/160/240MHz) than the Raspberry Pi Pico W (133MHz). The programming languages that are supported by both boards are C, C++, and MicroPython. The Raspberry Pi Pico W can be programmed using drag-and-drop because it appears as mass storage when connected to a PC via USB. The ESP32 supports MicroPython and C++ for smaller applications; however, larger projects should use the ESP-IDF (Espressif IoT Development Framework) with an Eclipse plugin or Visual Code extension. The Raspberry Pi Pico W is an excellent board for both novice and seasoned Raspberry Pi users. It is easy to use and straightforward to get started with. Although despite having a somewhat more challenging learning curve, experienced users can still utilize the ESP32. Both boards provide power-saving features to lower power consumption. The ESP32 has multiple power modes, including hibernation mode, which uses less than 5µA, while the Raspberry Pi Pico W needs approximately 91mA in the popcorn test (Olujinmi, 2022). The project's specifications and budget will determine which microcontroller is used. For those on a tight budget or just starting, the Raspberry Pi Pico W is a good option; otherwise, the ESP32 is the alternative option.

2.3 Sensor

2.3.1 Soil Moisture Sensor

Soil moisture sensors are electrical devices that measure the total amount of water in soil. A resistive soil moisture sensor consists of two probes that allow electricity to pass through the soil to measure resistance corresponding to the soil moisture level. More water in the soil allows more current to flow between the sensors, resulting in a lower resistance value. On the other hand, when there is less water present, the soil shows increased resistance and lower conductivity (Technicals, 2021). Unlike usual moisture sensors, which use resistive sensing to measure the moisture level of the soil, capacitive soil moisture sensors use capacitive sensing to measure soil moisture levels. Resistive sensors identify the concentration of ions in the soil, whereas capacitive sensors monitor changes in the dielectric constant or the amount of water in the soil directly. It can store more charge on its plates when there is more water in the soil, which raises the capacitance value. The soil retains less charge on its plates when there is less water present, which lowers its capacitance value (Biomaker, 2021). Although capacitive sensors are more expensive than resistive sensors, it has better measurements of the soil's moisture content and resistant to corrosion compared to the resistive sensor.

2.3.2 DHT-22

The DHT22 is a typical temperature and humidity sensor that features a dedicated NTC for temperature measurement and an 8-bit CPU that outputs the temperature and humidity measurements as serial data. The sensor generates digital data and is factory-calibrated, making interfacing with other microcontrollers straightforward. The DHT22 can measure temperature in the range of -40°C to 80°C and humidity in the range of 0% to 100% with an accuracy of 1°C and 1%. The finding that the capacitance of the humidity sensing element varies with the moisture content of the air forms the basis of the operational theory of the DHT22 sensor. The temperature of the air is measured by the thermistor, which is used to calculate the relative humidity. A compared sensor, the DHT11, has an accuracy of ±1°C and ±1% and can measure temperature from 0°C to 50°C and humidity from 20% to 90% (Ada, 2021). As a result, the DHT22 costs a little more than the DHT11, but it can measure a larger range and is slightly more accurate.

2.3.3 Light Dependent Resistor (LDR) sensor

A photoresistor is also called a light-dependent resistor (LDR). It is one of the types of resistors that operates on the principle of photoconductivity. It means that the resistor is dependent on the light intensity. Photosensitive semiconductor materials such as cadmium sulfide (CdS), lead sulfide, etc. are used to produce an LDR sensor. Since the LDR sensor is working on the principle of photoconductivity, it means that the decrease of the resistance is affected by the electrons moving from the surface to the conduction band when the light hits the photoconductive material. This phenomenon occurs due to the light and has enough energy pass through the bandgap of the material. When the light intensity is low or close to zero, the LDR has the highest resistance (approximately 10^{12} ohm). In the other way, the resistance decreases if the intensity of light increases. Therefore, it allows more current to pass through. According to its theory of operation, the sensor's resistance and light intensity are inversely related. Two types of LDR are available for the LDR sensor: intrinsic and extrinsic LDR. For intrinsic LDR, it is produced by a pure semiconductor such as silicon and germanium. When the light is presented, the electron will absorb the energy and transition to the conduction band to reduce the resistance. Extrinsic LDR is produced by a doped semiconductor like phosphorus. This type of LDR is used for longer light wavelengths (Rathore, 2020).

2.4 Plant Growth and Cultivation

The environment has a big impact on how plants develop and where they are found. Any condition affects the growth and dispersal of a plant. The article states that light, temperature, water, humidity, and nutrition are the five environmental factors that control plant growth (Neina, 2019). Soil moisture is one of the most important aspects of a soil's physical makeup that influences plant growth. The quantity of water left in the soil after any surplus has been drained away is known as soil moisture, and the texture of the soil affects how quickly water runs through it. Then, both plants and soil organisms are impacted by the temperature. The temperature range in which typical soil organisms can be active is roughly 0°C to 60°C, while no one species is likely to be active over the full range (Food and Agriculture Organization, n.d.). Plants require light for growth because it is an important component of photosynthesis, transforming light, water, and oxygen into the carbohydrates that provide plants with energy. Low levels of light intensity cause plants to produce fewer carbohydrates, eventually running out of energy and dying. If there are not enough light causes plants to decrease chlorophyll, which results in sallow leaves. Moreover, older, or mixed types may return to solid green as plants shed their leaves. Without enough light intensity present, flowering plants may not be able to form buds. On the other hand, too much light can damage the plants by causing their leaves to burn and become whiter. To nurture healthy plants, it is necessary to provide a balanced light environment (University of Minnesota Extension, 2020). Plant growth and development are affected by temperature, which also affects some biological functions in plants. Each type of plant has a different ideal temperature range if excesses the ideal temperature can be damaged to the plant. Warmer temperatures generally will speed up the processes like respiration and photosynthesis, which enable faster rates of growth (UMass Amherst, 2015). Further, heat stress caused by extreme temperatures may damage plant tissues and result in wilting and decreased yields (eos.com, 2023). On the other hand, the biological processes and the growth and development of plants will be slowed down due to the low temperature. If the temperature reaches 0 °C or lower freezing and frost will occur. It will cause damage to the plant cells, which leads to withering, yellowing leaves, and death of the plant (Pennisi *et al.*, 2022). In general, keeping a suitable temperature is important for increasing the productivity of the plant, and the taking care of the health of the plant.

2.5 IoT Plant Monitoring System

Deepan *et al.*, in 2022, suggests an Internet of Things (IoT)-based smart agriculture concept that uses a Penta-sensor to keep track of the water quality during irrigation. The system provides an Internet of Things-based facility agriculture ecosystem built on a smart monitoring platform and system architecture. The authors argue that promoting advances in technology in agriculture will help with several problems that farmers face, like inaccurate data and communication that lowers productivity. The research emphasizes that there are a few additional factors that have a higher negative impact on production than environmental monitoring and that environmental monitoring is not the only way to boost agricultural yield. Automation in agriculture is necessary to solve these problems. They offer an independent device that uses adjustable, cost-effective, and easily developed hardware platforms, which has enabled IoT to function as a tool for the objective of autonomous, self-organized decision-making in the farming and agriculture sectors. Among the primary uses, according to the authors, are crop production process management, crop growth optimization, automated irrigation scheduling, monitoring of farmland, and precision agriculture. By referring to an article, the IoT-based smart plant monitoring system is designed to optimize irrigation and increase the productivity of the crops. It contains sensors for temperature, humidity, and soil moisture monitoring to identifying external factors like barriers. NodeMCU ESP8266, DHT11 Temperature Sensor, Soil Moisture Sensor, and Proximity Sensor are important parts to perform the system. The system sends and displays data using the Blynk software. In comparison between monitored plants and non-monitored plants, the monitored plants keep higher moisture levels, lower humidity, and stable temperatures based on the study results from a week-long experiment. After that, water saving, low cost, and real-time monitoring through mobile devices are some benefits of the system. In general, it offers significant perspectives on improving agricultural productivity and environmental health (Nehra *et al.*, 2023). Finally, referencing another article, a smart plant watering system that automates the process of watering plants and minimizes water waste by using IoT sensors and cloud-based databases. The technology is made to irrigate plants according to their needs, minimizing human intervention and preventing water waste. To monitor the soil conditions of the plant, the system makes use of many modules, including an Arduino as a controller, temperature, wetness, humidity, and PH sensors. Different types of sensors analyze the sensor values, and the water motor will be turned on or off depending on the result. Through mobile devices or laptops, the system gives the user input. The automated system decreases the demand for labor, which also decreases mistakes. The farmer's smart devices make it simple for them to keep an eye on the system (Barhate *et al.*, 2020).

CHAPTER 3

METHODOLOGY

3.1 Project Management

Figures 3.1 and 3.2 present the final year project (FYP) schedule, which help to make sure all tasks can be completed on time. In FYP part 1, all the tasks were completed successfully, and the project moved forward as a whole. For example, the project proposal was turned in during the first week, and the supervisor first gave her approval for the project title. The literature review was started to prepared in week two, right in line with the Python programming language study. Furthermore, the project equipment purchase process started in Week 6 following the completion of the literature study and all items were delivered by Week 7. After obtaining the required tools, the prototype gadget was built between weeks eight and fourteen. The reporting process started in week nine and ended in week twelve at the same time. Finally, in Week 14, the oral presentation preparations started. The installation of a water level sensor, a light-detection system, and the assembly of the project's plan constitute the next steps in the development of FYP part 2. Week 12 marks the submission of the final year project report following that. And last, in Week 13, there is an oral presentation.

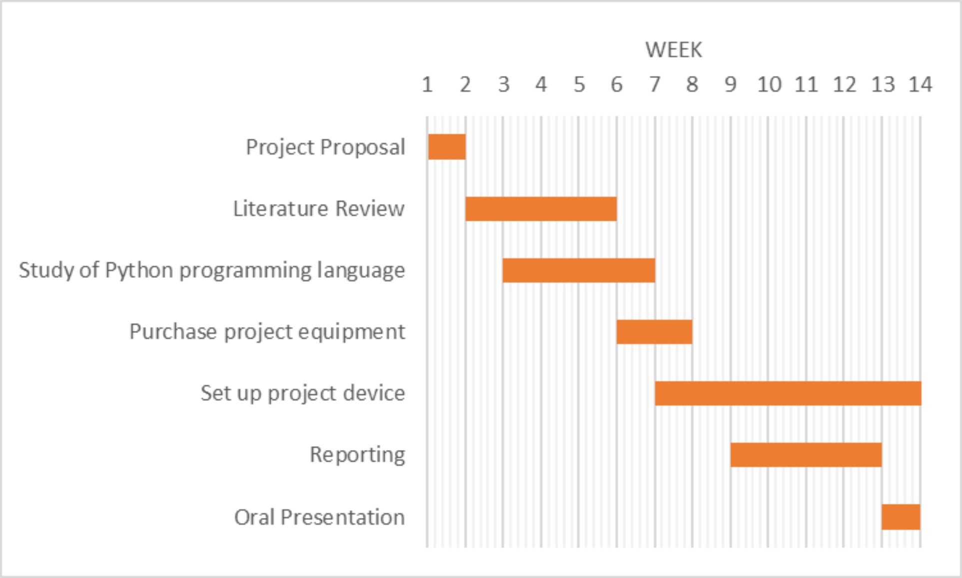

Figure 3.1: Gantt Chart for Final Year Project 1

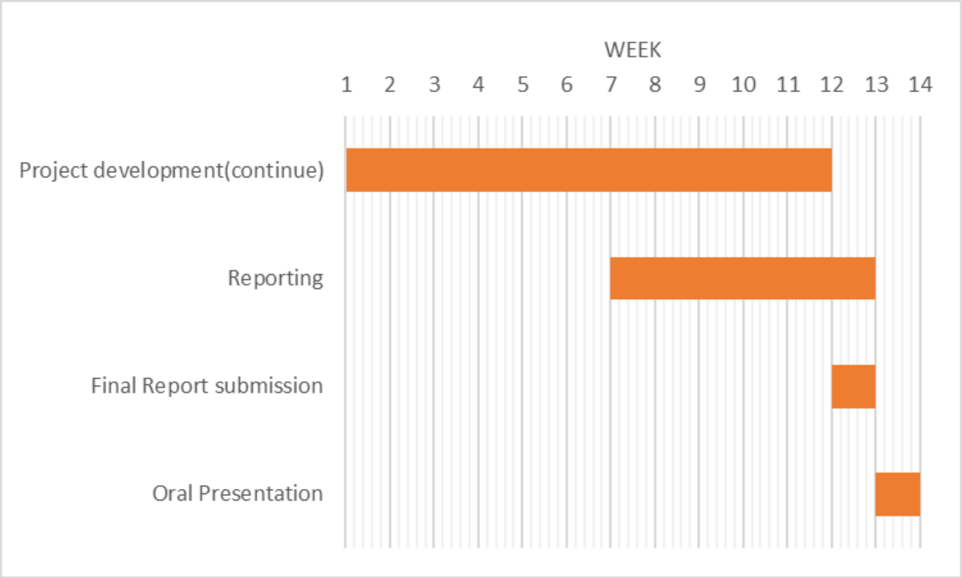

Figure 3.2: Gantt Chart for Final Year Project 2

3.2 Requirements for Hardware and Software

3.2.1 Raspberry Pi Pico W

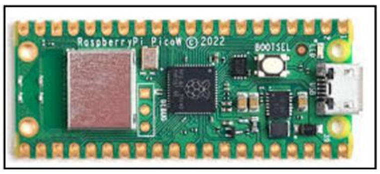

Figure 3.3: Raspberry Pi Pico W

In order to create the project device, the Raspberry Pi Pico W's input/output pins (GPIO) were used. In addition to being used for input and output, the GPIO pins on the Raspberry Pi Pico W board (Figure 3.3) can also be used to connect I2C devices, read and write data from other devices, and handle other tasks. Wi-Fi functionality was required to turn the project into an Internet of Things gadget. Later, it might utilize different programming languages, such as Python, C, or C++, however, for this project, Python code was applied. Table 3.1 displayed the features and specs of the Raspberry Pi Pico, whereas Figure 3.4 displayed the pinout of the Raspberry Pi Pico W.

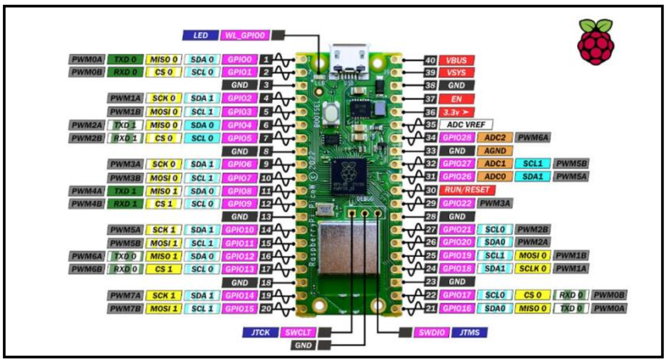

Figure 3.4: Pinout of Raspberry Pi Pico W (Mischianti, 2022)

Table 3.1: Feature and Specifications of Raspberry Pi Pico W

Feature/Specification	Value
CPU	Dual-core Arm Cortex M0+ processor
Clock Speed	Up to 133 MHz
RAM	264kB of SRAM
Flash Memory	2MB of on-board flash memory
Operating Voltage	1.8–5.5V
GPIO Pins	26 multi-function GPIO pins
Wi-Fi	Single-band (2.4 GHz) wireless interface
Bluetooth	Bluetooth 5.2
Memory	2MB of on-board flash memory
Operating System	None (microcontroller)
USB	USB 1.1 with device and host support
Operating Temperature	-20°C to +85°C
Power Input	1.8–5.5V

3.2.2 Resistive Soil Moisture Sensor Module

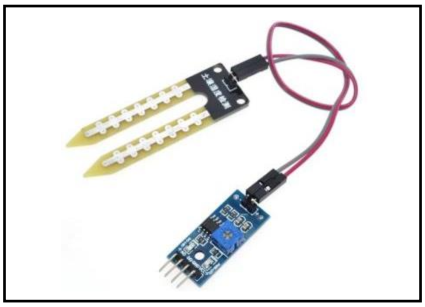

Figure 3.5: Resistive Soil Moisture Sensor Module

To find out how moist the soil was, the resistive soil moisture sensor module was applied. Blynk (IoT control software) will then record the data after the microcontroller has read it. Table 3.2 provides specifications for the resistive soil moisture sensor module.

Table 3.2: Specification of Resistive Soil Moisture Sensor Module

Specification	**Value**
Operating Voltage	3.3V to 5V DC
Operating Current	15mA
Output Digital	0V to 5V, adjustable trigger level from preset
Output Analog	0V to 5V based on infrared radiation from fire flame falling on the sensor
LEDs indicating output and power	Available

3.2.3 Temperature and Humidity Sensor (DHT-22)

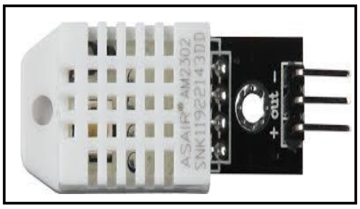

Figure 3.6: DHT-22 Sensor

The temperature and humidity of the surrounding air were measured using the DHT-22 sensor, which can be observed in Figure 3.6. After that, Blynk will record the data and the microcontroller will read it. Table 3.3 displays the specifications of the DHT-22 sensor.

Table 3.3: Specifications of DHT-22 Sensor

Specification	**Value**
Operating Voltage	3.5V to 5.5V DC
Operating Current	0.3mA (measuring stage), 60uA (standby stage)
Output	Serial data
Temperature Range	-40°C to 80°C
Humidity Range	0% to 100%
Accuracy	±0.5°C and ±1%

3.2.4 Water Pump

Figure 3.7: 5V Water Pump

The water was pumped via the water pipe from the container to the soil using the water pump depicted in Figure 3.7. The 5V power supply is required in order to run the water pump.

3.2.5 Fan

Figure 3.8: 5V Fan

In Figure 3.8, the diagram shows the fan was used for the cooling system of the project. Its function was decreasing the temperature of the environment when the temperature of the environment was high.

3.2.6 Liquid Crystal Display (LCD) with I2C Module

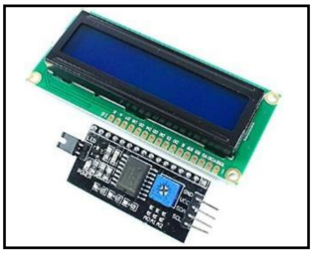

Figure 3.9: Liquid Crystal Display (LCD) with I2C Module

Based on diagram in Figure 3.9, the LCD was used for displaying the reading of the sensor or the stage of the system. Its dimensions were 16 columns x 2 rows. The I2C module was used to simplify the wiring between the LCD and the microcontroller. It can reduce the usage of the GPIO pin from 16 pins to 4 pins.

3.2.7 Relay Module

Figure 3.10: 5V Relay Module

A relay shown in Figure 3.10 acts as a switch that can be turned on or off by a digital signal. The relay was used for turning on or off the fan in the cooling system, the water pump in the watering system, and also the growing light.

3.2.8 Growing Light

Figure 3.11: Growing Light

The growing light shown in Figure 3.11 is used to help the plant stimulate the photosynthesis at night, cloudy or raining day. It is a full spectrum of growing light so that it can produce the light in the range of the wavelength from 400nm to 700nm. Therefore, it is suitable for used to replace the sunlight if the sunlight is absent.

3.2.9 Light Dependent Resistor (LDR) Sensor Module

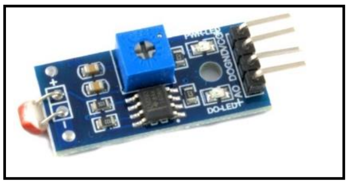

Figure 3.12: LDR Sensor Module

A light detector sensor is the sensor module that is depicted in Figure 3.12. Its job is to measure the ambient light intensity and provide the microcontroller with a digital output. The resistance of the LDR is low in the presence of light and high in the absence of it.

3.2.10 Water Level Sensor

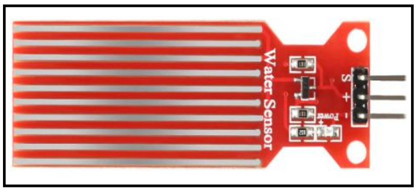

Figure 3.13: Water Level Sensor

The water level sensor is used to determine the container's water level, as seen in Figure 3.13. The analog output will be generated and sent to the microcontroller. When the water level is high, the device functions as a potentiometer due to its high conductivity and low resistance. On the other hand, poor conductivity and high resistance occur when the water level is low.

3.2.11 Light-Emitting Diode (LED)

Figure 3.14: Light-Emitting Diode (LED)

The LED show in Figure 3.14 is used for indicating the status of the system. There are different colors that can be selected to show the status of the system.

3.2.12 Thonny

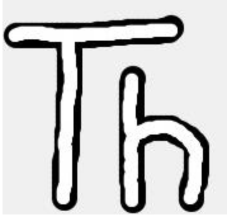

Figure 3.15: Thonny Software

Thonny (in Figure 3.15) is a software, which employs the Python programming language to write programs for the Raspberry Pi Pico W. In addition, it is also simple to use by anyone with no experience and is free software that can be downloaded from the internet.

3.2.13 Blynk

Figure 3.16: Blynk Software

This software is free software that is available on the internet and mobile devices. Users can observe the data and analyze the data through this software. It is free software, but users can unlock the advanced features by purchasing the plan provided by Blynk.

3.3 Block Diagram

A complete smart plant monitoring system's operating design is presented in the block diagram (Figure 3.17). Starting with powering the Raspberry Pi Pico W microcontroller, the next stages include connecting it to an LCD for displaying data and using relays as digital switches to operate the fan, water pump, and growth light based on signals from the microcontroller. Important sensors that provide important data on soil moisture levels, the surrounding temperature, and humidity, such as the DHT22 and soil moisture sensor, enable automated decision-making. With the help of these inputs, the system can automatically modify airflow and watering schedules to keep growing conditions ideal. Furthermore, an LDR sensor module is used to monitor light intensity, turning on the growth light when ambient light levels are sufficient. While Blynk acts as a channel for transferring and recording all data collected, LED indicators provide real-time feedback on the circuit's state, ensuring thorough environmental monitoring and control of the plant.

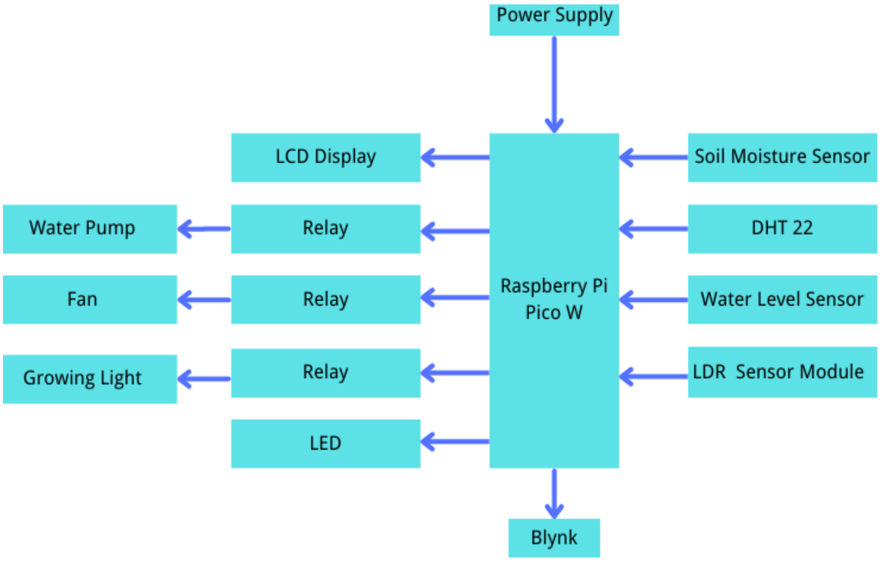

Figure 3.17: Block Diagram of Monitoring System

3.4 Setup for Raspberry Pi Pico W

The firmware needs to be downloaded from the official website in order to configure the Raspberry Pi Pico W. The Raspberry Pi Pico W and laptop should then be linked via the USB and micro-USB ports by pressing the bootsel button. The firmware file is then inserted into the Raspberry Pi Pico W file. Additionally, the Raspberry Pi Pico W's setup process is finished. The remaining setup steps, such as downloading the necessary libraries and connecting the program and hardware, are dependent on the coding software.

3.5 Smart plant monitoring system

The fan and DHT22 are used to perform a cooling system in this project. The humidity and temperature of the surrounding air will be measured by the DHT22. The Raspberry Pi Pico will get the reading result, which Blynk will record. After that, a relay controls the fan, and when the condition is met, both the fan and the LED turn on. For instance, the fan and LED automatically switch on when the surrounding temperature rises above 33°C; otherwise, they turn off. The water pump and moisture sensor are then combined to form the system for watering. The soil's moisture content will be determined by the moisture sensor. The Raspberry Pi Pico W will get the reading result, which Blynk will record. The water pump and LED switch on for one second once the criteria are met, and the water pump is then controlled by a relay. For instance, the water pump and LED will automatically switch on and transfer water to the soil if the soil's moisture content is less than 10%; if not, it will turn off. Next, a growth light and an LDR sensor module operate the lighting system. The microcontroller will get a digital signal from the LDR sensor module when it detects the amount of light in its surroundings. Light will provide a signal to the microcontroller when its intensity drops, and the microcontroller will then send a signal to the relay to switch on the growing light. The developing light will go out otherwise. Finally, the water level sensor will determine the container's water level.

The LED will light up to show that there is not enough water in the container when the water level is low. The user must then fill the container with water.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Circuit Diagram

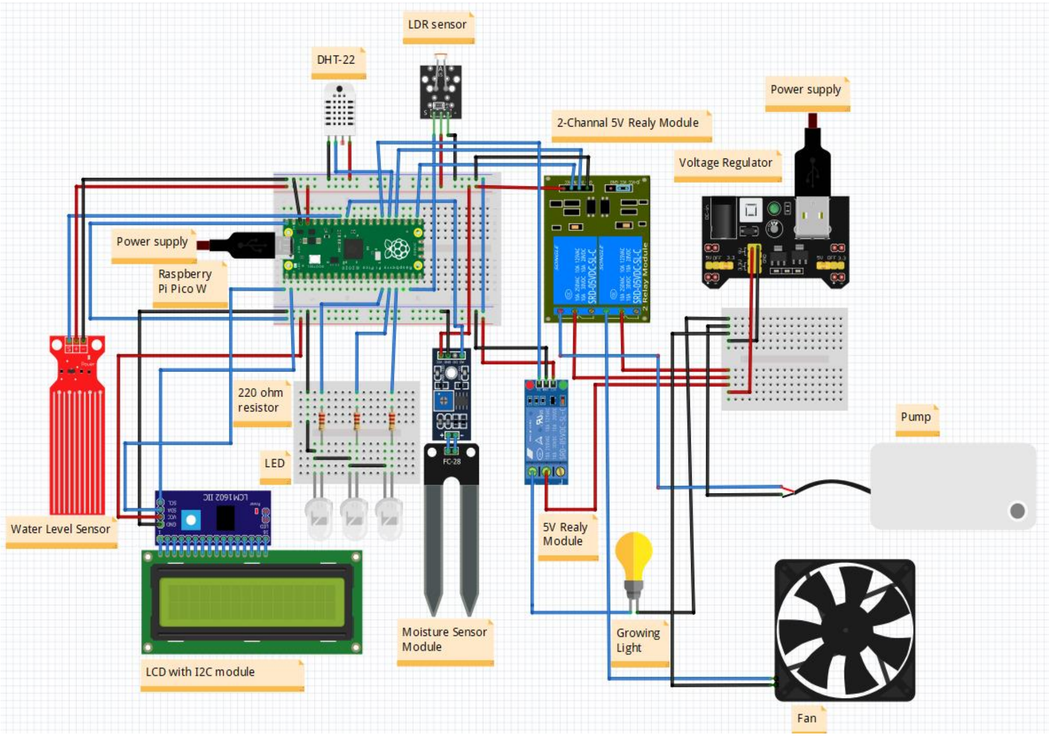

Figure 4.1: Circuit Framework of Monitoring System

Figure 4.1 shows the circuit construction of the project. All the components listed in Chapter 3 with the subsection 3.2 Hardware and Software Requirements had been used. The red color wire is represented the input voltage, the black color wire is represented the ground, and the blue color is represented the input and output signal of the sensor. Furthermore, the LCD screen and a single channel 5V relay are connected to 5V as input voltage. It is due to the power is not sufficient to make these two components work properly. Then, the rest of the sensor and LED are connected to 3.3V as input voltage. The voltage is sufficient to support all the rest of the sensor and relay. Next, the fan, water pump, and growing light are connected to an external voltage supply. If those components get the power from Raspberry Pi Pico W, the whole circuit will stop operating due to a lack of power to maintain the operation. Therefore, the voltage regulator is used to provide the voltage to the fan, water pump and growing light.

4.2 Physical Design of The Project

Figure 4.2: Front View of Monitoring System

Figure 4.3: Side View of Monitoring System

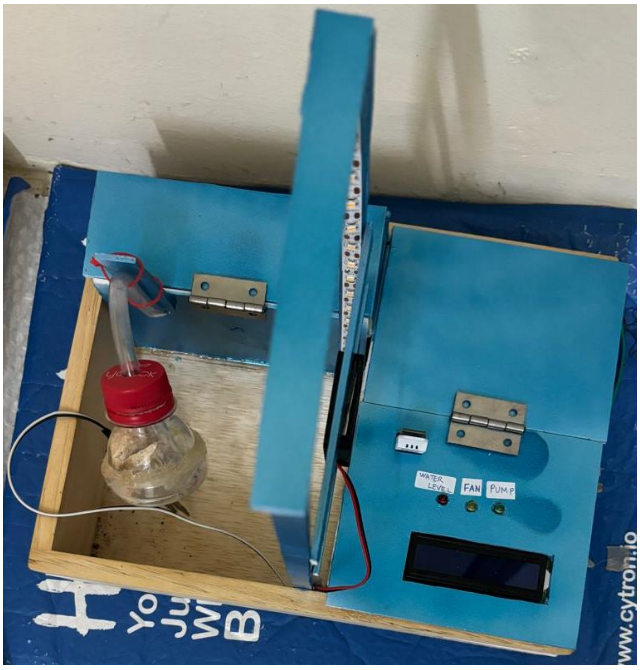

Figure 4.4: Top View of Monitoring System

Based on the diagrams obtained in Figures 4.1 – 4.4, the physical layout of the project had been completely designed. The LCD will display the reading, status of the fan, water pump, growing light, and the water level of the container. The LED light will turn on when the respective component is turned on or the water level of the container is low. Users also can open the cover to add water and plug in or replace the wire. The LDR sensor module is placed outside the box. It is to prevent the light provided by growing light affect the function of the LDR sensor.

4.3 Plant Growing Condition

4.3.1 Humidity

Figure 4.5: Humidity Observation in First Week

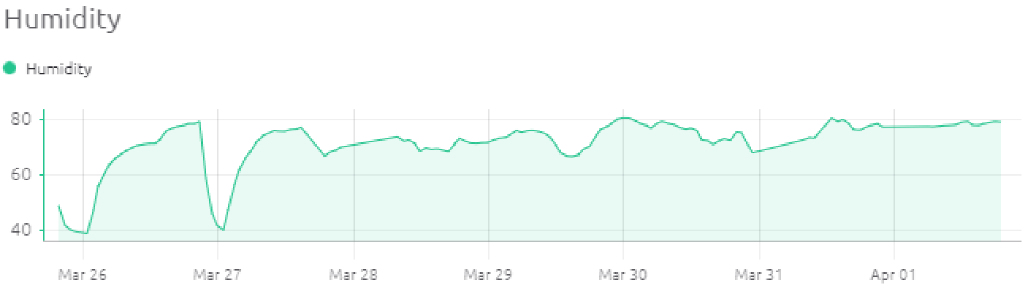

Figure 4.6: Humidity Observation in Second Week

As can be seen the diagrams in Figures 4.5 and 4.6, the results provide useful graphical representations of the patterns of humidity throughout the two weeks. The first week's observations showed a significant variation in humidity, with daytime averages ranging from 75% to 80% and nighttime averages of 40% to 46%. The primary cause of this difference is where the air conditioner operates. The injection of air conditioning efficiently lowers the monitored environment's total humidity levels in addition to lowering the surrounding temperature (Team, 2022). Therefore, the air conditioner's operation causes a rapid drop in humidity, showing the important effect of environmental controls on humidity. By comparison, the humidity level during the second week was more stable than the one before it. Humidity levels stayed relatively stable over this time, ranging from 65% to 80%. This stability can be explained by knowing that the air conditioner was not running at this period.

4.3.2 Temperature

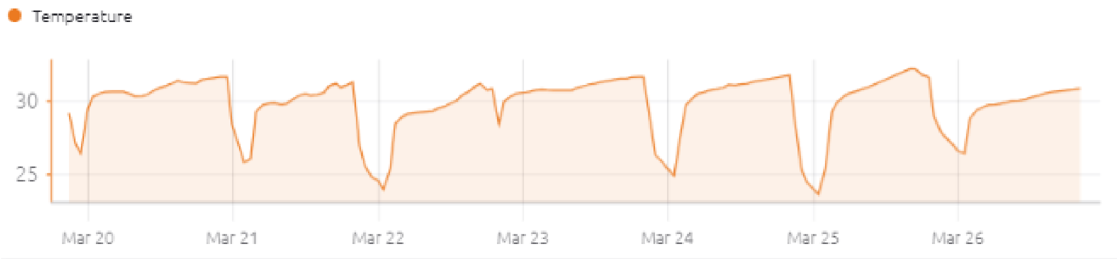

Figure 4.7: Temperature Observation in First Week

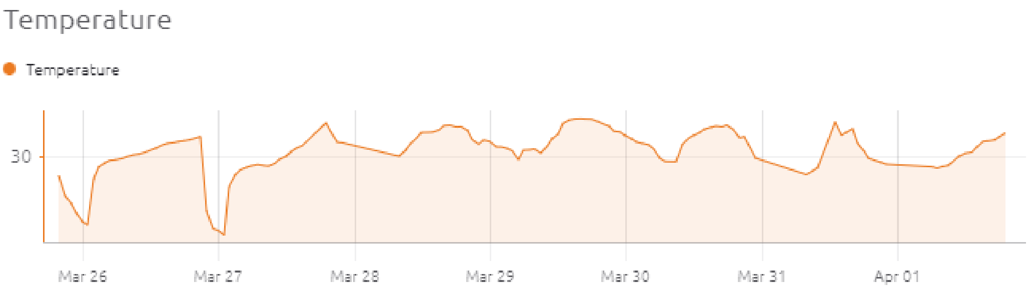

Figure 4.8: Temperature Observation in Second Week

For the first week shown in the diagram in Figure 4.7, the daily temperatures ranged from 27°C to 32°C, which produced a warm environment that was ideal for plant growth. But when nightfall arrived, there was a rapid drop in temperature, with nighttime measurements reaching between 23°C and 25°C. This can be attributed to the air conditioning system turning on. On the other hand, the diagram in Figure 4.8 shows the second week's temperature data was more consistent in structure, maintaining between 30°C and 32°C. The absence of air conditioning during this period can be directly related to the temperature's stability.

4.3.3 Moisture

Figure 4.9: Moisture Level in First Week

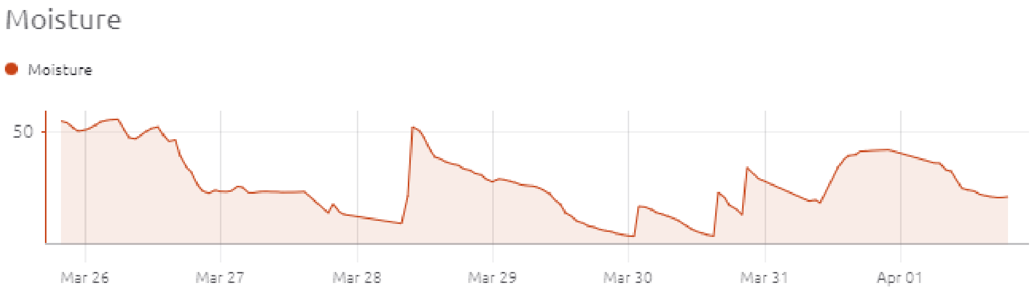

Figure 4.10: Moisture Level in Second Week

According to the diagram in Figure 4.9, the graph shows that the soil's initial moisture content was roughly 35%. The device maintained the soil moisture between 10% and 60% for the entire two-week duration. However, because the moisture sensor is small compared to the flowerpot, it might not be able to accurately measure the moisture level of the entire soil. The difference may cause the sensor to detect moisture levels below 10% even in situations where the pot's overall moisture content is higher. For the water pump, it will provide around 7.5ml of water per time. It is sufficient for the soil to keep moisture and will not overwatering. By observing the diagram in Figure 4.9, the water pump was turned on between March 23 and 24, even though the graph's moisture level didn't drop below 10%. The reason for the difference was that the moisture sensor detected a temporarily low moisture level, possibly because of movement or a small impact on the sensor. On the other hand, the moisture level in second week preforms a stable and consistence duration to adding water as seen in Figure 4.10. By observing the graph's trends, one observes that the level of moisture decreases for about four days, from 60% to 10%. The environment in which the plants are located, and their pace of decline could differ.

Plants that are placed outside usually lose moisture more quickly than plants that are indoors due to factors like sunlight, wind, and higher temperatures.

4.3.4 Water Level

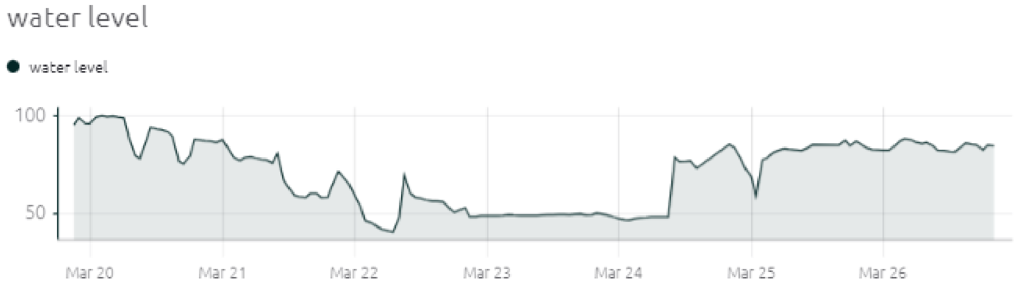

Figure 4.11: Water Level of Container in First Week

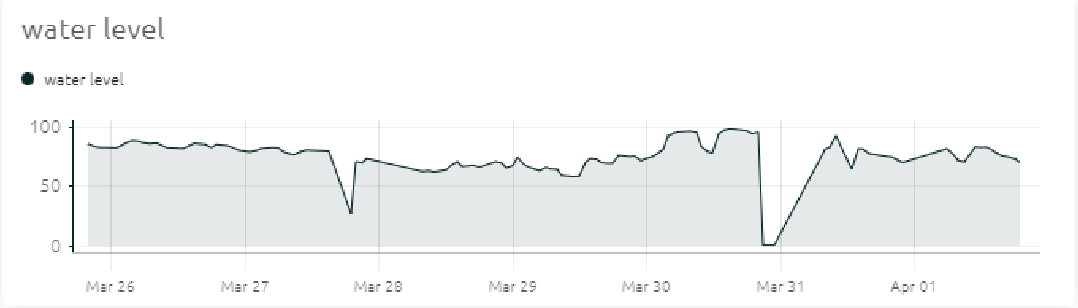

Figure 4.12: Water Level of Container in Second Week

The water level sensor is a key component in this part for controlling the water level inside the container. The system notifies the user to refill the container with water by turning on an LED indicator and displaying a notification on the LCD screen when the water level falls below 30%. The diagram in Figure 4.11 shows the first week of the water level decreased but the pump had not turned on and then kept consistent. This was caused by the water in the container being evaporated. After adding a cover, to reduce the area of the container exposed to the environment. The water level of the container had become more const. However, a failure in the water level sensor caused an issue in the system between March 30 and March 31, as seen in Figure 4.12. After the defective water level sensor was replaced, normal operation was resumed, and the water level was maintained at roughly 70% to 80%. Maintaining proper water levels is critical to plant health. Automatic watering systems rely on regular monitoring and maintenance to guarantee that plants get enough moisture. Inadequate water levels may affect water distribution and limit plant growth.

4.3.5 LDR sensor

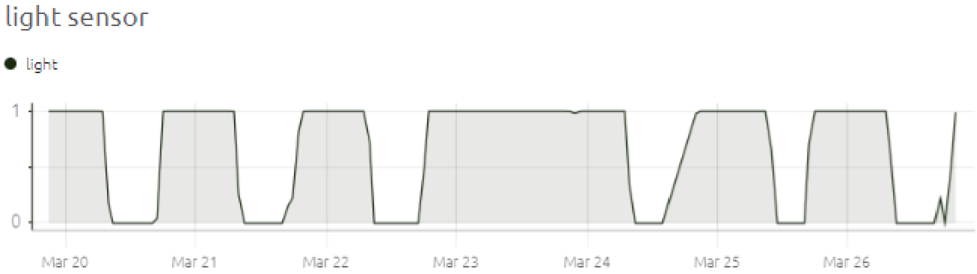

Figure 4.13: Light Detection Observation in First Week

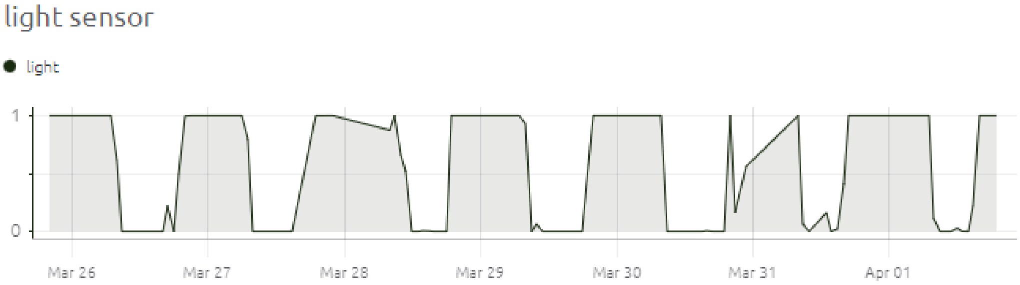

Figure 4.14: Light Detection Observation in Second Week

To determine the light intensity in the surroundings, the LDR sensor is required. The typical sequence of light presence throughout the day and absence during the night is shown in Figure 4.13. On March 23 and 24, however, there is one occurrence that is noted: light is absent all day long. The reason for this abnormality is that it is a rainy day, which causes the LDR sensor to continuously not register the light. As a result, the growing light is turned on to provide sufficient lighting for plant growth when natural light is not present. The sequence of light intensity during the second week is very similar to that of the first. However, as Figure 4.14 shows, a special waveform shows between March 30 and March 31. The reason for this difference is that the

LDR sensor detects a higher amount of light in the room. The growing light is turned on as a safety precaution to keep the ideal lighting conditions for plant growth, even though the light intensity does not reach to one but instead stays above 0.5. To save energy and avoid unnecessary light, the growth light stays off if the light intensity stays at or below 0.5.

4.4 Plant Growing With And Without Monitoring System

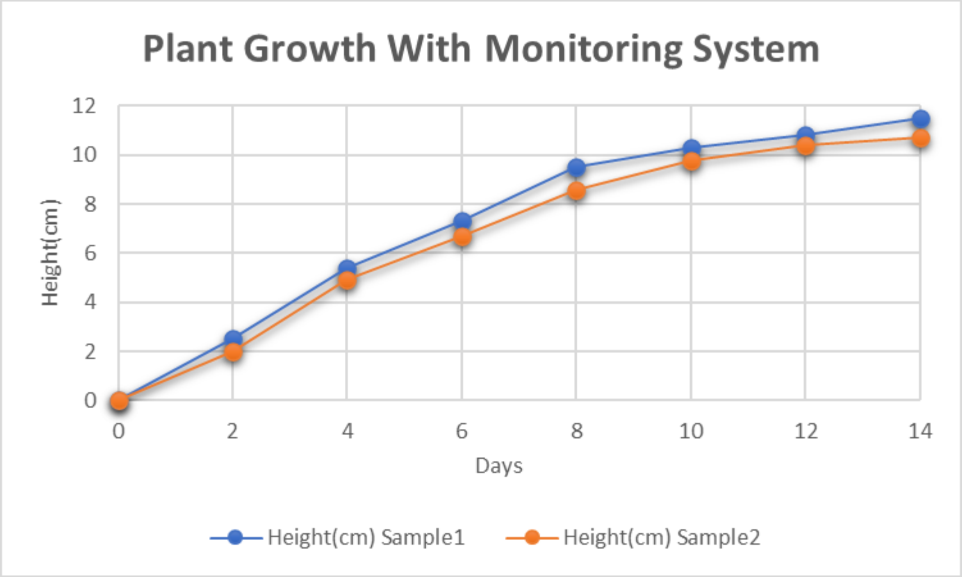

Figure 4.15: Plant Growing with Studied Monitoring System

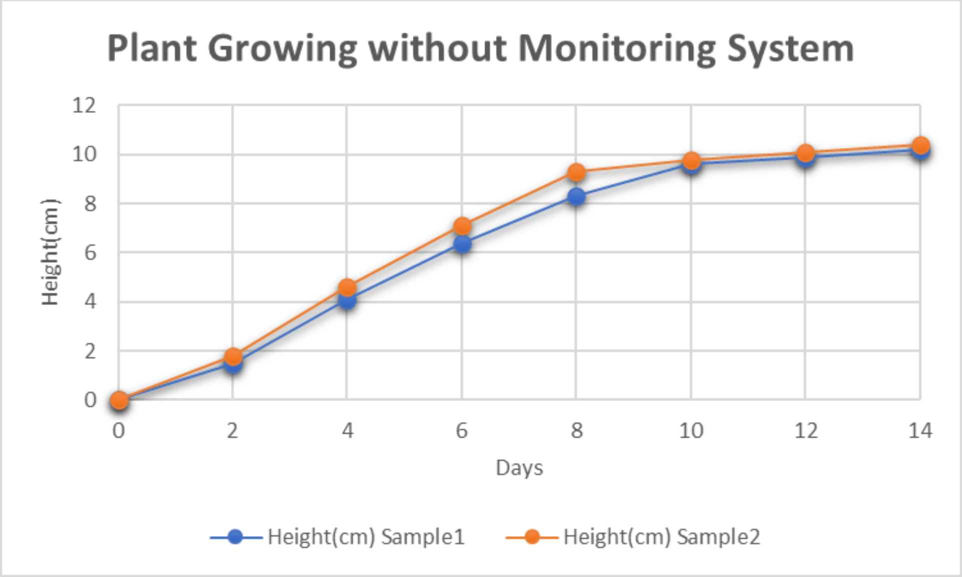

Figure 4.16: Plant Growing without Monitoring System

The selection of okra as the plant under observation in this experiment provides important information about the plant's particular environmental needs. Okra grows best in environments with carefully controlled soil moisture levels because of its hardiness and low water requirements (Westerfield, 2022). In addition, it is imperative to preserve an ideal temperature range for the growth of okra, with 25°C to 35°C being the perfect range for its production. Extreme temperature swings, however, can have a negative impact on okra yield and growth. For example, temperatures below 17°C might prevent seed germination, while temperatures above 42°C can cause blossom and leaf drop (Kumar, Ramjan, and Das, 2019). The height growth of okra plants, observed by the automated system during a two-week period, is shown in Figure 4.15. Over the first eight days, there is an initial period of rapid growth, which is due to the rapid growth of the plant's structure. Then, when the plants concentrate on growing larger leaves rather than taller ones, growth levels out. Notably, the plant's leaves have an excellent, healthy green color, which is an indication of optimal conditions for growth. On the other hand, the okra plants that are manually observed and shown in Figure 4.16 are irrigated with about 40ml of water every two days. Although at first, their growth rate can be similar to the plants under system monitoring, problems with seed shell keeping lead to later poor growth. One possible explanation for this delay in seed shell shedding is the compaction of the soil, which prevents plant growth. Additionally, as seen by the appearance of yellowing leaves, overwatering may contribute to an unbalance in soil moisture, which can result in a lack of nutrients and soil hardening after drying. When the conditions of the plants and the monitoring systems are compared, it is known that the plants under system monitoring have healthier development patterns. This shows that by efficiently controlling environmental factors like temperature and soil moisture, the automated monitoring system supports the best possible growth and general health for plants.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The completion of the project's goals can be determined by the successful cultivation of okra under healthy conditions for the duration of the monitoring period. Through the use of Blynk software, users can easily obtain real-time data analysis and monitoring of the okra plants, which improves their ability to make accurate judgments about necessary environmental modifications or actions. Users can easily track important variables like temperature, moisture level in the soil, and plant growth using laptops or smartphones, allowing for early control of the growing environment. After the project's objectives are completely achieved, users can optimize okra growing methods and increase yield potential by using the data obtained from the monitoring system. User can guarantee the health and energy of their okra plants by keeping perfect growing conditions and quickly solving any issues or problems found with the Blynk program. In the end, the monitoring system's successful deployment highlights how well it works to support efficient and productive plant growth while giving users a convenient platform for data analysis and decision-making.

5.2 Future recommendations and improvements

For further modifications, including a camera for leaf condition classification in the monitoring system is a major improvement. With the help of machine learning algorithms, this technology will enable the automatic examination of leaf health, giving users useful data about plant health that goes beyond simple environmental measurements. Users can more accurately evaluate problems like stress or disease by extending the system's capabilities to include visual assessment of leaf condition. This allows for immediate action and enhances overall plant care. Additionally, by making the system flexible to a larger range of plant species and cultivation conditions, this aspect increases the system's potential effect in agricultural environments.

REFERENCES

Ada, L., (2021). *DHT11, DHT22 and AM2302 Sensors* [Online]. Available at: https://www.mouser.com/datasheet/2/737/dht-932870.pdf [Accessed:1 September 2023].

Barhate, Y., Borse, R., Adkar, N., & Bagul, G., (2020). *Plant Watering and Monitoring System Using IoT and Cloud Computing*. International Journal of Scientific Development and Research (IJSDR), vol.5, no.4, pp.157-162. [Online] Available at: https://www.ijsdr.org/papers/IJSDR2004025.pdf [Accessed: 3 September 2023]

Biomaker, (2021). *Capacitive Soil Moisture Sensor* [Online]. Available at: https://www.biomaker.org/block-catalogue/2021/12/17/soil-moisture-sensor-aideepen-v12#:~:text=Most%20capacitors%20contain%20at%20least,separated%20by%20a%20dielectric%20medium.&text=A%20capacitive%20moisture%20sensor%20works,the%20changes%20in%20the%20dielectric [Accessed: 1 September 2023].

Deepan, R., Manojkumar, R., Jeevanandhan, B. & Sathishdhurai, V. S., (2022). *A Pico Microcontroller based IoT Device for Automation Agriculture and Irrigation Management.* International Journal of Engineering Research & Technology, vol 10, no.9, pp.21-24. [Online] Available at: https://www.ijert.org/a-pico-microcontroller-based-iot-device-for-automation-agriculture-and-irrigation-management [Accessed : 3 September 2023] eos.com. (2023). *Heat Stress In Plants: Symptoms, Prevention, And Recovery.* [online] Available at: https://eos.com/blog/heat-stress-in-plants/. [Accessed: 2 April 2024].

Food and Agriculture Organization, (n.d.). NSP – Physical affecting soil organisms. [Online]. Available at: https://www.fao.org/agriculture/crops/thematic-sitemap/theme/spi/soil-biodiversity/soil-organisms/physical-factors-affecting-soil-organisms/en/#:~:text=Organisms%20have%20a%20specific%20range,active%20throughout%20the%20entire%20range. [Accessed: 2 September 2023]

King, P., (2022). *What Is a Raspberry Pi Pico W and What Can You Use It For?* [Online]. Available at: https://www.makeuseof.com/raspberry-pi-pico-w-what-you-can-use-it-for/ [Accessed: 1 September 2023].

Lim, I. (2021). *DoSM: Covid-19 pandemic cut Malaysia's 2020 food output as farms downsized, fish catch fell from foreign labour shortage* [online]. Available at: https://www.malaymail.com/news/malaysia/2021/08/26/dosm-covid-19-pandemic-cut-malaysias-2020-food-output-as-farms-downsized-fi/2000670. [Accessed: 1 April 2024].

Mischianti, R.(2022). *Raspberry Pi Pico W:high-resolution pinout and specs* [Online]. Available at: https://mischianti.org/2022/09/16/raspberry-pi-pico-w-high-resolution-pinout-and-specs/ [Accessed: 1 September 2023].

Nehra, V., Sharma, M. and Sharma, V. (2023). *IoT Based Smart Plant Monitoring System.* [online] IEEE Xplore. Available at: https://doi.org/10.1109/Confluence56041.2023.10048792. [Accessed: 3 September 2023]

Neina, D., (2019). *The Role of Soil pH in Plant Nutrition and Soil Remediation.* [Online] Available at: https://www.hindawi.com/journals/aess/2019/5794869/ [Accessed: 2 September 2023].

Obias, R., (2023). *6 Best Raspberry Pi Models* [Online]. Available at: https://www.ign.com/articles/best-raspberry-pi-models [Accessed: 1 September 2023].

Olujinmi, T, (2022). *Raspberry Pi Pico vs. ESP32: Which Microcontroller Is Right for You?* [Online]. Available at: https://www.makeuseof.com/raspberry-pi-pico-vs-esp32-microcontroller/ [Accessed: 1 September 2023].

Pennisi, B., Thomas, P. and Stallknecht, E. (2022). *EFFECTS OF LOW TEMPERATURE on Plants.* [online] Available at: https://secure.caes.uga.edu/extension/publications/files/pdf/B%201467_2.PDF. [Accessed: 2 April 2024].

Pounder, L.,(2021). *Raspberry Pi Pico vs Arduino: Which Board Is Better?* [Online]. Available at: https://www.tomshardware.com/features/raspberry-pi-pico-vs-arduino [Accessed: 1 September 2023].

Rathore, A. (2020). *What Is Light Dependent Resistors (LDR)?* [online] Available at: https://www.electronicsforu.com/technology-trends/learn-electronics/ldr-light-dependent-resistors-basics. [Accessed: 2 April 2024].

Reckitt Benckiser (2021). Rate of photosynthesis: Limiting Factors. [online]. Available at: https://edu.rsc.org/download?ac=12620. [Accessed: 1 April 2024].

Technicals, N., (2021). *Soil Moisture Sensor working principle and applications* [Online]. Available at: https://www.nandantechnicals.com/2021/03/soil-moisture-sensor-working-principle.html [Accessed: 1 September 2023].

UMass Amherst (2015). *Growing-on at Cooler than Optimum Temperatures.* [online] Available at: https://ag.umass.edu/greenhouse-floriculture/fact-sheets/growing-on-at-cooler-than-optimum temperatures#:~:text=As%20the%20temperature%20increases%20above,than%20those%20from%20cooler%20climates. [Accessed: 2 April 2024].

University of Minnesota Extension (2020). *Lighting for Indoor Plants and Starting Seeds.* [online] Available at: https://extension.umn.edu/planting-and-growing-guides/lighting-indoor-plants#:~:text=All%20plants%20require%20light%20for. [Accessed: 2 April 2024].

Whitmore, C., (2022). *What is a Raspberry Pi and how does it work?* [Online]. Available at: https://nordvpn.com/blog/what-is-raspberry-pi/ [Accessed: 1 September 2023].

APPENDICES

Appendix A: Computer Programme Listing

A. Main python code for this project

```
import time
import network
from machine import Pin, I2C, ADC
from dht import DHT22
from pico_i2c_lcd import I2cLcd
from time import sleep
from blynklib import Blynk
# Initialize I2C communication
i2c = I2C(0, scl=Pin(1), sda=Pin(0), freq=400000)
lcd = I2cLcd(i2c, 0x27, 2, 16)
# Initialize the network connection
ssid = 'Chong_2.4GHz@unifi'
password = 'chuen4476'
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(ssid, password)
# Display "Connecting..." on the LCD
lcd.clear()
lcd.putstr('Connecting...')
time.sleep(1)
# Wait for network connection
wait = 100
```

```
while wait > 0:
    if wlan.status() < 0 or wlan.status() >= 3:
        break
    wait -= 1
    print('waiting for connection...')
    time.sleep(1)

if wlan.status() != 3:
    lcd.clear()
    lcd.putstr('Connection')
    lcd.move_to(0, 1)
    lcd.putstr('failed')
    raise RuntimeError('network connection failed')
    time.sleep(1)
else:
    lcd.clear()
    lcd.putstr('Connected')
    ip = wlan.ifconfig()[0]
    lcd.move_to(0, 1)
    lcd.putstr('IP: {}'.format(ip))
    print('connected')
    print('IP: ', ip)
    time.sleep(1)

    # Initialize Blynk instance with your authentication token
    BLYNK_AUTH = "6nTfi72BE2JlIEXn42bcu2j9WzZGZ8DA"
    blynk = Blynk(BLYNK_AUTH)

    # Display "Connecting to Blynk..." on the LCD
    lcd.clear()
    lcd.putstr('Connecting to')
    lcd.move_to(0, 1)
    lcd.putstr('Blynk...')
```

```
time.sleep(1)

def read_sensor():
    return sensor_pin.read_u16()

def map_to_percentage(reading):
    # Map the sensor reading to a percentage value
    percentage = ((reading - min_reading) / (max_reading - min_reading)) * 100
    # Ensure the percentage is within 0-100 range
    return max(0, min(100, percentage))

# Initialize the relay pins
relay = Pin(20, Pin.OUT)
fan_pin = Pin(16, Pin.OUT)
pump_pin = Pin(18, Pin.OUT)
pump_on = Pin (11, Pin.OUT)
fan_on = Pin(12, Pin.OUT)
water_low = Pin(13, Pin.OUT)

# Define the pin connected to the LDR
ldr = Pin(14, Pin.IN, Pin.PULL_DOWN)

# Initialize DHT22 sensor
dht_pin = Pin(19, Pin.IN, Pin.PULL_UP)
dht = DHT22(dht_pin)

# Initialize soil moisture sensor
soil = ADC(Pin(26))
min_moisture = 0
max_moisture = 65535

# Define the pin connected to the sensor
sensor_pin = ADC(Pin(27))  # ADC pin 27
```

```
    # Calibration values - adjust these according to your sensor and application
    min_reading = 0  # Minimum ADC reading corresponding to 0% water level
    max_reading = 42000  # Maximum ADC reading corresponding to 100% water level

    # Initialize the relay, fan, and pump pins, setting them to default OFF state
    relay_pin = Pin(20, Pin.OUT, value=1)  # Relay default OFF
    fan_pin = Pin(16, Pin.OUT, value=1)     # Fan default OFF
    pump_pin = Pin(18, Pin.OUT, value=1)   # Pump default OFF

    while True:
        try:
            dht.measure()
            temp = dht.temperature()
            hum = dht.humidity()
            moisture = (max_moisture - soil.read_u16()) * 100  / (max_moisture - min_moisture)
            moi = round(moisture, 1)
            sensor_reading = read_sensor()
            water = map_to_percentage(sensor_reading)

            print("moisture: " + "%.2f" % moisture + "% (adc: " + str(soil.read_u16()) + ")")
            print("T: {}°C   H: {:.0f}%  moisture: {:.0f}%".format(temp, hum, moi))
            print(f"ADC Value: {sensor_reading} | Water Level: {water:.1f}%")

            # Displaying to LCD
            lcd.clear()
            lcd.putstr('T:' + str(temp) + "C")
            lcd.move_to(9, 0)
            lcd.putstr('H:' + str(hum) + "%")
            lcd.move_to(0, 1)
            lcd.putstr('M:' + str(moi) + "%")
```

```
lcd.move_to(8, 1)
    lcd.putstr('WL:{:.0f}%'.format(water))
    time.sleep(2)

    # Control fan based on temperature
    if temp >= 33:
      fan_pin.value(0)  # Turn on the fan
      fan_on.value(1)
      lcd.clear()
      lcd.putstr('Fan is ON')
      time.sleep(1)
    else:
      fan_pin.value(1)  # Turn off the fan
      fan_on.value(0)
      lcd.clear()
      lcd.putstr('Fan is OFF')
      time.sleep(1)

    # Control pump based on soil moisture
    if moi <= 10:
      pump_pin.value(0)  # Turn on the pump
      pump_on.value(1)
      lcd.clear()
      lcd.putstr('Pump is ON')
      sleep(1)           # Wait for 1 second
      pump_pin.value(1)
      pump_on.value(0)
      lcd.clear()
      lcd.putstr('Pump is OFF')
      sleep(1)

    else:
      pump_pin.value(1)  # Turn off the pump
```

```
    pump_on.value(0)
    lcd.clear()
    lcd.putstr('Pump is OFF')
    sleep(1)

# Control relay based on LDR value
if ldr.value():
    relay.value(0)
    lcd.clear()
    lcd.putstr('Light is ON')
    sleep(1)
else:
    relay.value(1)
    lcd.clear()
    lcd.putstr('Light is OFF')
    sleep(1)

if water <= 30:
    water_low.value(1)
    lcd.clear()
    lcd.putstr('Water is Low,   ADD WATER')
    sleep(1)
else:
    water_low.value(0)

# Update Blynk pins
blynk.virtual_write(5, fan_pin.value())  #  virtual pin 5 for the fan
blynk.virtual_write(1, pump_pin.value())  #  virtual pin 1 for the pump
blynk.virtual_write(7, relay.value())  #  virtual pin 7 for the relay
blynk.virtual_write(2, temp)  #  virtual pin 2 for the temperature
blynk.virtual_write(3, hum)  #  virtual pin 3 for the humidity
blynk.virtual_write(4, moi)  #  virtual pin 4 for the moisture
blynk.virtual_write(6, ldr.value()) #  virtual pin 6 for the LDR
```

```
        blynk.virtual_write(8, water) #  virtual pin 8 for the water level

        # Run Blynk
        blynk.run()

    except Exception as e:
        print("An error occurred:", e)
        lcd.clear()
        lcd.putstr('Error occurred')
        lcd.move_to(0, 1)
        lcd.putstr('Reconnecting...')
        print("Attempting to reconnect to Blynk...")
        try:
            blynk.connect()
            print("Reconnected to Blynk successfully.")
            lcd.clear()
            lcd.putstr('Reconnected to')
            lcd.move_to(0, 1)
            lcd.putstr('Blynk successfully')
        except Exception as e:
            print("Reconnection to Blynk failed:", e)
            lcd.clear()
            lcd.putstr('Reconnection to')
            lcd.move_to(0, 1)
            lcd.putstr('Blynk failed')
            print("Please check your network connection and Blynk configuration.")
```