

DECOMPLEX

SMART PLANT POT INTEGRATING WITH THE INTERNET OF THINGS (IOT)



BY

TRI THAMWIWAT

WEERAPHAT WONGBUNNAK

WORAPON YINGYONG

YODKWAN THEERANITI

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF BACHELOR OF
ENGINEERING IN ROBOTICS AND AI**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY
LADKRABANG**

ACADEMIC YEAR 2022

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

ACKNOWLEDGEMENTS

The completion of this project could not have been possible without the help of the Wongbunnak family. Additionally, we would like to give exclusive thanks to Wongbunnak family for the workplace, supplies, equipment, and advice. The advice from Wongbunnak family significantly enhanced our project development progress and made the project approach our objective.

We express our deepest thanks to the Robotics and AI engineering that provided us with the budget to complete this project. Finally, we are indebted to Dr. Pitikhate Sooraksa, the project's chief consultant. With his permission and advice, it allow us to be more ambitious to present the magnificent project.



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

TABLE OF CONTENTS

	Page
ABSTRACT	II
ACKNOWLEDGEMENTS	III
TABLE OF CONTENTS	IV
LIST OF TABLES	IX
LIST OF FIGURES	X
LIST OF SYMBOLS/ABBREVIATIONS	XII
CHAPTER 1 INTRODUCTION	
1.1 Agriculture History	1
1.2 Agriculture Technology	1
1.3 Problems	2
1.4 Research	2
1.4.1 Internet of Things (IoT)	2
1.4.2 Micro Controller	2
1.5 Business Model	3
1.6 Report Outline	3
CHAPTER 2 HYPOTHESIS	
2.1 Green House	4
2.2 Plant Growth	5
2.3 Project Hypothesis	6
2.3.1 Hypothesis	6
2.3.2 Rationale	6
CHAPTER 3 METHODOLOGY	
3.1 Timeline	7
3.1.1 Gantt Chart	7
3.1.1.1 Planning	7
3.1.1.2 Research	7
3.1.1.3 Development	8
3.1.1.4 Paper	8
3.2 Plant Research	9
3.2.1 Environment for Plant	9
3.3 Technology Research	11
3.3.1 Micro Controller	11
3.3.1.1 Raspberry Pi	11

This material is prepared for educational use only, not allowed for commercial use.

TABLE OF CONTENT (Continue)

	Page
3.3.1.2 ESP32	12
3.3.2 Sensor	13
3.3.2.1 Humidity & Temperature Sensor (DHT11)	13
3.3.2.1 Ambient Light Sensor (TEMT6000)	13
3.3.2.2 Water Float Switch (EP-4510)	14
3.3.2.3 Soil Humidity Sensor (KS0049)	14
3.3.3 Actuator	15
3.3.3.1 Power Supply	15
3.3.3.2 Relay Module (4 Channel)	15
3.3.3.3 LED Strip	16
3.3.3.4 Water Pump	16
3.3.3.5 Fan	16
3.4 Designing	17
3.4.1 Sketch	17
3.4.2 Autodesk Inventor	17
3.4.3 Drawing File	18
3.5 Hardware Development	19
3.5.1 Material Selection	19
3.5.2 Plant Box	19
3.5.3 Circuit Box	20
3.5.4 Water Storage	21
3.5.5 Base	21
3.6 System Design	22
3.6.1 Raspberry Pi	23
3.6.2 Power Supply	24
3.6.3 ESP32 Input/Output	25
3.6.4 Terminal	26
3.6.5 Sensors with Breadboard	27
3.7 Software Development	28
3.7.1 ESP32	28
3.7.1.1 Code	28
3.7.1.1.1 Introduction	28

This material is for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

TABLE OF CONTENT (Continue)

	Page
3.7.1.1.2 Purpose	28
3.7.1.1.3 Structure	28
3.7.1.1.4 Input and Output	28
3.7.1.1.5 Code Snippet	29
3.7.1.1.6 Explanation	37
3.7.1.2 Flow Chart	38
3.7.2 Firebase	39
3.7.2.1 Authentication	39
3.7.2.2 Real-time Database	39
3.7.3 Raspberry Pi	40
3.7.3.1 Sending and Receiving Data	40
3.7.3.2 Showing Data	41
3.7.4 Flutter	42
3.7.4.1 Introduction	42
3.7.4.2 Overall Design	42
3.7.4.2.1 Registration	43
3.7.4.2.2 Login	46
3.7.4.2.3 Dashboard	48
3.7.4.2.4 Control	50
3.7.4.2.5 Side navigation bar	52
3.7.4.3 Navigation	53
3.7.4.3.1 Login Screen	53
3.7.4.3.2 Registration Screen	53
3.7.4.3.3 Registration Complete Screen	53
3.7.4.3.4 Dashboard Screen	53
3.7.4.3.5 Control Screen	53
3.7.4.3.6 Side Navigation Bar	53
3.7.4.3.7 Navigation Chart	54
3.7.4.4 Input and Output Elements	55
3.7.4.5 Dependencies	55
3.7.4.5.1 Rive Icons	55
3.7.4.5.2 Flutter Launcher Icons	55

This material is for personal use only, not allowed for commercial use.

TABLE OF CONTENT (Continue)

	Page
3.7.4.5.3 Form Field Validator	55
3.7.4.5.4 Page Transition	55
3.7.4.5.5 Firebase Core	55
3.7.4.5.6 Firebase Authentication	55
3.7.4.5.7 Firebase Database	55
3.7.4.5.8 Firebase Messaging	55
3.7.4.6 Code Explanation	56
3.7.4.6.1 animated_bar.dart	56
3.7.4.6.2 side_menu_tile.dart	57
3.7.4.6.3 side_menu.dart	59
3.7.4.6.4 menu_button.dart	62
3.7.4.6.5 rive_asset.dart	63
3.7.4.6.6 rive_utils.dart	64
3.7.4.6.7 userprofile.dart	64
3.7.4.6.8 user_controller.dart	65
3.7.4.6.9 user_dashboard.dart	75
3.7.4.6.10 entry_point.dart	85
3.7.4.6.11 main.dart	91
3.7.4.6.12 pubspec.yaml	93
CHAPTER 4 EVALUATION AND RESULT	
4.1 Evaluation Criteria	96
4.2 Testing Method	96
4.3 Result	97
4.3.1 Height Table Data	97
4.3.2 Temperature, Air and Soil Humidity Table Data	98
4.3.3 Height Graph	99
4.3.4 Temperature, Air and Soil Humidity Graph	100
4.4 Data Analyze	102
4.4.1 Analyze – Plant Height	102
4.4.2 Analyze – Temperature, Air and Soil Humidity	102
CHAPTER 5 CONCLUSION	
5.1 Learning Point	103

This document is intended for educational use only, not allowed for commercial use.

TABLE OF CONTENT (Continue)

	Page
5.1.1 IoT Protocol	103
5.1.2 Cloud Control System	103
5.1.3 Control Room	104
5.2 Future Development	105
5.2.1 Improved Sensor Technology	105
5.2.2 Advanced Actuator Systems	105
5.2.3 Extended Experiment Duration	105
5.2.4 Iterative System Refinement	106
5.2.5 Plant Diversity and Specialty	106
References	107
APPENDICES	
APPENDIX A: NODERED UI PAGE GRAPH	110
APPENDIX B: SENSOR POSITION ON BREADBOARD	111
APPENDIX C: ORIGINAL PROJECT SKETCH	112
APPENDIX D: PROJECT 3D MODEL	113
BIOGRAPHY	114

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

LIST OF TABLES

Tables	Page
Table 4.1.1 Reason for not choosing the criteria	96
Table 4.2.1 Explanation of each control environment	96
Table 4.3.1.1 Plant Height Table	97
Table 4.3.2.1 Auto environment data of Temperature, Air and Soil Humidity	98
Table 4.3.2.2 Control environment data of Temperature, Air and Soil Humidity	98
Table 4.4.1.1 Analyzing Plant Height	102
Table 4.4.2.1 Analyzing Temperature, Air and Soil Humidity from auto and control environment	102



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content **IX** and cite the document when use.

LIST OF FIGURES

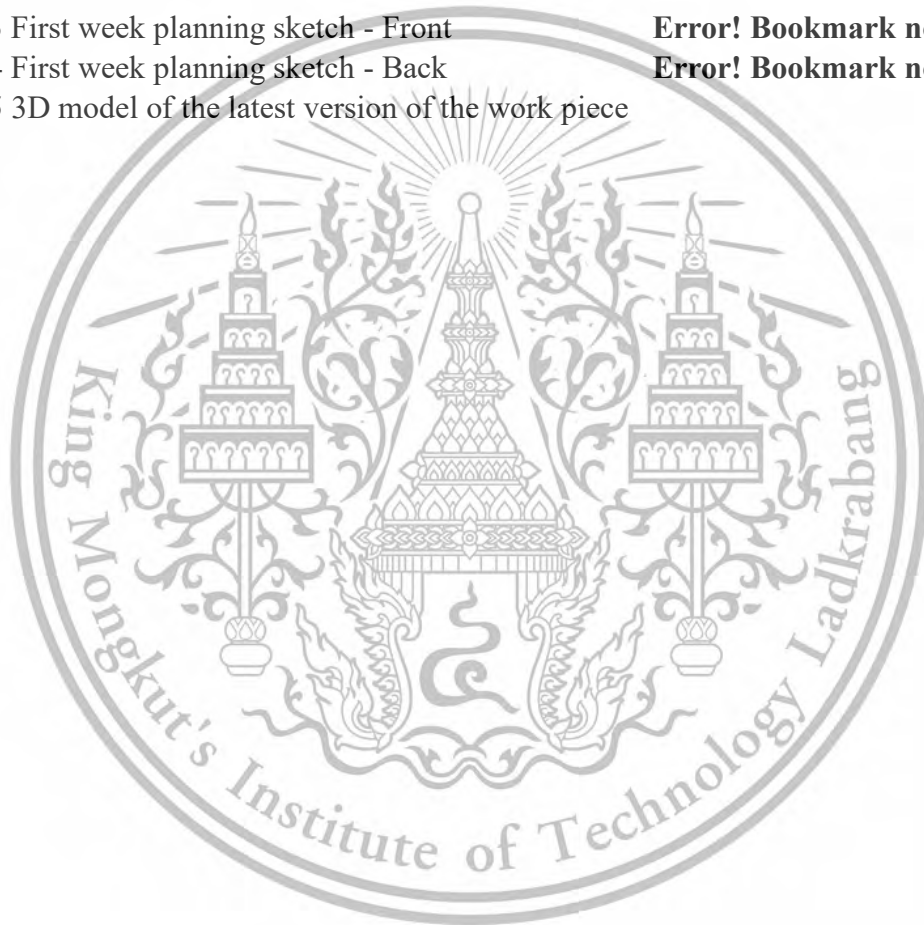
Figures	Page
Figure 1. 1 Business Model Canvas	3
Figure 2. 1 A small green house	4
Figure 2. 2 Measuring test plant	5
Figure 3. 1 Gantt chart for the project	7
Figure 3. 2 Asparagus use in testing	9
Figure 3. 3 Raspberry Pi 4 (Model B)	11
Figure 3. 4 ESP32-S	12
Figure 3. 5 DHT11 Humidity and Temperature Sensor	13
Figure 3. 6 TEMT6000 Light Sensor	13
Figure 3. 7 EP-4510 Float Switch	14
Figure 3. 8 Ks0049 Soil Humidity Sensor	14
Figure 3. 9 Power supply box	15
Figure 3. 10 4 Channel Relay Module	15
Figure 3. 11 LED light strip	16
Figure 3. 12 9x9cm fan	16
Figure 3. 13 Autodesk Inventor app Logo	17
Figure 3. 14 Engineering drawing of box side with fan hole	18
Figure 3. 15 Engineering drawing of overall project	18
Figure 3. 16 The plant box with a plant in it	19
Figure 3. 17 Arranging position of hardware before circuit box assembly	20
Figure 3. 18 Inside the water tank with water sensor and pump	21
Figure 3. 19 Wiring in the circuit box	22
Figure 3. 20 Raspberry Pi Wiring Diagram	23
Figure 3. 21 Power Supply Wiring Diagram	24
Figure 3. 22 ESP32 and Relay Wiring Diagram	25
Figure 3. 23 Terminal Block Wiring Diagram	26
Figure 3. 24 Sensor Wiring Diagram	27
Figure 3. 25 ESP32 code flowchart	38
Figure 3. 26 Data structure in Firebase	39
Figure 3. 27 Code snippet from nodeRed	40
Figure 3. 28 Code for nodeRed UI	41
Figure 3. 29 UI page from nodeRed	41
Figure 3. 30 Decomplex app registration page	44
Figure 3. 31 Registration success confirmation screen	45
Figure 3. 32 Log-in screen demonstration	47
Figure 3. 33 Dashboard in the application showing data reading from the workpiece	49
Figure 3. 34 Control console	51
Figure 3. 35 Side Navigation Bar	52
Figure 3. 36 Chart showing possible navigation of each screen	54

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content ~~X~~ And cite the document when use.

LIST OF FIGURES (Continue)

Figures	Page
Figure 4. 1 Plant Height Graph	Error! Bookmark not defined.
Figure 4. 2 Compare Temperature of auto and control	Error! Bookmark not defined.
Figure 4. 3 Compare Air Humidity of auto and control	Error! Bookmark not defined.
Figure 4. 4 Compare Soil Humidity of auto and control	Error! Bookmark not defined.
Figure 5. 1 nodeRed UI page graph section	Error! Bookmark not defined.
Figure 5. 2 Humidity, Temperature, and Light sensor install in the control box	Error! Bookmark not defined.
Figure 5. 3 First week planning sketch - Front	Error! Bookmark not defined.
Figure 5. 4 First week planning sketch - Back	Error! Bookmark not defined.
Figure 5. 5 3D model of the latest version of the work piece	113



LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
IoT	Internet of Things
MCU	Micro Controller Unit
Temp	Temperature
MQTT	MQ Telemetry Transport
CAD	Computer Aided Design Software



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

CHAPTER 1

INTRODUCTION

Food is important to every living thing in the world because all living things need food to live. Each living things have their own methods to find food. Agriculture is one of the methods that human use for finding food supply. Bill and Melinda gate foundation is quoted as stating, "Investments in agriculture are the best weapons against hunger and poverty, and they have made life better for billions of people." According to the quote it shows how important agriculture by stating that better agriculture means better life. This paper represents the SMART PLANT POT INTEGRATING WITH THE INTERNET OF THING (DECOMPLEX) which is an agriculture monitoring system. Develop by Robotics and AI student from King Mongkut's Institute of Technology Ladkrabang (KMITL).

1.1 Agriculture History

Agriculture, food, and goods cultivated by farming are the main method to produce the world's food supply. In long view of human history, human have nomadic hunter-gatherer lifestyles in their first period and then they continue to develop the tools to help them survive through the brutal period. Human have developed many tools that help them to survive and some of them have change their lifestyle forever such as agriculture. Agriculture first found about 12,000 years ago and it change human lifestyle from nomadic hunter-gatherer lifestyles to permanent settlement and farming (National Geographic, 2023). With the permanent settlement and farming lifestyle it means that the food supply can be farmed to meet the demand and cause the human population to rocketed and it also created a new job like farmer as well. Therefore, agriculture have stay with human for a long time and it can change the way that human live by provide them higher opportunities to survive.

1.2 Agriculture Technology

Since the 2000, human population tend to grow up and demand more food to survive. With the higher food demand. Human invented many methods to improve their food supply. To solve the insufficient of food supply human have use modern technologies to make higher crop productivity, reduce the budget, reduce impact on natural ecosystems, reduce chemical use and increase worker safety. To achieve those benefit human, have implement the modern technologies such as GPS technology, robotics system or even precision agriculture devices in their agriculture system which called modern agriculture (USDA,2023). From using those technologies, there are many products that use for modern agriculture such as greenhouse system and automatic agriculture system. To summaries, human pollution and food supply are direct variation and to maintain the food supply from direct variation state human have fused both agriculture and technologies together and called modern agriculture.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

1.3 Problems

Growing of high-quality economic plant depend on many factors. Factors such as water, temperature, light, moisture in the soil and humidity. Plant in unstable environment can massively impact the quality and in the worst case it might cause the plant to die. Other than stable environment. Plant needs a daily care which require owner to water it every day and sometime the owner forgot to watering the plant and that give a negative impact to the plant. To solve those problems a system to look after the plants was create. A system which monitors the environment and automate the process to keep the plant in optimal growing condition. By integrating IoT to the product to give user more flexibility in monitoring and manage the control of the plant environment.

1.4 Research

1.4.1 Internet of Things (IoT)

Internet of Things or IoT is an object or device that have the capability to connect and exchange data with other device and system in the network. Usually refer to as Smart device. They are capable of collecting and processing data. Then communicate with other devices to be individually control.

IoT are synonymous with the smart home concept. Multiple devices each with their own function. Connected to a mobile application over internet. Allow user to monitor and control each service from a centralize location. The used of IoT is very widespread. Seeing application in Manufacturing and Agriculture sector as well.

1.4.2 Micro Controller

Micro Controller or MCU is a small computing device on a single integrated circuit chip. Design to be use in embedded application and automatic device. The small size of MCU has led to them being integrated into many applications ranging from handheld power tools and toys to office machine, remote control, and medical device.

1.5 Business Model

Key partners <ul style="list-style-type: none"> Hardware provider Agriculture store Furniture retail company 	Key Activity <ul style="list-style-type: none"> Development Maintenance Marketing Customer support 	Value proposition <ul style="list-style-type: none"> Save client time Can monitor plant anywhere Make plant not a routine work Convenient 	Customer relationship <ul style="list-style-type: none"> Customer support After service support 	Customer segment <ul style="list-style-type: none"> Agriculturist Plant lover Gardener
	Key Resources <ul style="list-style-type: none"> Capital Staff Agriculturist partnership 		Channels <ul style="list-style-type: none"> Mobile application Partnership Social media 	
Cost <ul style="list-style-type: none"> Box fame Hardware supply Software developer Technician Marketing 		Revenue <ul style="list-style-type: none"> Rental fees Sell After service fees 		

Figure 1.1 Business Model Canvas

The target customer for this project will be agriculturist, plant lover and gardener because this produce can help the customer by look after the plant automatically and make gardening easier. This project can be developed from education project to product for sell by following the business model.

1.6 Report Outline

The rest of this report is organized as follows:

Chapter 2 goes over the theory and hypothesis in the project. The theory discusses cover the concept of greenhouse and it effect and use for growing plant. And the way plant growth can be evaluated. The second half of the chapter explain the theory of the project and rationale behind them.

Chapter 3 describe the design and implementation of both hardware and software part of the project. Hardware side include wiring diagram and assembly of the work piece while Software side will discuss the programming of Esp32 board, Raspberry Pi, Node RED dashboard, Integration of database with Firebase and coding of the mobile application with Flutter

Chapter 4 Explain the testing process. How the data is collected and evaluated. Then create a summary and analyze the data collected to compare to the project hypothesis.

Chapter 5 closing the report. Draw a conclusion from the project and talk about what we learn while working on the project and find what could be improve in the future.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

CHAPTER 2 HYPOTHESIS

2.1 Green House



Figure 2.1 A small green house

A greenhouse is a building made to control variables like temperature, humidity, light, and ventilation to provide the ideal environment for plant growth. Regardless of the weather outside, it offers a safe environment for plants to develop and thrive.

The most popular materials used to construct greenhouses are glass or transparent plastic, which permit light to enter and retain heat within the structure. In comparison to the outside world, the greenhouse's climate is warmer and more stable thanks to the trapped heat. The advantages of this regulated environment for plant growth include:

1. **Temperature control:** Greenhouses make it possible to regulate the temperature, which enables plants to grow in a predictable and ideal range. By doing so, the growth season is extended and it is possible to grow plants that would not thrive in colder areas.
2. **Protection from severe weather:** Greenhouses protect plants from damaging weather elements like frost, high heat, a lot of rain, wind, or hail, ensuring a more stable climate and reducing damage.
3. **Better humidity control:** Plants that require precise moisture conditions for optimal growth will benefit most from greenhouses' capacity to regulate humidity levels.
4. **Pest and disease control:** By enclosing plants, greenhouses build a barrier against pests and diseases, lowering the danger of infestation and lowering the need for chemical treatments.
5. **Increased light exposure:** Plants are exposed to more light for photosynthesis thanks to the transparent materials used in greenhouse construction, which allow maximum sunlight penetration. By reflecting sunlight to locations that receive less direct light, this can still be improved further.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

2.2 Plant Growth

To evaluate plant growth, several parameters can be measured and assessed.

Height: Calculate the height of the plant by measuring it vertically from the ground up to the topmost leaf or stem. A ruler or tape measure can be used to evaluate.

Biomass: Calculate the plant's biomass, or overall weight. By gathering the plant, removing extra moisture, and weighing the plant material on a scale, this metric can be determined.

Leaf area: Determine the size of the leaves, which may be a sign of their ability to absorb light. Using specialized tools like a leaf area meter or image analysis software.

Root development: Carefully take the plant out of the growing medium and look at the roots to assess the root system. Utilizing methods like washing the roots and utilizing a root scanner.

Flowering and fruiting: Monitoring the production of flowers and fruits is vital because they serve as key markers of reproductive development. Assess the quantity of flowers or fruits that are produced as well as their quality (size, color, weight, etc.).



Figure 2.2 Measuring test plant

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

2.3 Project Hypothesis

2.3.1 Hypothesis

An automatic control system that can regulate water, temperature, humidity, and light for plant growth will be as effective as or more effective than traditional control methods while requiring less labor and manual intervention.

2.3.2 Rationale

Environmental Parameters Can Be Regulated Precisely: The automatic control system enables the accurate adjustment of environmental factors like water, temperature, humidity, and light intensity. Comparing this level of control to conventional control methods that rely on manual changes, it may result in increased growth rates and higher yields for the plants.

Consistency: By reducing swings and changes that could adversely affect plant development, the automatic control system creates a consistent environment for plant growth. The uneven growth circumstances caused by traditional control systems can be caused by human error or variances in manual adjustments.

Time Efficient: The control system's automation minimizes the need for manual intervention and supervision. When compared to conventional control methods, this may result in significant labor time savings, freeing up workers to focus on other crucial plant care and cultivation-related tasks.

Data Gathering and Analysis: The automatic control system is capable of gathering real-time data on plant responses and environmental variables. This information can be analyzed and adapted in order to improve the growth of plants. The ability to gather and analyze data with a comparable level of precision as well as effectiveness can sometimes be lacking in traditional control approaches.

CHAPTER 3 METHODOLOGY

3.1 Timeline

3.1.1 Gantt Chart

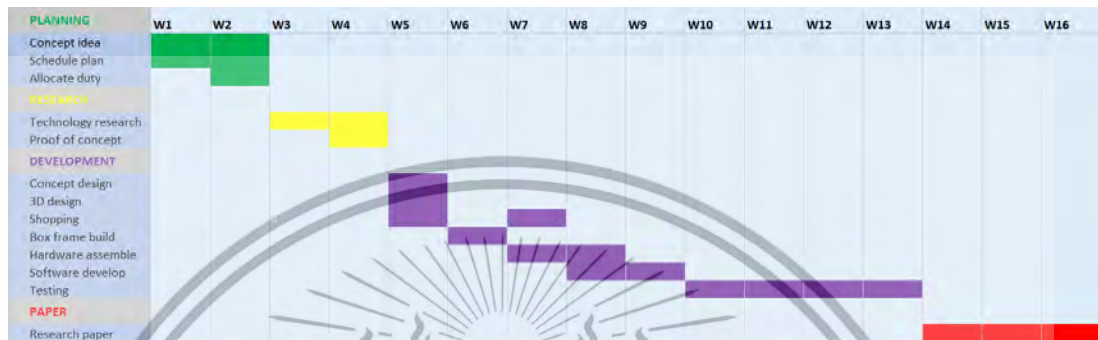


Figure 3. 1 Gantt chart for the project

The figure above is the Gantt chart the team use for planning while working on the project. It can be divided into four main phases.

3.1.1.1 Planning

First step of the project is to plan for the future. The team discuss concept idea. The scope and goal of the project are talk about in the first two weeks. Once the main idea started to form. Rough schedule was draft to get an idea of what have to be done to complete the project. Then each member was assigned main task and duty that they will be responsible for the project.

3.1.1.2 Research

For week three to four. The team then begin the process to research the theory and technology related to the project. Proof of concept is then tested before the team move on to begin working on the real work piece.

3.1.1.3 Development

This phase takes up the longest time in the project cycle as it involves designing, building, and testing of the main project itself. The first week is spend finalizing the design of the project. Before moving on to find the supply and material to build it. Extra time for shopping is allocate down the line for extra material and response to potential problem that might show up while assembling the workpiece framework.

With framework establish. Hardware and software are developed alongside one another as both require one another for testing to be done.

Once the work has been done. It is then sent off to be tested in real application.

3.1.1.4 Paper

The last three week of the Gantt chart plan is spent on writing paper report. With data from testing collected. All member help to create the report.



3.2 Plant Research



Figure 3. 1 Asparagus use in testing

Asparagus fern (*Asparagus setaceus*), is a native South African plant but well known in Thailand as a decorative plant. It is a multipurpose plant that can be used on various occasions for a decorative purpose, especially in ceremonies and events. Its charming appeal makes it one of the best choices for indoor decoration, as it can survive without requiring direct sunlight and can grow continuously to 3 meters tall.

3.2.1 Environment for Plant

To grow Asparagus fern successfully, it is very important to have a proper environment for the fern to grow, by providing a suitable environment and meeting the specific growth conditions. Some key factors to be considered are listed below.

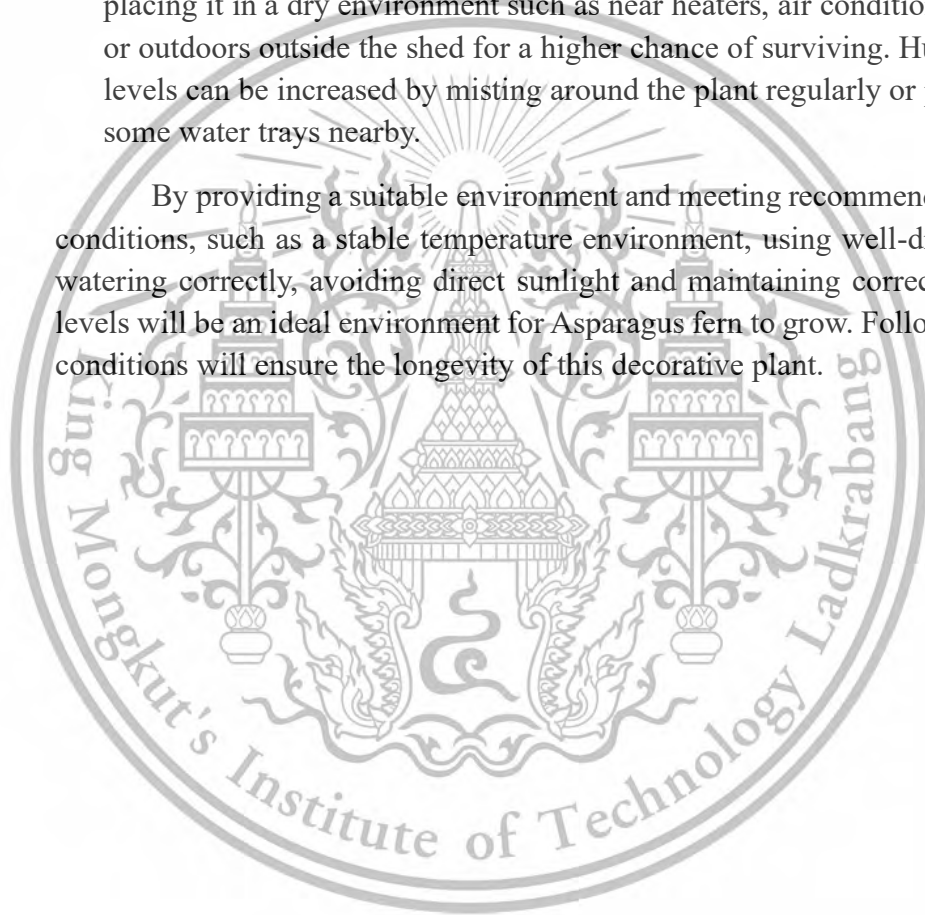
1. **Suitable Environment:** Asparagus fern can positively grow in indoor environments with stable temperature and humidity levels. With stable temperature and humidity levels as the best growth condition, it is best to grow the Asparagus fern indoors rather than outdoors.
2. **Temperature:** The best growth conditions require a temperature of 18 to 24 degrees Celsius. The extreme temperature fluctuations will have negative effects on Asparagus fern growth rate.
3. **Soil:** Asparagus fern prefer medium soil to grow because medium soil is well-drained and not immensely soggy, which prevents the root from rotting.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content,⁹ and cite the document when use.

4. **Watering:** Asparagus fern require frequent watering just to keep the soil moist, but you do not want large amounts of water at the same time. Overwatering can lead to root rot, which is negative for Asparagus fern. Watering when the top layer of the soil is dry only.
5. **Lighting:** Asparagus fern does not require direct sunlight, but as a plant it still needs moderate light to survive. Placing it in a location where it can receive some light will be sufficient for it to survive.
6. **Humidity:** Asparagus fern is at medium to high humidity levels. Avoid placing it in a dry environment such as near heaters, air conditioner vents or outdoors outside the shed for a higher chance of surviving. Humidity levels can be increased by misting around the plant regularly or placing some water trays nearby.

By providing a suitable environment and meeting recommended growth conditions, such as a stable temperature environment, using well-drained soil, watering correctly, avoiding direct sunlight and maintaining correct humidity levels will be an ideal environment for Asparagus fern to grow. Following those conditions will ensure the longevity of this decorative plant.



3.3 Technology Research

With parameters and condition to control to ensure plant growth. Including humidity, light, and water. Research was conducted to find equipment needed to read and collect data, communicate the data to other part of the project in order to automate the process of controlling the growth condition of the plant.

3.3.1 Micro Controller

3.3.1.1 Raspberry Pi



Figure 3.2 Raspberry Pi 4 (Model B)

Raspberry Pi is a small single-board computers. Developed by the Raspberry Pi Foundation in association with Broadcom. Originally develop as a teaching tool for computer science. Its versatility and performance see it adopt into many practical applications. The version we use in the project is Raspberry Pi 4 (Model B) which feature the ability to connect the board to external monitor. Which greatly help simplify the development process. The main purpose for the Raspberry Pi board in this project is to communicate with the database on the internet for further used. The detail of the software will be explained in its own section.

3.3.1.2 ESP32



Figure 3. 3 ESP32-S

ESP32-S is a small microcontroller developed by Espressif Systems. Acting as a central hub for reading data collected by set of sensors in the work piece. Its small size combines with low power usage and excellence performance make it perfect for its role.



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.3.2 Sensor

3.3.2.1 Humidity & Temperature Sensor (DHT11)

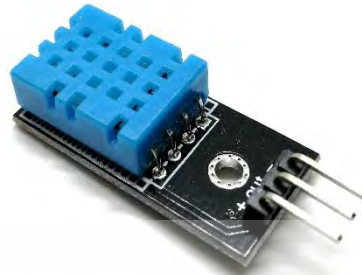


Figure 3. 4 DHT11 Humidity and Temperature Sensor

DHT11 is a sensor use to measure air humidity and temperature. According to the datasheet. The sensor has measurement range of 20-90%RH for humidity and 0-50 °C for temperature. Well within range of expected environment. It is also highly accurate with minimal deviation of 5% for humidity and 2°C for temperature.

3.3.2.1 Ambient Light Sensor (TEMT6000)



Figure 3. 5 TEMT6000 Light Sensor

A sensor used to monitor one of the main factors for plant growth and photosynthesis. The sensor works by detecting the incoming light. The higher the intensity the higher the output reading from the sensor will be. The sensor can only detect light within the human visible spectrum (390-700 nm) which is enough for the project as plant also use light in this spectrum.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.3.2.2 Water Float Switch (EP-4510)



Figure 3. 6 EP-4510 Float Switch

Use to detect water level in the water storage tank. The cylinder along the rod of the sensor will float while in water which close the circuit of the sensor. Sending output that indicated the condition to the micro controller. As the sensor have to be in water all the time. One of the requirements to select from is durability and EP-4510 achieve that goal while still being affordable and easy to install.

3.3.2.3 Soil Humidity Sensor (KS0049)



Figure 3. 7 Ks0049 Soil Humidity Sensor

Install in the plant pot. The sensor monitors the humidity in the soil to check when to water the plant. When soil is dry the analog output will decrease. Otherwise, it will increase. It can detect this by using the soil as conducting medium. Sending current from one of the leg of the sensor into another leg. If the soil is humid, that mean there are less resistance which make the analog output high.

3.3.3 Actuator

3.3.3.1 Power Supply



Figure 3. 8 Power supply box

Very important component as other component depends on it to for power. Take main electricity and turn them into lower voltage that ESP32 and Raspberry Pi require to operate. The specific model used in the project give the output of 12V 15A.

3.3.3.2 Relay Module (4 Channel)

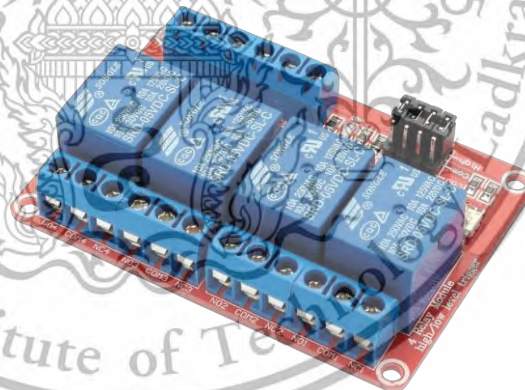


Figure 3. 9 4 Channel Relay Module

Relay act as a switch. It is control by microcontroller (ESP32 in this project) and is use to control other components in the workpiece. The relay receive signal from ESP32 which then toggle the operating state of the water pump, fan, and light.

3.3.3.3 LED Strip



Figure 3. 10 LED light strip

The LED provide light necessary for photosynthesis to the plant. The LED strip we choose is very efficient. Providing a lot of light for minimal power consumption. Its small size and light weight also make installing it on the workpiece relatively simple.

3.3.3.4 Water Pump

An important component use to move water from storage tank to use for watering the plant. The pump is a submerge pump type. Giving it high power and allowing it to move water over long distance.

3.3.3.5 Fan



Figure 3. 11 9x9cm fan

One factor we must control in the environment is air humidity. So, a fan is needed to help move the air out of the box to remove the humid air. Helping stabilize the environment. By doing so it also help to reduce the temperature of the box slightly as well.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.4 Designing

After deciding on what plant to grow. The scale of the project can be decided base on the maximum size of the plant.

3.4.1 Sketch

First, A rough sketch is drawn by hand. The size, shape, and position of the component are decided in this step. We conclude that we want the workpiece to be a box the size of 50x50x50 cm. This is where the plant pot will be placed in.

A box for circuit and microcontroller will be at the back on the same size that the fan will be at. With water storage tank in a separate box beside the circuit box.

Also housing the water sensor and pump. All three main box will then be place on a metal frame base for the added durability and to make sure all the box are leveled. It also allows for work access from the bottom side. Especially useful when working on the watering system.

3.4.2 Autodesk Inventor



Figure 3. 12 Autodesk Inventor app Logo

Inventor is a Computer Aided Design Software (CAD) develop by Autodesk. It is a program for design part and assembly use commonly in the industry. The program allow user to create part file (.ipt). Each representing individual component piece and let user put them together to create assembly file (.iam). In this project. The program is use to create each piece needed to create the box. Such as each side of the main plant box, and drilling position for hole.

Then combine multiple part to make an assembly file. All parts are put together to recreate the design from the hand drawn sketch. Once put together. A tool can be used to check for any conflict in the assembly. The conflict mean that the size or dimension of the part do not fit together or they overlap so they can't be put together if made according to the old size. Change can then be made to resolve any conflict before moving to the next step.

3.4.3 Drawing File

Another feature of Autodesk Inventor is the ability to create drawing file (.DWG) from assembly and parts. Drawing file are used to breakdown each part. Giving size and dimension to let other know how to recreate them.

All parts in the project are turn into drawing and is given to the rest of the team and act as a reference point so everyone now know what the final product should look like.

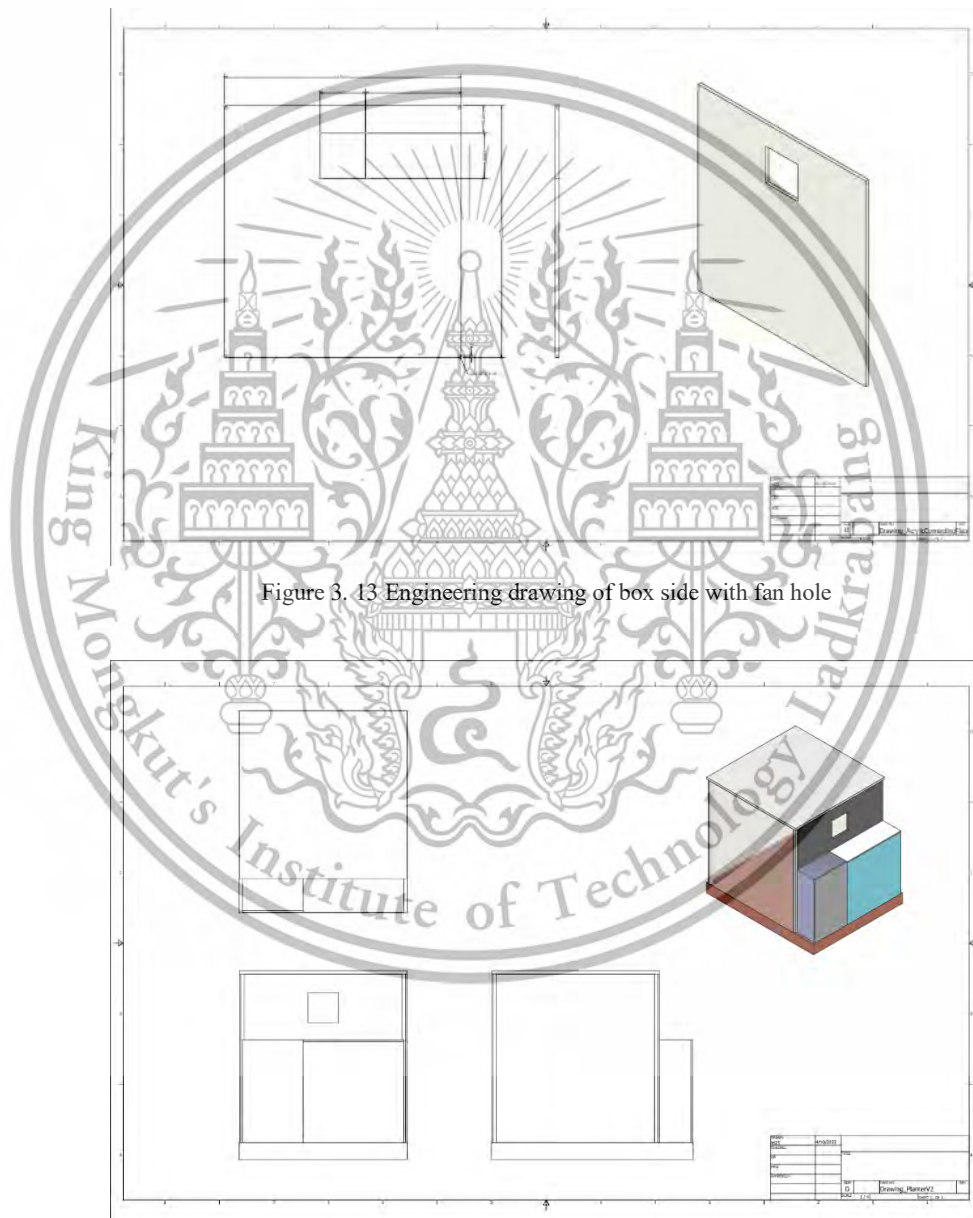


Figure 3. 13 Engineering drawing of box side with fan hole

Figure 3. 14 Engineering drawing of overall project

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.5 Hardware Development

3.5.1 Material Selection

Base: Must be strong and durable to support the weight of everything place on it. So aluminum was chosen for these reasons.

Box: The main purpose of the box is to hold the plant pot. So, it must be solid enough as to not bend underweight. Resist water and humidity due to work environment. But must also be transparent because it must allow sun light through for plant to conduct photosynthesis. And being easy to work with is also a consideration as the main material. With these in mind. Acrylic was chosen for its durability and characteristic for the expected work environment and ease of procurement.

3.5.2 Plant Box



Figure 3. 15 The plant box with a plant in it

The acrylics are cut according to the drawing. With one side having a square hole cut in the upper middle section for installing the fan. Each side are then put together and seal with silicone. The opposite side of fan is left open and install with magnet door lock so the box can be accessed once the assembly is finished.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.5.3 Circuit Box

Before cutting the acrylic. The component that will be put inside the circuit box are arrange to check for potential overlap. This is to account for potential error while assembling the box. Once the margin has been established. The acrylic is cut and put together similar to the plant box.



Figure 3. 16 Arranging position of hardware before circuit box assembly

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content²⁰ and cite the document when use.

3.5.4 Water Storage



Figure 3. 17 Inside the water tank with water sensor and pump

Beside the circuit box is the water storage box. The water is not fill directly into the acrylic box itself but into a plastic bucket instead. This is to provide space to install water sensor and pump as installing them directly on the acrylic wall and floor can cause leak which will negatively affect the overall system. The opening of the storage is the top lid instead of to the side like other boxes. This is to make refilling the water simple.

3.5.5 Base

The base is made of aluminum. This is because the base must be able to support the weight of all other component. For this part. We contract metal worker to weld the base for us. Providing them with the engineer drawing created from the 3D model.

3.6 System Design

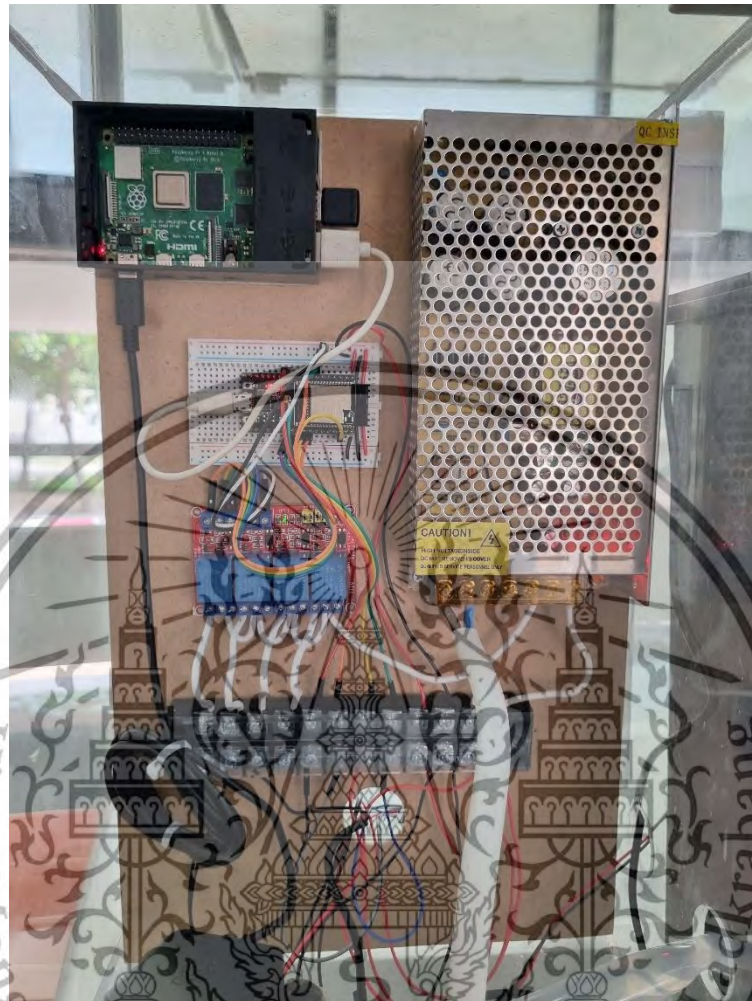


Figure 3. 18 Wiring in the circuit box

The figure above is a finished wiring of the hardware component used in the project. The following will breakdown how each component is connected to each other.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content²² and cite the document when use.

3.6.1 Raspberry Pi

RASPBERRY PI

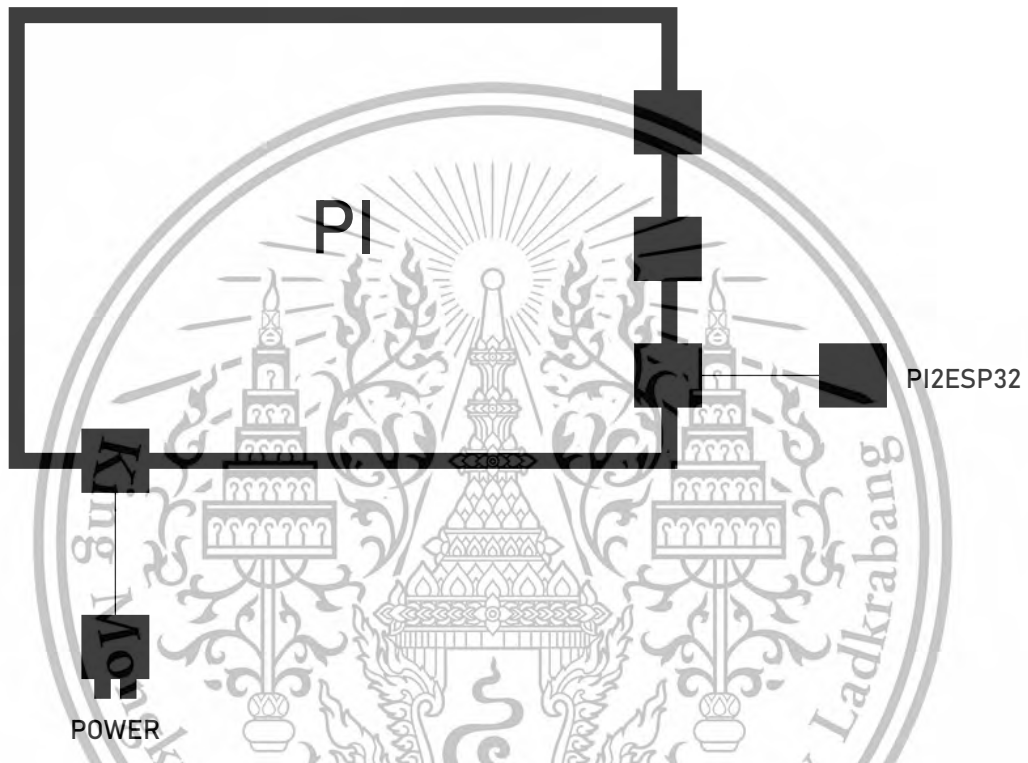


Figure 3. 19 Raspberry Pi Wiring Diagram

Raspberry Pi only needs two connections. One to the power and one with connection cable to ESP32.

3.6.2 Power Supply

POWER SUPPLY

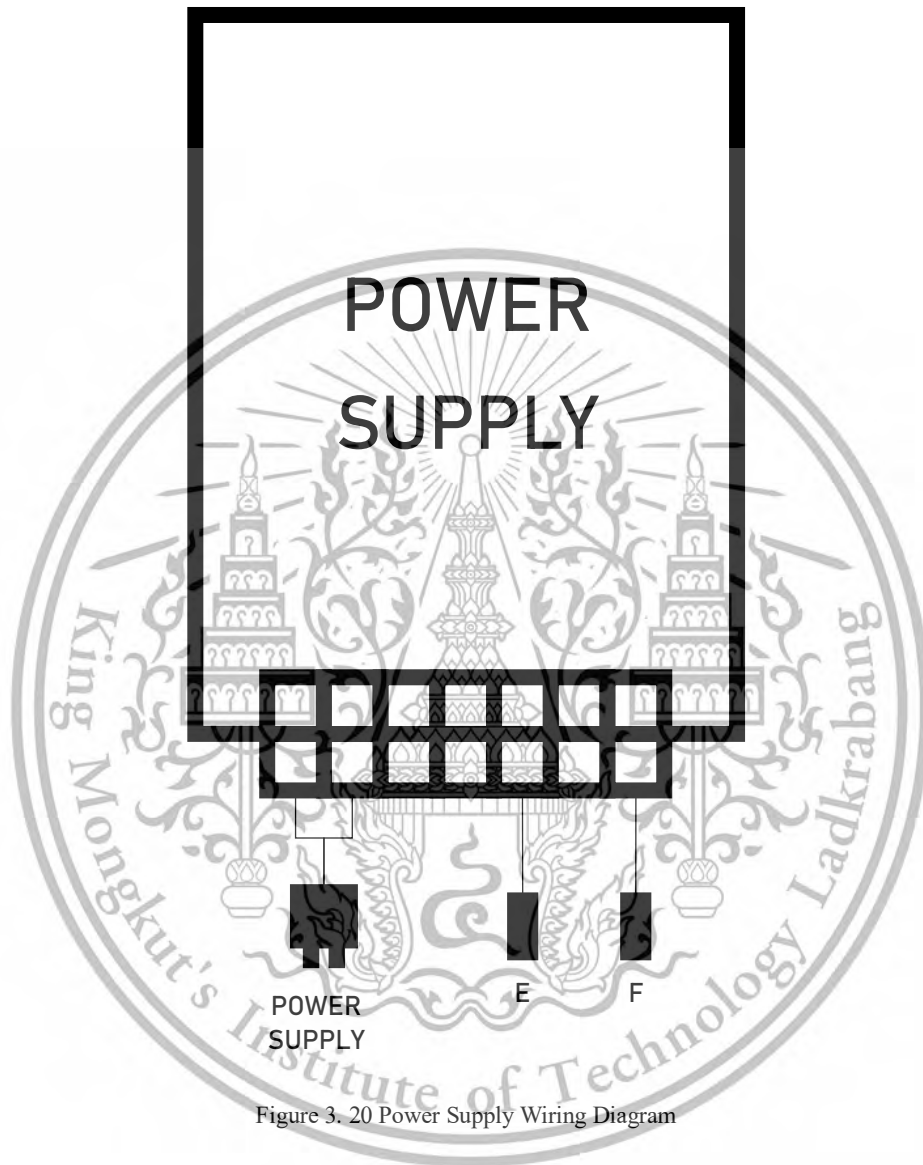


Figure 3. 20 Power Supply Wiring Diagram

Power supply line connect to the main electricity to convert them to lower voltage before passing them to relay with E, F line.

3.6.3 ESP32 Input/Output

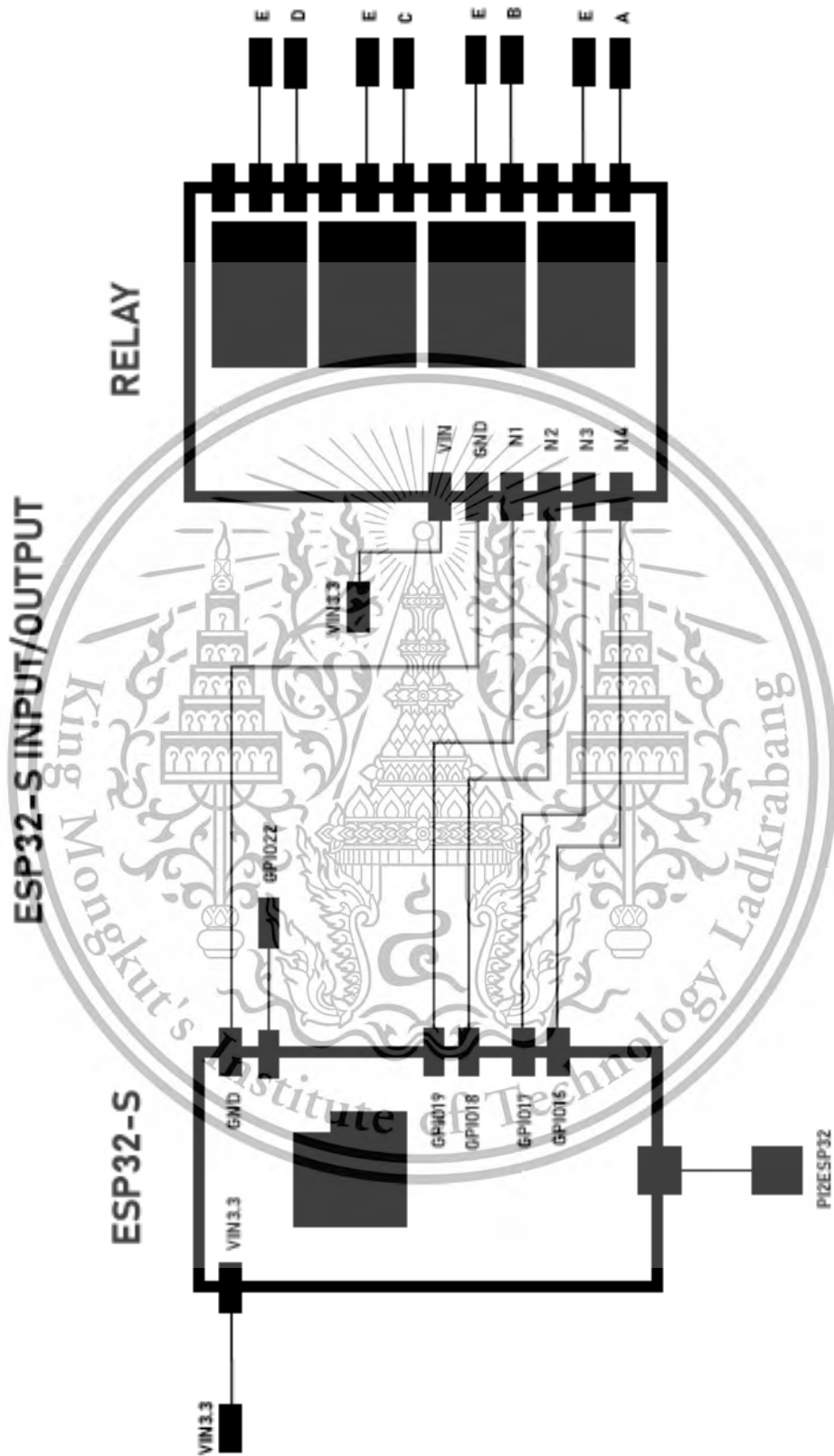


Figure 3. 21 ESP32 and Relay Wiring Diagram

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.6.4 Terminal

Terminal help organize wiring as such it is connect to all other hardware in the project.

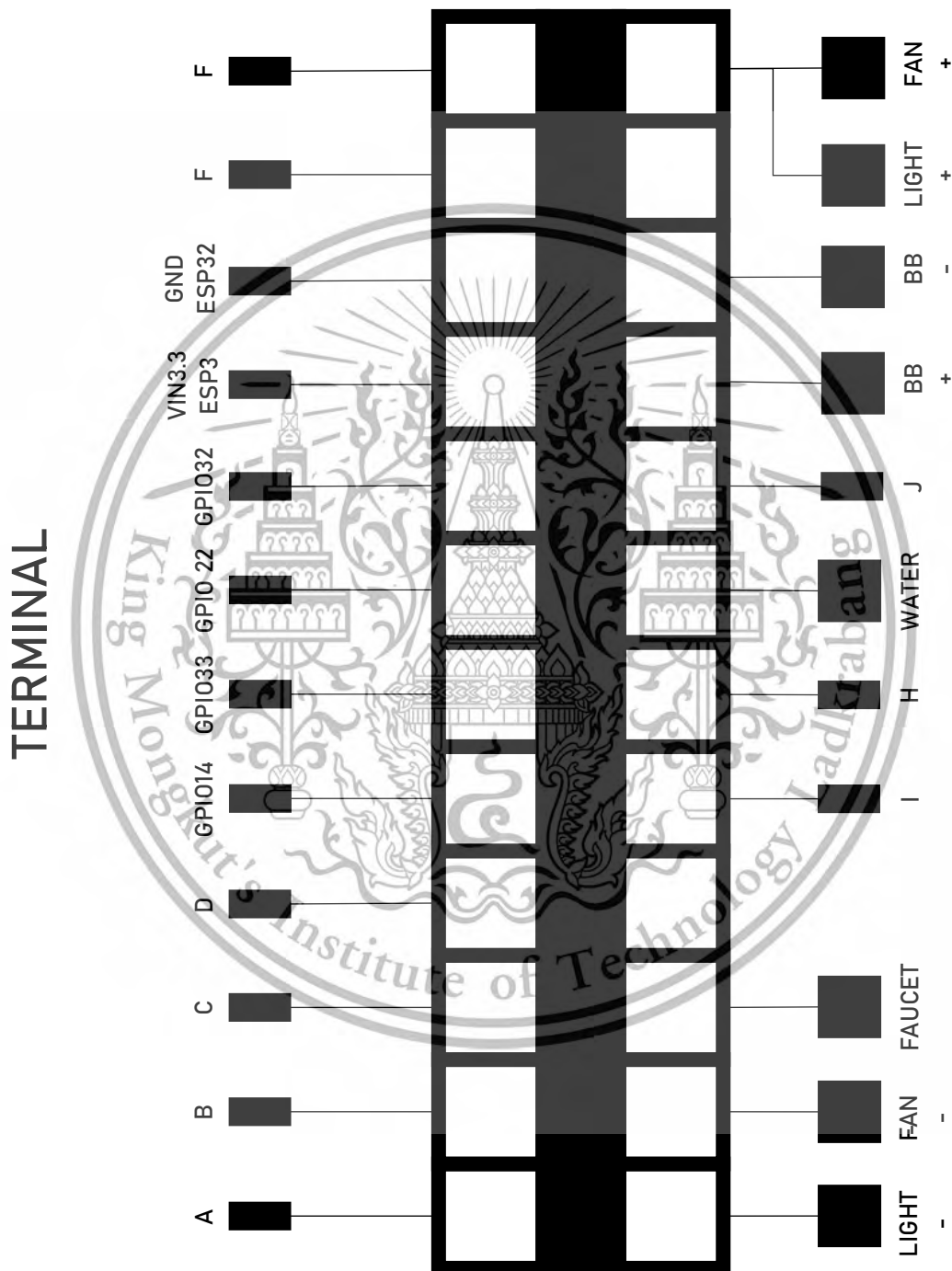


Figure 3. 22 Terminal Block Wiring Diagram

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.6.5 Sensors with Breadboard

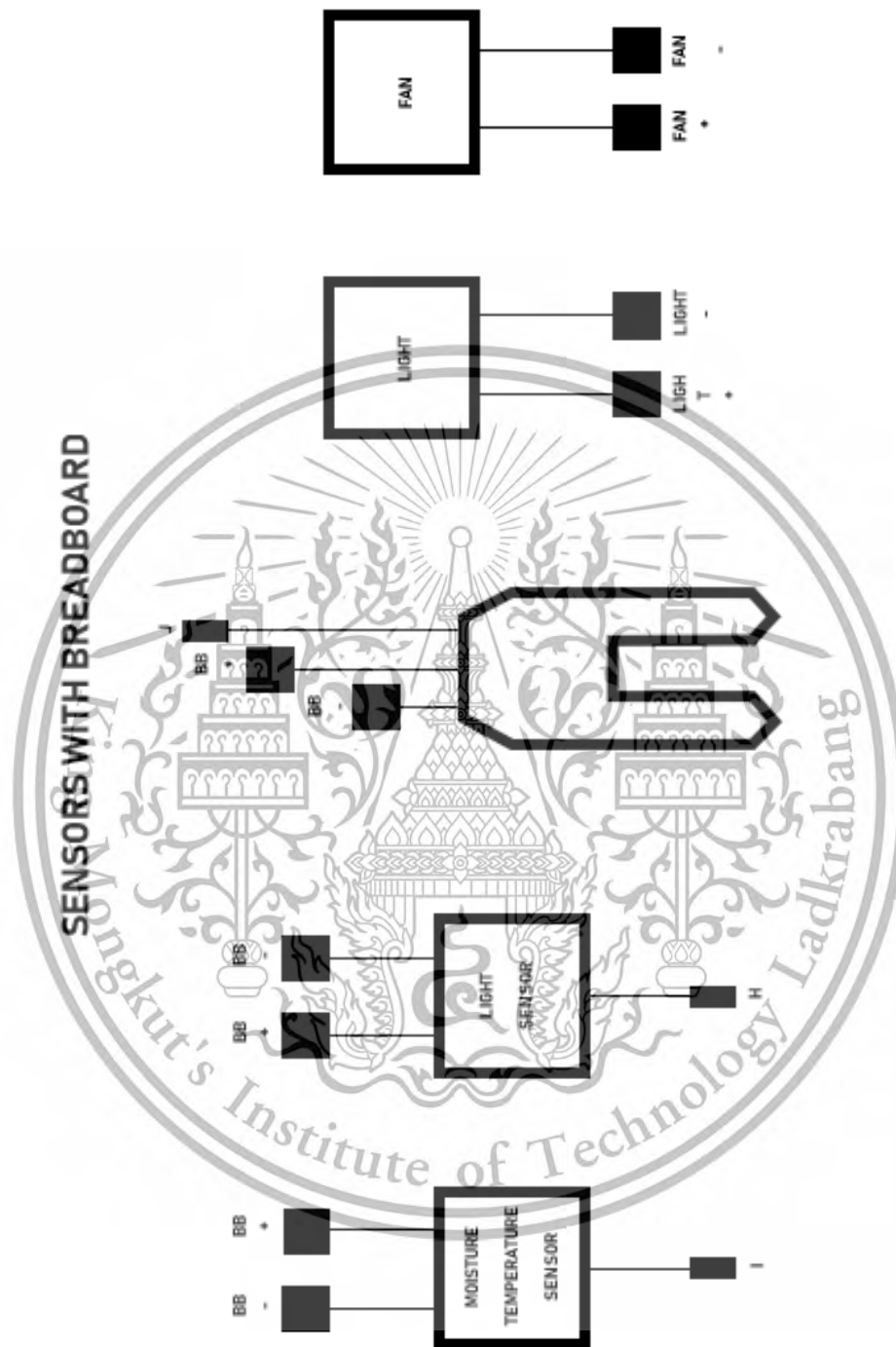


Figure 3. 23 Sensor Wiring Diagram

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.7 Software Development

3.7.1 ESP32

3.7.1.1 Code

3.7.1.1.1 Introduction

The following code snippet is written for an ESP32 microcontroller and is a part of this project to send the data to raspberry pi with air-humidity, soil-humidity, temperature, light, and water level sensors.

3.7.1.1.2 Purpose

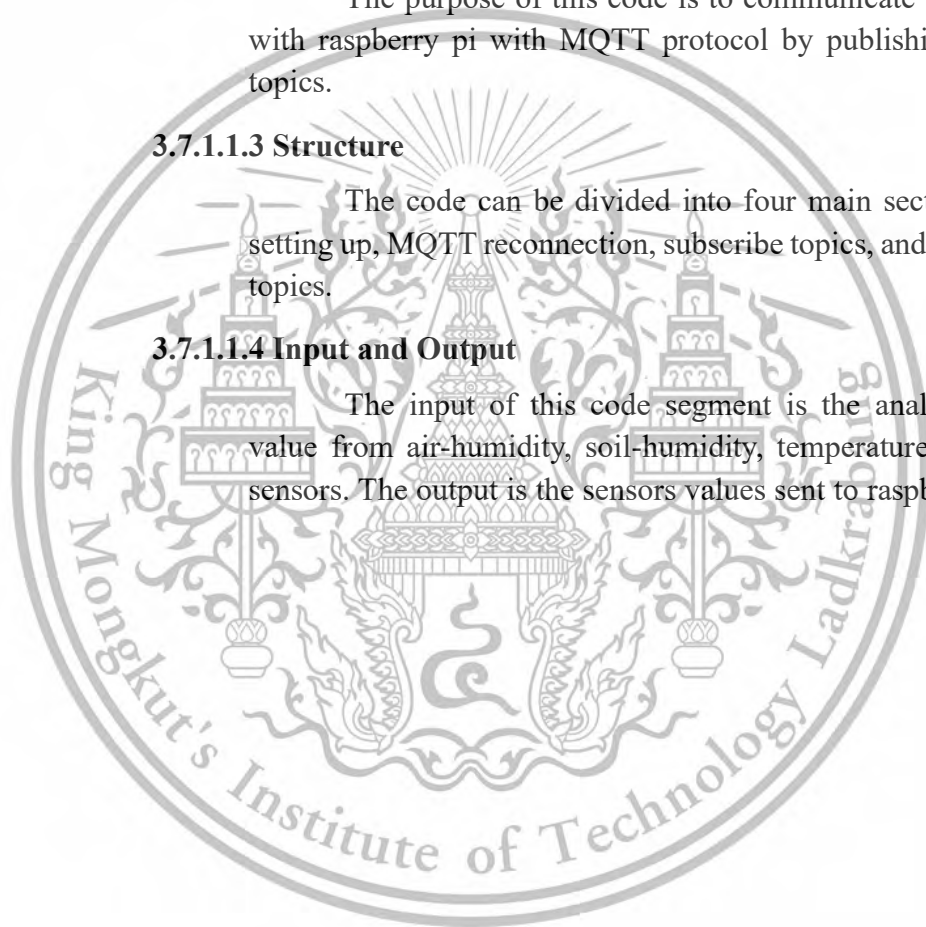
The purpose of this code is to communicate the sensors with raspberry pi with MQTT protocol by publishing sensors topics.

3.7.1.1.3 Structure

The code can be divided into four main sections: wi-fi setting up, MQTT reconnection, subscribe topics, and publishing topics.

3.7.1.1.4 Input and Output

The input of this code segment is the analog voltage value from air-humidity, soil-humidity, temperature, and light sensors. The output is the sensors values sent to raspberry pi.



3.7.1.1.5 Code Snippet

```
#include <Adafruit_Sensor.h>
#include <DHT_U.h>
#include <WiFi.h>
#include <PubSubClient.h>

//Defining Pins
#define DHTPIN 14
#define LIGHTPIN 33
#define WATERPIN 22
#define SOILPIN 32
#define OUT1 19
#define OUT2 18
#define OUT3 17
#define OUT4 16
```



```

//DHT parameters
#define DHTTYPE DHT11
DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;
int autom =0;

//MQTT Credentials
const char* ssid = "noo3";
const char* password = "AAAA1234";
//Check IP of Host (Raspberry PI)
const char* mqttServer = "192.168.46.120";
const char* mqttUserName = "atompI";
const char* mqttPwd = "raspberry";
const char* clientID = "ESP32-Decomplex";
const char* topicT = "esp32T";//publish topic
const char* topicH = "esp32H";//publish topic
const char* topicW = "esp32W";//publish topic
const char* topics = "esp32S";//publish topic
const char* topicL = "esp32L";//publish topic

//parameters for using non-blocking delay
unsigned long previousMillis = 0;
const long interval = 5000;
String msgStr = "";
float temp, hum,light,soil, water;

//setting up wifi and mqtt client
WiFiClient espClient;
PubSubClient client(espClient);

```

This material is reserved for educational use only, not allowed for commercial use

Forbidden to modify the content and cite the document when use.

```

void setup_wifi() {
  delay(10);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    if (client.connect(clientID, mqttUserName,
mqttPwd)) {
      Serial.println("MQTT connected");
      client.subscribe("output");
      Serial.println("Topic Subscribed");
    }
    else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000); // wait 5sec and retry
    }
  }
}
}
}
}

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

```

//Subscribe call back
void callback(char* topics, byte* payload, unsigned
int length) {
    Serial.print("Message arrived in topic: ");
    Serial.println(topics);
    Serial.print("Message:");
    String data = "";
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
        data += (char)payload[i];
    }
    Serial.println();
    Serial.print("Message size :");
    Serial.println(length);
    Serial.println();
    Serial.println("-----");
    Serial.println(data);
    if (autom == 1){
    if(data == "autoF"){
        autom = 0;
        Serial.println("Auto OFF");}
    }
    if (autom == 0){
    //state check
    if(data == "autoT"){
        autom = 1;
        Serial.println("Auto ON");
    }

    if(data=="OUT1T"){
        Serial.println("OUT1 ON");
        digitalWrite(OUT1, HIGH);
    }
    else if(data=="OUT1F"){
        Serial.println("OUT1 OFF");
        digitalWrite(OUT1, LOW);
    }
}
}

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

```
else if(data=="OUT2T"){
    Serial.println("OUT2 ON");
    digitalWrite(OUT2, HIGH);
}
else if(data=="OUT2F"){
    Serial.println("OUT2 OFF");
    digitalWrite(OUT2, LOW);
}
else if(data=="OUT3T"){
    Serial.println("OUT3 ON");
    digitalWrite(OUT3, HIGH);
}
else if(data=="OUT3F"){
    Serial.println("OUT3 OFF");
    digitalWrite(OUT3, LOW);
}
else if(data=="OUT4T"){
    Serial.println("OUT4 ON");
    digitalWrite(OUT4, HIGH);
}
else if(data=="OUT4F"){
    Serial.println("OUT4 OFF");
    digitalWrite(OUT4, LOW);
}
}
data="";
}
```

```
void setup() {
  Serial.begin(115200);
  // Initialize device.
  dht.begin();
  // get temperature sensor details.
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  dht.humidity().getSensor(&sensor);
  pinMode(OUT1, OUTPUT);
  digitalWrite(OUT1, LOW);
  pinMode(OUT2, OUTPUT);
  digitalWrite(OUT2, LOW);
  pinMode(OUT3, OUTPUT);
  digitalWrite(OUT3, LOW);
  pinMode(OUT4, OUTPUT);
  digitalWrite(OUT4, LOW);
  pinMode(WATERPIN, INPUT_PULLUP);

  setup_wifi();
  //setting MQTT server
  client.setServer(mqttServer, 1883);
  //defining function which will be called
  when message is received.
  client.setCallback(callback);
}
```

```
void loop() {
    if (!client.connected()) { //if client is
not connected
        reconnect(); //try to reconnect
    }
    client.loop();
    unsigned long currentMillis = millis();
    //read current time
    if (currentMillis - previousMillis >=
interval) { //if current time - last time > 5
sec
        previousMillis = currentMillis;
        //read temp and humidity
        sensors_event_t event;
        dht.temperature().getEvent(&event);
        if (isnan(event.temperature)) {
            Serial.println(F("Error reading
temperature!"));
        }
        else {
            Serial.print(F("Temperature: "));
            temp = event.temperature;
            Serial.print(temp);
            Serial.println(F("°C"));
        }
        // Get humidity event and print its
value.
        dht.humidity().getEvent(&event);
        if (isnan(event.relative_humidity)) {
            Serial.println(F("Error reading
humidity!"));
        }
    }
}
```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

```

else {
    Serial.print(F("Humidity: "));
    hum = event.relative_humidity;
    Serial.print(hum);
    Serial.println(F("%"));
}

light = float(analogRead(LIGHTPIN));
Serial.print(F("Light: "));
Serial.println(light);
soil = float(analogRead(SOILPIN));
Serial.print(F("Soil: "));
Serial.println(soil);
water = float(digitalRead(WATERPIN));
Serial.print(F("Water: "));
Serial.println(digitalRead(WATERPIN));

Payload(topic,t,emp);
Payload(topic,h,hum);
Payload(topic,l,light);
Payload(topic,s,soil);
Payload(topic,w,water);

Serial.println(autom);
delay(50);
}
}

void Payload(const char* topic,float y){
    msgStr = String(y);
    byte arrSize = msgStr.length() + 1;
    char msg[arrSize];
    Serial.print("PUBLISH DATA:");
    Serial.println(msgStr);
    msgStr.toCharArray(msg, arrSize);
    client.publish(topic, msg);
    msgStr = "";
}
}

```

3.7.1.1.6 Explanation

The code begins by including the necessary libraries: 'DHT_U' for the DHT sensor, 'WiFi' for Wi-Fi connectivity, and 'PubSubClient' for MQTT communication.

The code defines the DHT sensor type as 'DHT11' and creates an instance of the 'DHT_Unified' class called 'dht'.

Several variables and constants are declared, including the delay time, an automation flag ('autom'), and MQTT credentials and topics.

The code defines the Wi-Fi client ('espClient') and MQTT client ('client'). The 'setup_wifi' function is responsible for connecting the ESP32 to the Wi-Fi network.

The 'reconnect' function attempts to establish an MQTT connection with the MQTT server specified by 'mqttServer'.

The 'callback' function is a callback for when messages are received. It prints the received message and performs specific actions based on the received data.

In the 'setup' function, the serial communication is initialized, the DHT sensor is initialized, and pin modes are set for output pins.

The 'loop' function is the main program loop that runs continuously. It checks if the MQTT client is connected and calls 'client.loop' to handle incoming MQTT messages.

Inside the 'loop' function, there is a check to determine if a certain time interval (5 seconds) has passed. If so, sensor readings are obtained, and the data is published to specific MQTT topics using the 'Payload' function.

The 'Payload' function prepares the data to be published to MQTT by converting it to a string and then publishing it to the specified topic using 'client.publish'.

3.7.1.2 Flow Chart

This flow chart, as shown in this figure 3.25, is used to illustrate how ESP32 subscribes and publishes the data to the raspberry pi.

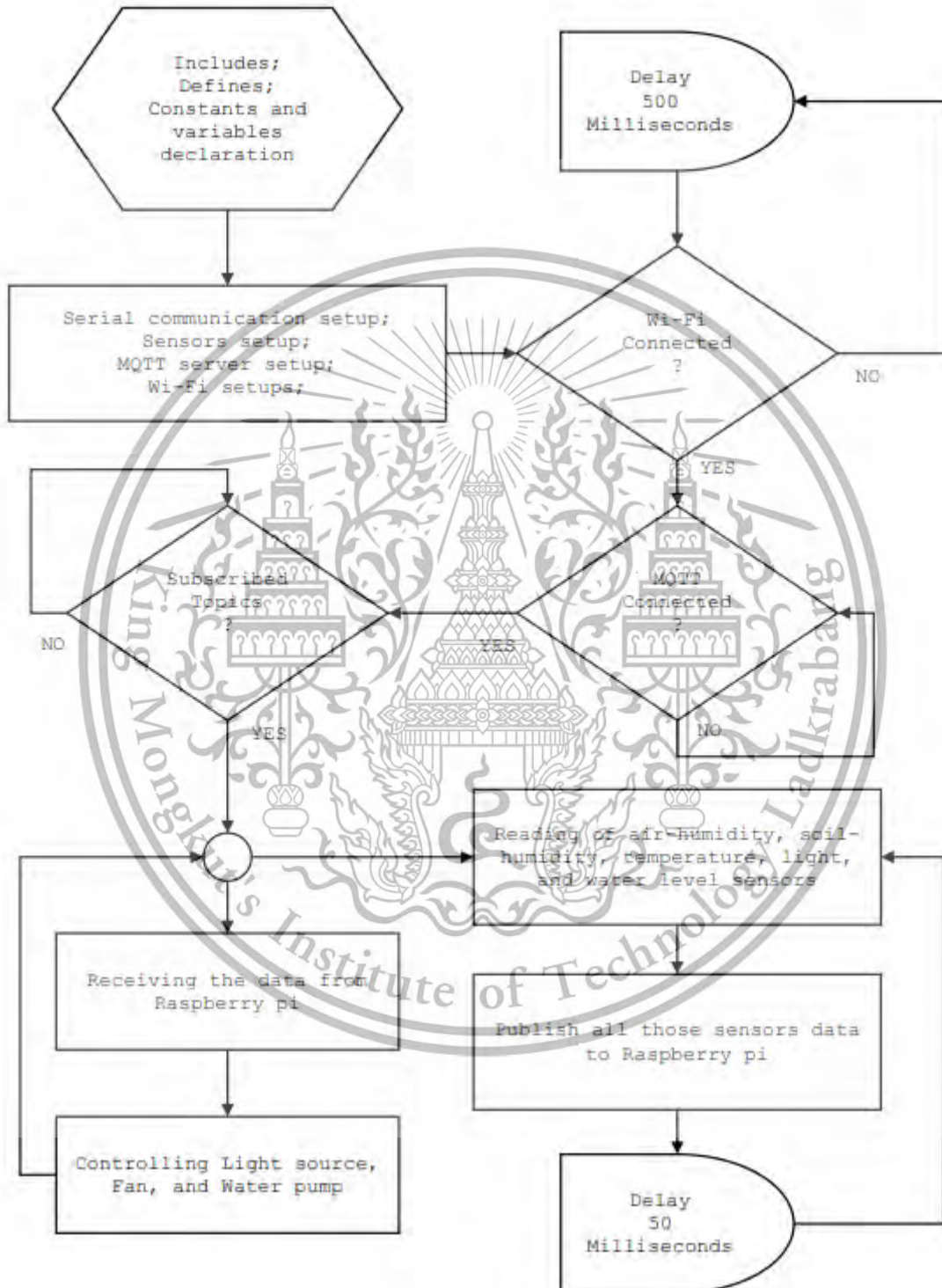


Figure 3. 24 ESP32 code flowchart

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.7.2 Firebase

Firestore is a mobile and web application development platform that provides backend services and tools. It offers many of features, including real-time database, cloud storage, authentication, hosting, and more. We applied authentication and real-time database services for this project.

3.7.2.1 Authentication

Firestore can provide us the authentication system including sign-in method, email verification, password reset, email change, and also short message service (SMS) verification. For sign-in method, firestore provides us native providers, additional providers, and custom providers. In this project, we used Email/Password as a sign-in method.

3.7.2.2 Real-time Database

Real-time database is a NoSQL cloud-hosted database that designed to handle real-time data synchronization and for applications that require instant updates. It uses a JSON-based data model which allows for hierarchical and flexible data structures. This is our data structure, as shown in figure 3.26, is used for storing the input and output data in real-time. Control collection contains switch status as Boolean data type and Sensor collection contains values of the sensors as double data type.

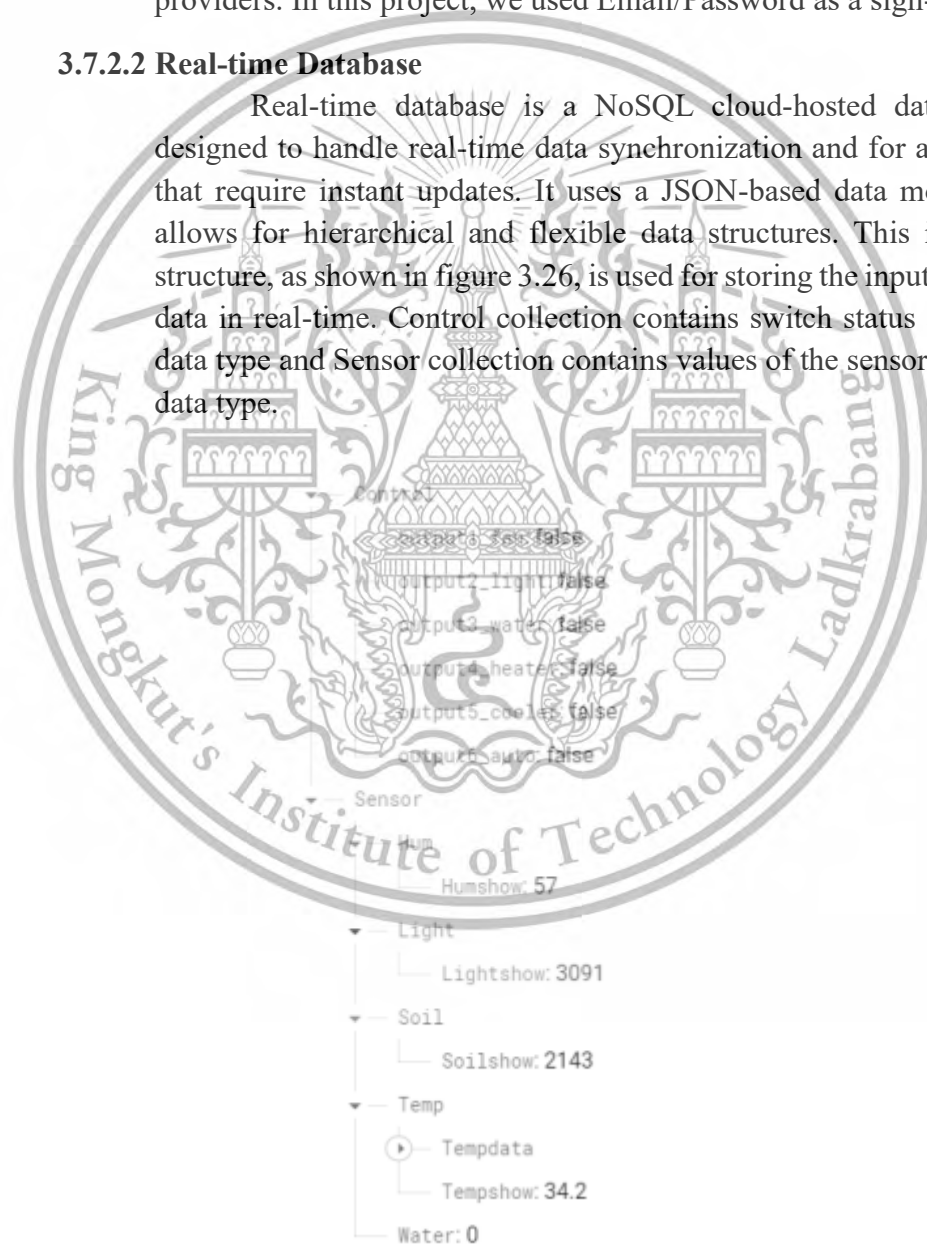


Figure 3. 25 Data structure in Firebase

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.7.3.2 Showing Data

These nodes, as shown in Figure 3.28, are connected to the display node, and sending data nodes. The display nodes are used to show the graph in Node-Red UI. The sending data node is used to send the data to the real-time database of Firebase.

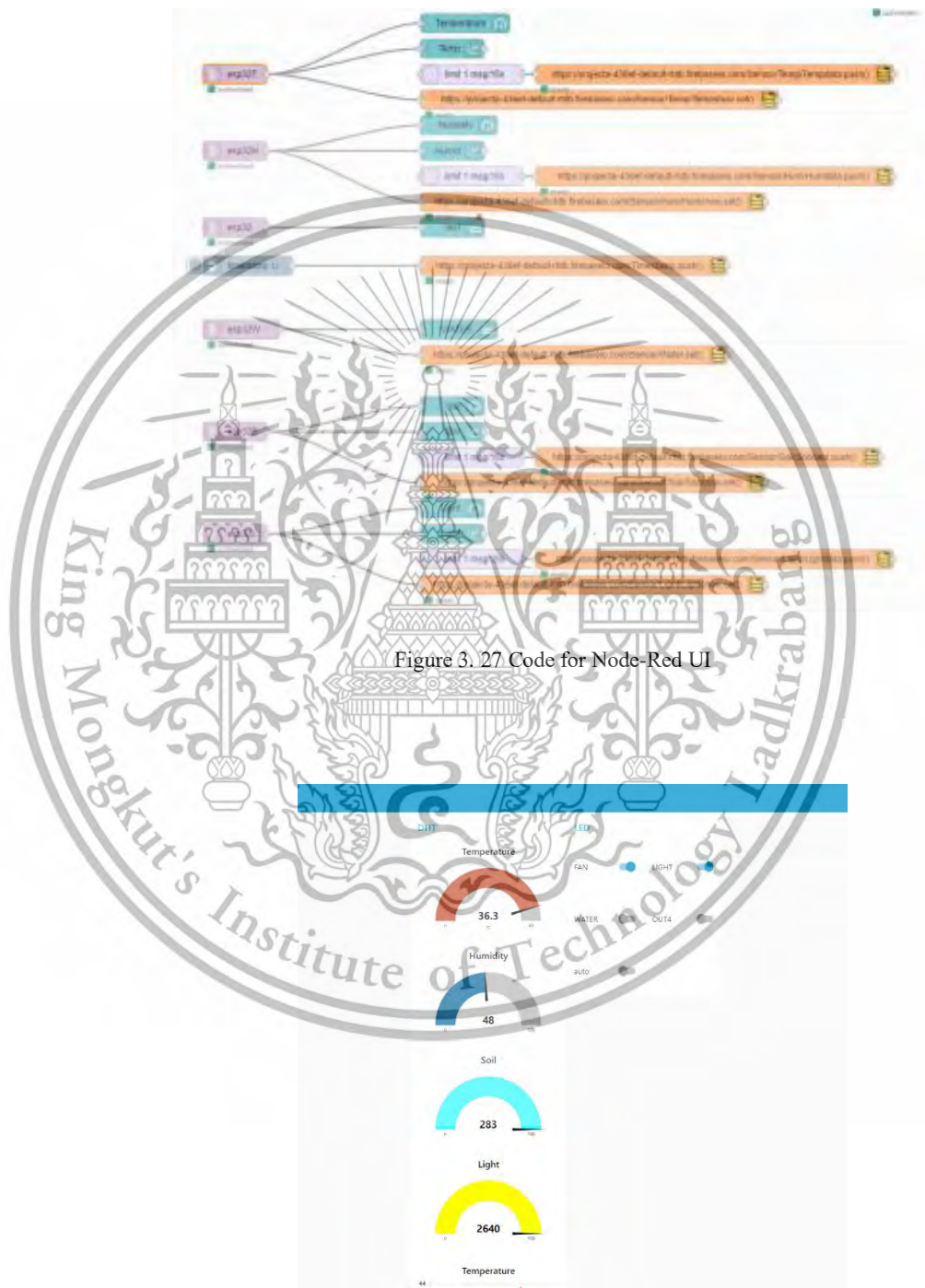


Figure 3. 27 Code for Node-Red UI

Figure 3. 28 UI page from Node-Red

3.7.4 Flutter

Flutter is an open-source software UI toolkit developed by Google for building applications for web, mobile, and desktop platforms from a single codebase. Flutter uses Dart programming language and offers hot-reload features for instant code changes, many pre-built UI components, and comprehensive community support. Flutter also provides developers to build their styles and fast apps that run smoothly on various platforms, making it an efficient tool for cross-platform application development.

3.7.4.1 Introduction

This mobile application is proposed to show and control the dashboard of the smart pot integrating with IoT devices. The dashboard can illustrate the temperature, air humidity, soil humidity, light, and water level in digits and characters. The user can control the device for lighting, watering, and removing moisture. This application also includes a register system and login system for users to be able to connect their own devices.

3.7.4.2 Overall Design

The design of this application has four main parts: registration, login, dashboard, and control. The registration part, containing the register screen and the register-complete screen, is used for creating an account for using the application. The login part is used for logging in to the application to access the dashboard and control pages. The dashboard part is used for illustrating the sensors' data: temperature, air humidity, soil humidity, light intensity, and water level, and also showing the data in digits and horizontal bar graphs. The control part has buttons and sliders that are used to control the device, such as opening the light source, watering the plant, and opening the fan to remove moisture. This application uses white and blue as a theme of the application.

3.7.4.2.1 Registration

The design in the registration part contains large 'SIGN UP' text, conditions text for creating an account, text form fields, a notifiable warning text, a login button, a register button, and a back button.

The large 'SIGN UP' text is used for telling the users when they are in the registration part. This text design should be easy to notice, and this text color is blue to contrast with the white background.

The conditions text is used for telling the user the conditions of creating the account. This text is specified only in password conditions containing eight characters long, at least one uppercase and lowercase character, and an extra character such as underscore, octothorpe, and more.

The text form fields are separated into three forms: email, password, and confirm password. Each text form field has a different icon and hint text used for telling the user to complete the form correctly. The icon in each text form field has blue color used to contrast with the white background. The password and confirm password text form fields have eye icons at the end of the form for showing and hiding the password when the users fill out the form.

The notifiable warning text is used to tell the user to check their forms when filling out the forms completely. It has a grey color that makes this text inconspicuous.

The login button is used to go back to the login screen when the users already have their accounts. It has small text which is not cluttered, and the blue color is accessible to find it easily.

The register button is used to submit the text form fields. It has a large button that is easy to notice, and a blue color is not only the application theme but also contrasts with the white background of the registration screen.

The back button is the same purpose as the login button, but it has a larger text and uses grey color. It is the alternative way to go back to the login screen.

The registration screen, as shown in Figure 3.30, is applied on the mobile application with the Android operating system.

This material is reserved for educational use only, not allowed for commercial use.

When the users finish filling and submitting the forms, the complete registration screen, as shown in Figure 3.31, will show up to inform the users. This screen contains a success icon, text, and back buttons. The texts show the status to tell the users that the account has been successfully created. The users can use the back button to return to the login screen. The register button also has a large button and a blue color button which contrasts with the white background and is easy to notice.

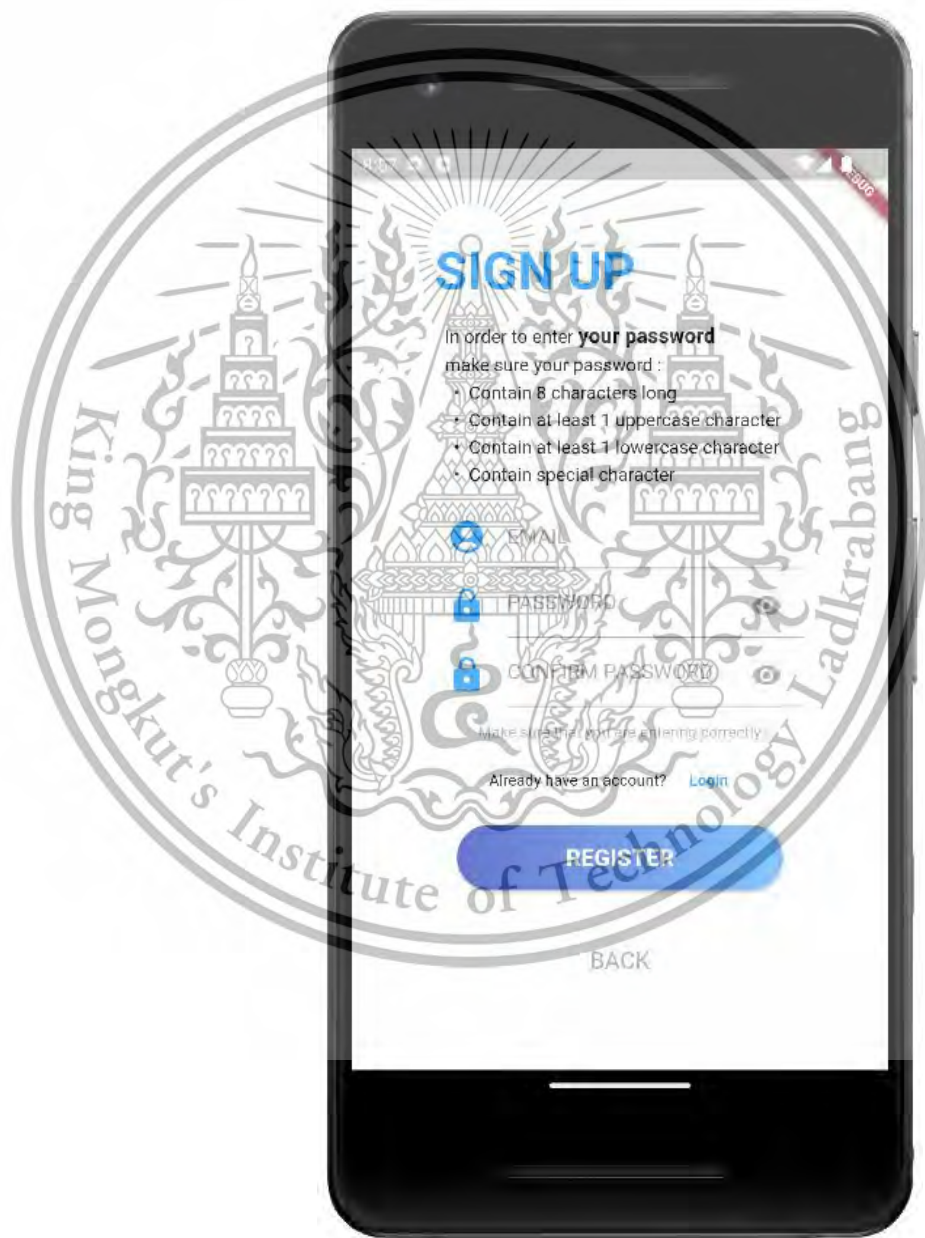


Figure 3. 29 Decomplex app registration page

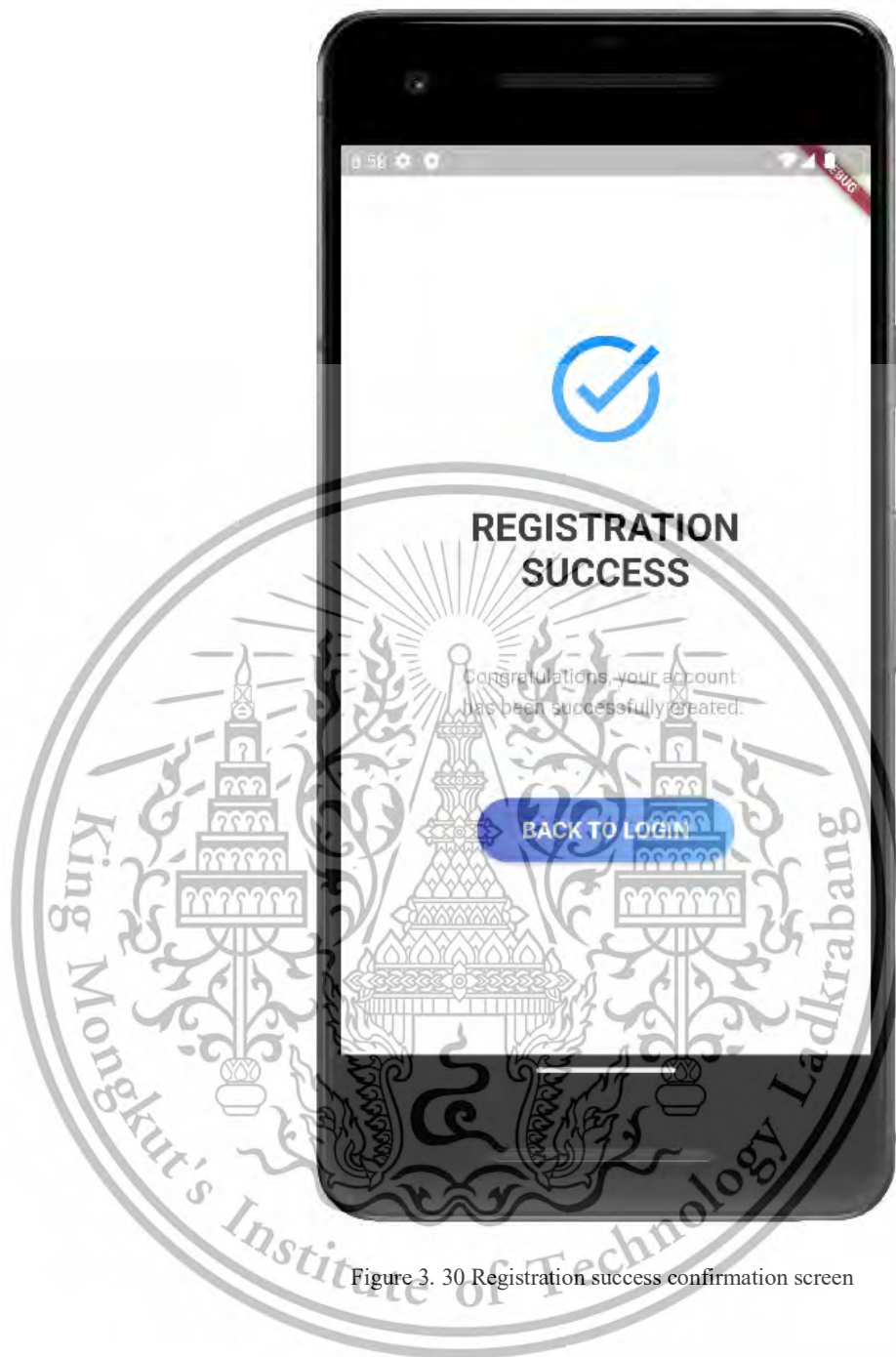


Figure 3. 30 Registration success confirmation screen

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

3.7.4.2.2 Login

The design in the login part, as shown in Figure 3.32, contains a large logo, welcome text, text form fields, remember me check box, a signup button, and a login button.

The large logo of Decomplex represents the application used to illustrate the dashboard sensors' data and control the actuators. There is a welcome text used to welcome the users to this application.

The text form fields consist of email and password text form fields. Users can fill in their email and password in these text form fields to access this application.

The remember me check box is used to save the email and password inputs. It is a small box that can be checked for memorizing the email and password in text form fields.

The signup button is used to enter the registration screen when the users want to create their accounts for use in this application. It has small text which is not cluttered, and the blue color is accessible to find it easily. The text on the left side of this button is used to hint to the users.

The users can use the login button which is at the bottom of the screen, to access the application by filling out the email and password, then press this button. This button also has a large button and a blue color button which contrasts with the white background and is easy to notice.



Figure 3. 31 Log-in screen demonstration

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

3.7.4.2.3 Dashboard

The dashboard design, as shown in Figure 3.33, is used to illustrate the sensors' data, including temperature, air humidity, soil humidity, light intensity, and water capacity.

The first block contains temperature and water capacity sensors' data. The temperature data is displayed in decimal with two precisions of Celsius unit, which is easy to understand. The water capacity data is shown in a text which has two conditions: full and empty. The water capacity status text will show a blue 'FULL' text when the device has water in the water capacity. In vice versa, the water capacity status text will show a red 'EMPTY' text when the device has no water left in the water capacity.

The second block contains the text, an icon, and a horizontal bar chart. The texts are used to name this usage of this block, indicate the bar chart, and show the light sensor data in percent. The icon of this block is the Sun, which is implied to be about the light intensity. The horizontal bar chart has a grey background color and fills with orange-yellow color that is used to match the Sun icon.

The third and fourth blocks contain text, an icon, and a horizontal bar chart similar to the second block, but the air humidity and soil humidity are displayed respectively.

The fifth block is used for showing the oxygen sensor's data which is not included in this project. It is used for further development with the oxygen sensor integrating with the devices.

This page has navigation bar at the bottom center of this screen which is used to switch between dashboard and control screen.

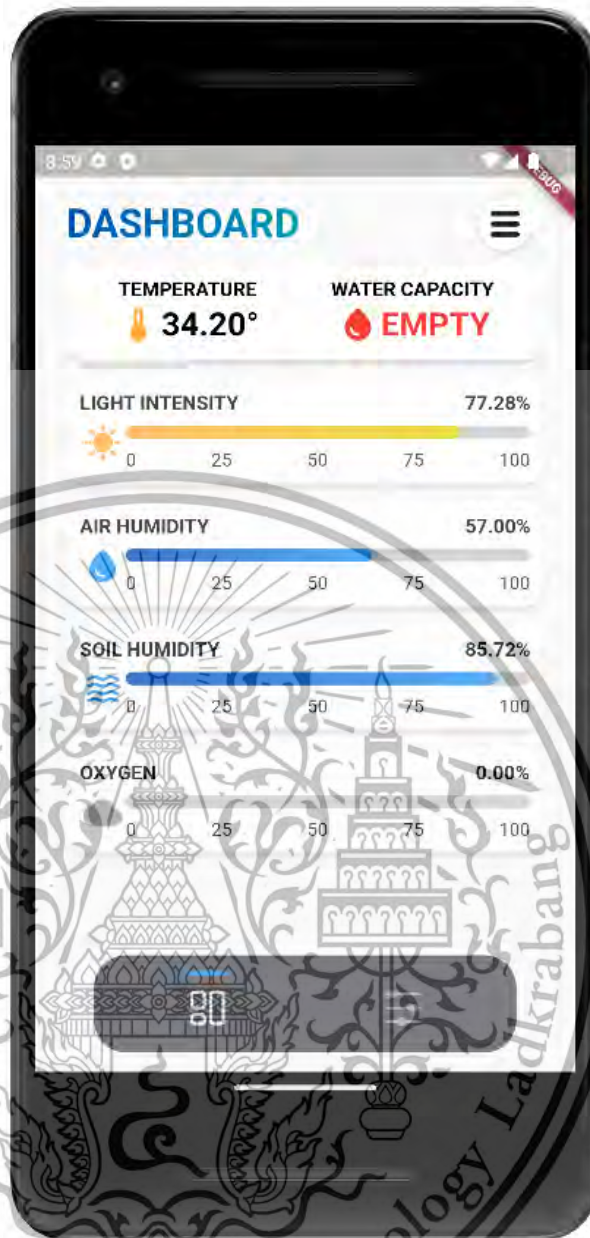


Figure 3. 32 Dashboard in the application showing data reading from the workpiece

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

3.7.4.2.4 Control

The control screen, as shown in Figure 3.34, is used to control the actuators: light source, fan, and water pump. The control design has three main blocks.

The first block contains the buttons that are used to control the actuators. Each button has text under the button which indicate the actuator to be used. When the users press on the button, the button will change the color from white color to blue color and the center text of the button will change from 'OFF' status to 'ON' status, accordingly.

The second block is used to control the temperature that is not included in this device.

The third block is used to control fan speed which is also not included in this device.

This page has navigation bar at the bottom center of this screen which is used to switch between dashboard and control screen.

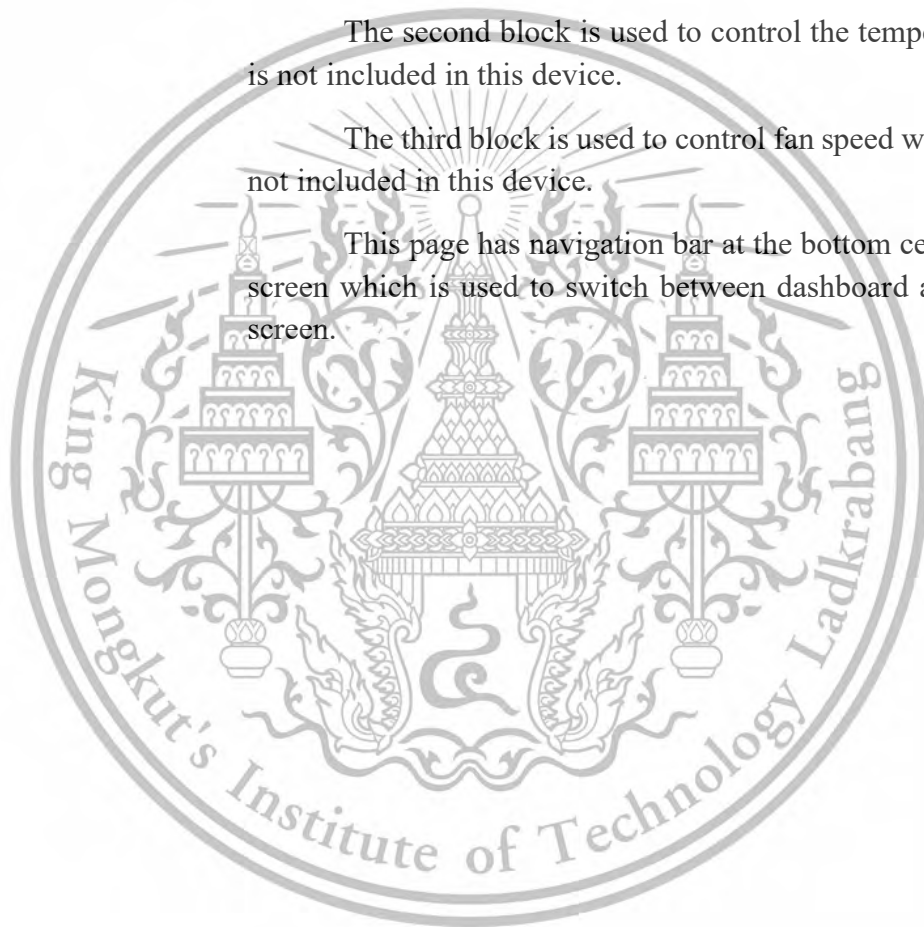




Figure 3. 33 Control console

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.7.4.2.5 Side navigation bar

The side navigation bar is a button that will be able to pop up the side navigation bar by pressing this button. This side navigation bar contains user email for telling the users their email that used in this application. It also contains browse and user detail header text. In the browse header is contain 'HowToUse' button which is used to guide the users how to use this device. The user detail header contains setting and sign out buttons.



Figure 3. 34 Side Navigation Bar

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.7.4.3 Navigation

3.7.4.3.1 Login Screen

The login screen contains sign-up and login buttons. The sign-up button can navigate to the registration screen, and the login button can navigate to the dashboard screen.

3.7.4.3.2 Registration Screen

The registration screen contains login, register, and back buttons. The login button is used to navigate to the login screen. The register button is used to navigate to the registration complete screen. The back button is used to navigate to the login screen.

3.7.4.3.3 Registration Complete Screen

This page has only one navigation button which is used to navigate to the login screen.

3.7.4.3.4 Dashboard Screen

The dashboard screen contains a side navigation bar and a bottom navigation bar button. The side navigation bar is used to navigate to pop up the side navigation bar. The bottom navigation bar is used to switch between the dashboard and the control screen.

3.7.4.3.5 Control Screen

The control screen contains a side navigation bar and actuators' buttons. The side navigation bar is used to navigate to pop up the side navigation bar. The actuators' buttons are used to control the actuators: light source, fan, and watering.

3.7.4.3.6 Side Navigation Bar

The side navigation bar contains only sign-out buttons. The sign-out button is used to navigate to the login screen.

3.7.4.3.7 Navigation Chart

The navigation chart, as shown in Figure 3.36, is used to describe how the screen can navigate to the others.

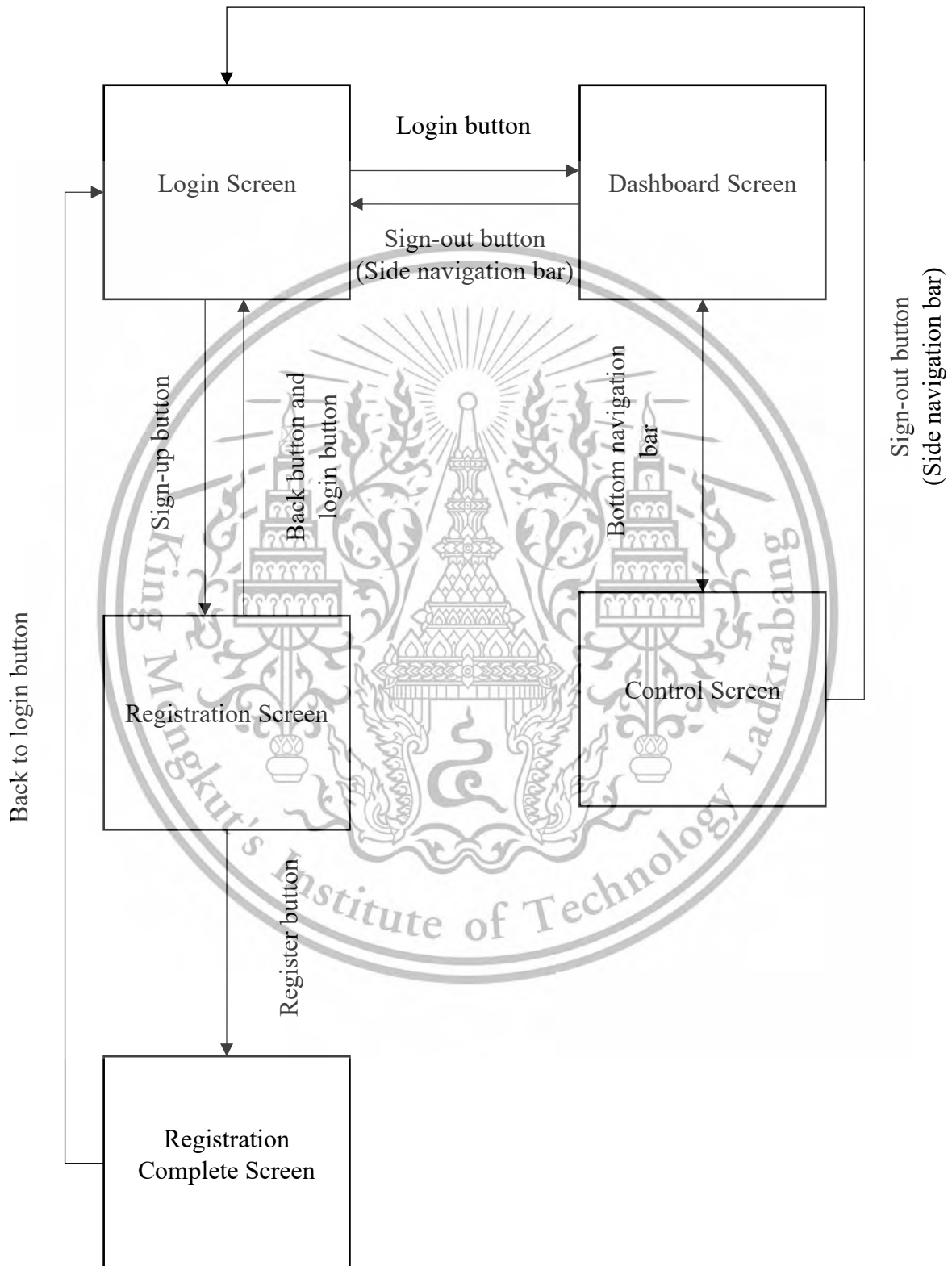


Figure 3. 35 Chart showing possible navigation of each screen

This material is reserved for educational use only, not allowed for commercial use.

3.7.4.4 Input and Output Elements

This application has input elements in three main parts: registration, login, and control screens. The output elements in the dashboard screen display the sensors' data. The registration screen has three text form fields as input elements for email, password, and confirm password. The login screen has two text form fields as input elements for email, and password. The control screen has buttons as input elements for controlling the actuators.

3.7.4.5 Dependencies

3.7.4.5.1 Rive Icons

Rive icons dependency allows this application to use the rive asset which is provide animated icons.

3.7.4.5.2 Flutter Launcher Icons

Flutter launcher icons dependency allow this application to use the Decomplex logo as an icon in mobile application.

3.7.4.5.3 Form Field Validator

Form field validator dependency is used to set the password conditions for the users in registration screen.

3.7.4.5.4 Page Transition

Page transition dependency allows this application to transit the screen smoothly and various style in transit.

3.7.4.5.5 Firebase Core

Firebase core dependency allows this application to connect to multiple Firebase apps.

3.7.4.5.6 Firebase Authentication

Firebase authentication dependency provides the login system to this application including reset password, change email, and SMS verification.

3.7.4.5.7 Firebase Database

Firebase database dependency allows this application to store the sensors' data and actuators' status in the real-time database.

3.7.4.5.8 Firebase Messaging

Firebase Messaging dependency allows this application to notification the users.

3.7.4.6 Code Explanation

3.7.4.6.1 animated_bar.dart

```
import 'package:flutter/material.dart';

class AnimatedBar extends StatelessWidget {
  const AnimatedBar({
    super.key,
    required this.isActive,
  });

  final bool isActive;

  @override
  Widget build(BuildContext context) {
    return AnimatedContainer(
      duration: const Duration(milliseconds:
200),
      margin: const EdgeInsets.only(bottom: 2),
      height: 4,
      width: isActive ? 30 : 0,
      decoration: const BoxDecoration(
        color: Colors.blue,
        borderRadius: BorderRadius.all(
          Radius.circular(12),
        ),
      ),
    );
  }
}
```

This `animated_bar.dart` script held the component for animated bar which is navigation bar in both dashboard and control screen.

3.7.4.6.2 side_menu_tile.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import
'package:projectiot/model/rive_asset.dart';
import 'package:rive/rive.dart';

class SideMenuTile extends StatelessWidget {
  const SideMenuTile({
    super.key,
    required this.menu,
    required this.press,
    required this.riveonInit,
    required this.isActive,
  });

  final RiveAsset menu;
  final VoidCallback press;
  final ValueChanged<Artboard> riveonInit;
  final bool isActive;

  @override
  Widget build(BuildContext context) {
    return Column(
      children: [
        Padding(
          padding: const EdgeInsets.only(left:
24),
          child: Divider(
            color: Colors.white24,
            height: 1,
          ),
        ),
        Stack(
          children: [
            AnimatedPositioned(
              duration: Duration(milliseconds:
200),
              curve: Curves.fastOutSlowIn,
              height: 56,
              width: isActive ? 288 : 0,
              left: 0,
              child: Container(
                decoration: BoxDecoration(
                  color: Colors.blue,
                  borderRadius:
BorderRadius.all(Radius.circular(10)),
                ),
              ),
            ),
            ListTile(
              onTap: press,
              leading: SizedBox(
                height: 34,
                width: 34,
                child: RiveAnimation.asset(
                  menu.src,
                  artboard: menu.artboard,
```

This material is reserved for educational use only, not allowed for commercial use.

```

        onInit: riveonInit,
      ),
    ),
    title: Text(
      menu.title,
      style: TextStyle(color:
Colors.white),
    ),
  ),
  ],
),
],
);
}
}

class InfoCard extends StatelessWidget {
  const InfoCard({
    super.key,
    required this.name,
    required this.subtitle,
  });

  final String name, subtitle;

  @override
  Widget build(BuildContext context) {
    return ListTile(
      leading: CircleAvatar(
        backgroundColor: Colors.white24,
        child: Icon(
          CupertinoIcons.person,
          color: Colors.white,
        ),
      ),
      title: Text(
        name,
        style: TextStyle(color: Colors.white),
      ),
      subtitle: Text(
        subtitle,
        style: TextStyle(color: Colors.white),
      ),
    );
  }
}

```

The `side_menu_tile.dart` contains the icons from `rive` assets which is the part inside the side menu navigation bar.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content⁵⁸ and cite the document when use.

3.7.4.6.3 side_menu.dart

```
import
'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import
'package:page_transition/page_transition.dart';
import
'package:projectiot/components/side_menu_tile.dart';
import
'package:projectiot/model/rive_asset.dart';
import
'package:projectiot/utils/rive_utils.dart';
import 'package:rive/rive.dart';

import '../screen/LoginSession/login.dart';

class SideMenu extends StatefulWidget {
  const SideMenu({super.key});

  @override
  State<SideMenu> createState() =>
  _SideMenuState();
}

class _SideMenuState extends State<SideMenu> {
  final auth = FirebaseAuth.instance;
  RiveAsset selectMenu = sideMenus.first;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: 288,
        height: double.infinity,
        color: Color(0xff1d1e26),
        child: SafeArea(
          child: Column(
            crossAxisAlignment:
            CrossAxisAlignment.start,
            children: [
              InfoCard(
                name: "USER",
                subtitle:
                "${auth.currentUser?.email!}",
              ),
              Padding(
                padding: const
                EdgeInsets.only(left: 24, top: 32, bottom: 16),
                child: Text(
                  "Browse".toUpperCase(),
                  style: Theme.of(context)
```

```

        .textTheme
        .titleMedium!
        .copyWith(color:
Colors.white70),
    ),
    ),
    ...sideMenus.map(
      (menu) => SideMenuTile(
        menu: menu,
        riveonInit: (artboard) {
          StateMachineController
controller =

RiveUtils.getRiveController(artboard,
                              stateMachineName:
menu.stateMachineName);
        menu.input =
controller.findSMI("active") as SMIBool;
        },
        press: () {
          menu.input!.change(true);
Future.delayed(Duration(seconds: 1), () {
          menu.input!.change(false);
        });
        setState(() {
          selectMenu = menu;
        });
        },
        isActive: selectMenu == menu,
      ),
    ),
    Padding(
      padding: const
EdgeInsets.only(left: 24, top: 32, bottom: 16),
      child: Text(
        "User Detail".toUpperCase(),
        style: Theme.of(context)
          .textTheme
          .titleMedium!
          .copyWith(color:
Colors.white70),
      ),
    ),
    ...sideMenus2.map(
      (menu) => SideMenuTile(
        menu: menu,
        riveonInit: (artboard) {
          StateMachineController
controller =

RiveUtils.getRiveController(artboard,

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

```

stateMachineName:
menu.stateMachineName);
        menu.input =
controller.findSMI("active") as SMIBool;
    },
    press: () {
        menu.input!.change(true);

Future.delayed(Duration(seconds: 1), () {
        menu.input!.change(false);
    });
    if (menu.title == "SignOut")
    {
        Future.delayed(const
Duration(seconds: 1), () {
auth.signOut().then((value) {
Navigator.pushReplacement(
context,
PageTransition(
type:
PageTransitionType.fade,
child: const
LoginScreen(),
));
});
setState(() {
selectMenu = menu;
});
},
isActive: selectMenu == menu,
),
),
),
);
}
}

```

The side_menu.dart is the main structure of the side menu navigation bar which is appeared in dashboard and control screen.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.7.4.6.4 menu_button.dart

```
import 'package:flutter/material.dart';
import 'package:rive/rive.dart';

class MenuButton extends StatelessWidget {
  const MenuButton({
    super.key,
    required this.press,
    required this.riveOnInit,
  });

  final VoidCallback press;
  final ValueChanged<Artboard> riveOnInit;

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: GestureDetector(
        onTap: press,
        child: Row(
          mainAxisAlignment:
            MainAxisAlignment.end,
          children: [
            Container(
              height: 40,
              width: 40,
              decoration: BoxDecoration(
                color: Colors.white,
                shape: BoxShape.circle,
                boxShadow: [
                  BoxShadow(
                    color: Colors.black12,
                    offset: Offset(0, 3),
                    blurRadius: 3),
                ],
            ),
            child: RiveAnimation.asset(
              "assets/riveAssets/menu_button.riv",
              onInit: riveOnInit,
            ),
          ],
        ),
      ),
    );
  }
}
```

This menu_button.dart is the main structure of the bottom navigation bar in dashboard and control screen.

3.7.4.6.5 rive_asset.dart

```
import 'package:rive/rive.dart';

class RiveAsset {
  final String artboard, stateMachineName, title,
src;
  late SMIBool? input;

  RiveAsset(
    this.src, {
    required this.artboard,
    required this.stateMachineName,
    required this.title,
    this.input,
  });

  set setInput(SMIBool status) {
    input = status;
  }
}

List<RiveAsset> bottonNavs = [
  RiveAsset(
    "assets/riveAssets/pot.riv",
    artboard: "DASHBOARD",
    stateMachineName: "DASHBOARD Interactivity",
    title: "Dashboard",
  ),
  RiveAsset(
    "assets/riveAssets/pot.riv",
    artboard: "CONTROLLER",
    stateMachineName: "CONTROLLER Interactivity",
    title: "Controller",
  ),
];

List<RiveAsset> sideMenus = [
  RiveAsset(
    "assets/riveAssets/pot.riv",
    artboard: "HOWTOUSE",
    stateMachineName: "HOWTOUSE Interactivity",
    title: "HowToUse",
  ),
];

List<RiveAsset> sideMenus2 = [
  RiveAsset(
    "assets/riveAssets/icons.riv",
    artboard: "SETTINGS",
    stateMachineName: "SETTINGS Interactivity",
    title: "Setting",
  ),
  RiveAsset(
    "assets/riveAssets/icons.riv",
```

This material is reserved for educational use only, not allowed for commercial use.

```

artboard: "USER",
stateMachineName: "USER_Interactivity",
title: "SignOut",
),
];

```

The rive_asset.dart is used to extract the data from rive file which contain set of icon and animated icons.

3.7.4.6.6 rive_utils.dart

```

import 'package:rive/rive.dart';

class RiveUtils {
  static StateMachineController
  getRiveController(Artboard artboard,
    {stateMachineName = "State Machine 1"}) {
    StateMachineController? controller =
    StateMachineController.fromArtboard(artboard,
    stateMachineName);
    artboard.addController(controller!);
    return controller;
  }
}

```

The rive_utils.dart is used to be the tool for supporting the animated icons.

3.7.4.6.7 userprofile.dart

```

class UserProfiles {
  String email;
  String password;

  UserProfiles({required this.email, required
  this.password});
}

```

The userprofile.dart is used to store the users data with email and password.

This material is reserved for educational use only, not allowed for commercial use.

3.7.4.6.8 user_controller.dart

```
import
'package:firebase_database/firebase_database.dart
';
import 'package:flutter/material.dart';

class UserHistoryScreen extends StatefulWidget {
  const UserHistoryScreen({super.key});

  @override
  State<UserHistoryScreen> createState() =>
  _UserHistoryScreenState();
}

class _UserHistoryScreenState extends
State<UserHistoryScreen> {
  final _database =
  FirebaseDatabase.instance.ref();

  bool _controlFan = false;
  bool _controlLight = false;
  bool _controlWater = false;
  bool _controlHeater = false;
  bool _controlCooler = false;
  bool _controlAuto = false;

  double heating = 0.0;
  double fanspeed = 0.0;

  @override
  void initState() {
    super.initState();
    _activateListeners_OP_F();
    _activateListeners_OP_L();
    _activateListeners_OP_W();
    _activateListeners_OP_HE();
    _activateListeners_OP_CO();
    _activateListeners_OP_AUTO();
  }

  void _activateListeners_OP_F() {
    _database
      .child('Control/output1_fan')
      .onValue
      .listen((DatabaseEvent event) {
        bool? data_OP_F = event.snapshot.value as
        bool;
        setState(() {
          _controlFan = data_OP_F;
        });
      });
  }
}
```

This material is reserved for educational use only, not allowed for commercial use.

```

}

void _activateListeners_OP_L() {
    _database
        .child('Control/output2_light')
        .onValue
        .listen((DatabaseEvent event) {
            bool? data_OP_L = event.snapshot.value as
bool;
            setState(() {
                _controlLight = data_OP_L;
            });
        });
}

void _activateListeners_OP_W() {
    _database
        .child('Control/output3_water')
        .onValue
        .listen((DatabaseEvent event) {
            bool? data_OP_W = event.snapshot.value as
bool;
            setState(() {
                _controlWater = data_OP_W;
            });
        });
}

void _activateListeners_OP_HE() {
    _database
        .child('Control/output4_heater')
        .onValue
        .listen((DatabaseEvent event) {
            bool? data_OP_HE = event.snapshot.value as
bool;
            setState(() {
                _controlHeater = data_OP_HE;
            });
        });
}

void _activateListeners_OP_CO() {
    _database
        .child('Control/output5_cooler')
        .onValue
        .listen((DatabaseEvent event) {
            bool? data_OP_CO = event.snapshot.value as
bool;
            setState(() {
                _controlCooler = data_OP_CO;
            });
        });
}
}

```

This material is reserved for educational use only, not allowed for commercial use.

```

void _activateListeners_OP_AUTO() {
  _database
    .child('Control/output6_auto')
    .onValue
    .listen((DatabaseEvent event) {
      bool? data_OP_AUTO = event.snapshot.value
as bool;
      setState(() {
        _controlAuto = data_OP_AUTO;
      });
    });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SafeArea(
      bottom: false,
      child: Container(
        margin: const EdgeInsets.only(top: 18,
left: 24, right: 24),
        child: Column(
          children: [
            Row(
              mainAxisAlignment:
MainAxisAlignment.start,
              children: [
                Text(
                  "CONTROLLER".toUpperCase(),
                  style: TextStyle(
                    foreground: Paint()
                      ..shader =
LinearGradient(colors: <Color>[
                      Color.fromARGB(255, 0,
73, 182),
                      Color.fromARGB(255, 0,
155, 182),
                    ]).createShader(Rect.from
LTWH(0.0, 0.0, 200.0, 100.0)),
                    fontSize: 30,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
              ],
            ),
            Expanded(
              child: ListView(
                children: [
                  const SizedBox(height: 32),
                  Column(
                    children: [

```

This material is reserved for educational use only, not allowed for commercial use.

```

        Container(
          padding: const
EdgeInsets.symmetric(vertical: 10),
          decoration:
BoxDecoration(
              color: Colors.white,
borderRadius:
BorderRadius.circular(8),
              boxShadow: [
                BoxShadow(
                  color:
Colors.black12,
                  offset: Offset(0,
3),
                  blurRadius: 3),
              ],
        ),
        child: Column(
          children: [
            Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                Padding(
                  padding: const
EdgeInsets.symmetric(
24),
                  child: Text(
                    'MANUAL
BUTTONS',
                    style:
TextStyle(
                      color:
Colors.black,
                      fontSize:
15,
                      fontWeight: FontWeight.bold),
                ),
              ],
            ),
            SizedBox(
              child: Padding(
                padding: const
EdgeInsets.all(16.0),
                child: Row(
                  mainAxisAlignment:
MainAxisAlignment.spaceBetween,
                  children: [

```

This material is reserved for educational use only, not allowed for commercial use.


```

borderRadius:
BorderRadius.circular(8),
boxShadow: [
  BoxShadow(
    color:
Colors.black12,
    offset: Offset(0, 3),
    blurRadius: 3),
  ],
),
child: Column(
  crossAxisAlignment:
CrossAxisAlignment.start,
  children: [
    Padding(
      padding: const
EdgeInsets.symmetric(horizontal: 24),
      child: Text(
        'TEMPERATURE
CONTROLLER (AUTO)',
        style: TextStyle(
          color:
Colors.black,
          fontSize: 15,
          fontWeight:
FontWeight.bold),
      ),
    Slider(
      value: heating,
      onChanged: (newHeating) =>
        setState(() =>
          heating = newHeating);
    ),
    max: 50,
    activeColor:
Colors.grey,
    inactiveColor:
Color.fromARGB(96, 158, 158, 158),
    thumbColor:
Colors.grey,
  ),
  Padding(
    padding: const
EdgeInsets.symmetric(horizontal: 24),
    child: Row(
      mainAxisAlignment:
MainAxisAlignment.spaceBetween,
      children: [
        Text('0\u00B0'),
        Text('10\u00B0'),
        Text('20\u00B0'),

```

This material is reserved for educational use only, not allowed for commercial use.

```

Text('30\u00B0'),
Text('40\u00B0'),
Text('50\u00B0'),
    ],
    ),
    ),
    ],
    ),
    ),
const SizedBox(height: 16),
Container(
  padding: const
EdgeInsets.symmetric(vertical: 10),
  decoration: BoxDecoration(
    color: Colors.white,
    borderRadius:
BorderRadius.circular(8),
    boxShadow: [
      BoxShadow(
        color:
Colors.black12,
        offset: Offset(0, 3),
        blurRadius: 3),
    ],
  ),
  child: Column(
    crossAxisAlignment:
CrossAxisAlignment.start,
    children: [
      Padding(
        padding: const
EdgeInsets.symmetric(horizontal: 24),
        child: Text(
          'FAN SPEED',
          style: TextStyle(
            color:
Colors.black,
            fontSize: 15,
            fontWeight:
FontWeight.bold),
        ),
      ),
      Slider(
        value: fanspeed,
        onChanged:
(newfanspeed) {
          setState(() =>
fanspeed = newfanspeed);
        },
        max: 50,
        activeColor:
Colors.grey,

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.


```

    }
  }
} catch (e) {}
},
child: Container(
  width: 80,
  height: 80,
  padding: const
EdgeInsets.symmetric(vertical: 6, horizontal: 6),
  decoration: BoxDecoration(
    color: logic ? Colors.blue :
Colors.transparent,
    borderRadius:
BorderRadius.circular(40),
    border: Border.all(color:
Colors.blue),
  ),
  child: Column(
    mainAxisAlignment:
MainAxisAlignment.center,
    children: [
      Text(
        logic ? 'ON' : 'OFF',
        style: TextStyle(
          color: logic ? Colors.white :
Colors.blue,
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),
      ),
      const SizedBox(height: 28),
      Center(
        child: Text(
          title,
          style: TextStyle(
            color: Colors.black54, fontSize:
12, fontWeight: FontWeight.bold),
        ),
      ),
    ],
  ));
}
}

```

The `user_controller.dart` is the main structure of the control screen which is included the buttons for controlling the actuators.

This material is reserved for educational use only, not allowed for commercial use.

3.7.4.6.9 user_dashboard.dart

```
import
'package:firebase_database/firebase_database.dart
';
import 'package:flutter/material.dart';

class GraphDashboard extends StatefulWidget {
  const GraphDashboard({super.key});

  @override
  State<GraphDashboard> createState() =>
  _GraphDashboardState();
}

class _GraphDashboardState extends
State<GraphDashboard> {
  final _database =
  FirebaseDatabase.instance.ref();

  double _humidityValue = 0.0;
  double _lightValue = 0.0;
  double _soilValue = 0.0;
  double _tempValue = 0.0;
  double _oxygenValue = 0.0;
  double _waterInTank = 0;

  @override
  void initState() {
    _activateListeners_H();
    _activateListeners_L();
    _activateListeners_S();
    _activateListeners_T();
    _activateListeners_WT();
    super.initState();
  }

  void _activateListeners_H() {
    _database.child('Sensor/Hum/Humshow').onValue.listen((DatabaseEvent event) {
      int data_H = event.snapshot.value as int;
      setState(() {
        _humidityValue = (data_H / 100) * 100;
      });
    });
  }

  void _activateListeners_L() {
    _database
      .child('Sensor/Light/Lightshow')
      .onValue
```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content 75 and cite the document when use.

```

        .listen((DatabaseEvent event) {
            int data_L = event.snapshot.value as int;
            setState(() {
                _lightValue = (data_L / 4000) * 100;
            });
        });
    }

    void _activateListeners_S() {
        _database
            .child('Sensor/Soil/Soilshow')
            .onValue
            .listen((DatabaseEvent event) {
                int data_S = event.snapshot.value as int;
                setState(() {
                    _soilValue = (data_S / 2500) * 100;
                });
            });
    }

    void _activateListeners_T() {
        _database
            .child('Sensor/Temp/Tempshow')
            .onValue
            .listen((DatabaseEvent event) {
                if (event.snapshot.value.runtimeType ==
                    double) {
                    double? data_T = event.snapshot.value as
                    double;
                    setState(() {
                        _tempValue = data_T;
                    });
                } else {
                    int? data_T = event.snapshot.value as
                    int;
                    setState(() {
                        _tempValue = data_T.toDouble();
                    });
                }
            });
    }

    void _activateListeners_WT() {
        _database.child('Sensor/Water').onValue.listen((D
            atabaseEvent event) {
                if (event.snapshot.value.runtimeType ==
                    double) {
                    double? data_WT = event.snapshot.value as
                    double;
                    setState(() {
                        _waterInTank = data_WT;
                    });
                }
            });
    }

```

This material is reserved for educational use only, not allowed for commercial use.

```

    } else {
        int? data_WT = event.snapshot.value as
int;
        setState(() {
            _waterInTank = data_WT.toDouble();
        });
    }
});
}

@override
Widget build(BuildContext context) {
    final double sliderWidth =
MediaQuery.of(context).size.width * 0.8;
    return Scaffold(
        body: SafeArea(
            bottom: false,
            child: Container(
                margin: const EdgeInsets.only(top: 18,
left: 24, right: 24),
                child: Column(
                    children: [
                        Row(
                            mainAxisAlignment:
MainAxisAlignment.start,
                            children: [
                                Text(
                                    "DASHBOARD".toUpperCase(),
                                    style: TextStyle(
                                        foreground: Paint()
                                        ..shader =
LinearGradient(colors: <Color>[
                                        Color.fromARGB(255, 0,
73, 182),
                                        Color.fromARGB(255, 0,
155, 182),
                                    ]).createShader(Rect.from
LTWH(0.0, 0.0, 200.0, 100.0)),
                                    fontSize: 30,
                                    fontWeight:
FontWeight.bold,
                                ),
                            ),
                        ),
                    ],
                ),
                ),
            ),
            const SizedBox(height: 14),
            Expanded(
                child: ListView(
                    children: [
                        _MainDashboard(
                            icons1:
Icons.thermostat_sharp,
                            name1: 'TEMPERATURE',

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

```

        value1: _tempValue,
        icons2:
Icons.water_drop_rounded,
        name2: 'WATER CAPACITY',
        value2: _waterInTank,
        width: sliderWidth,
    ),
    _PercentDashboard(
        icons: Icons.sunny,
        iconcolor:
Colors.orangeAccent,
        name: 'LIGHT INTENSITY',
        value: _lightValue,
        width: sliderWidth,
        grecolor: LinearGradient(
            begin:
Alignment.topRight,
            end:
Alignment.bottomLeft,
            colors: <Color>[
                Color.fromARGB(255,
230, 214, 0),
                Colors.orangeAccent,
            ]),
        _PercentDashboard(
            icons:
Icons.water_drop_rounded,
            iconcolor: Colors.blue,
            name: 'AIR HUMIDITY',
            value: _humidityValue,
            width: sliderWidth,
            grecolor: LinearGradient(
                begin:
Alignment.topRight,
                end:
Alignment.bottomLeft,
                colors: <Color>[
                    Colors.blue,
                    Color.fromARGB(255,
0, 73, 182),
                ]),
            _PercentDashboard(
                icons:
Icons.waves_rounded,
                iconcolor: Colors.blue,
                name: 'SOIL HUMIDITY',
                value: _soilValue,
                width: sliderWidth,
                grecolor: LinearGradient(
                    begin:
Alignment.topRight,

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.


```

borderRadius:
BorderRadius.circular(15),
boxShadow: [
  BoxShadow(
    color: Colors.black12, offset:
Offset(0, 3), blurRadius: 3),
  ],
),
child: Row(
  mainAxisAlignment:
MainAxisAlignment.spaceAround,
  children: [
    SizedBox(
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.center,
        children: [
          Row(
            children: [
              Text(
                name1,
                style: TextStyle(
                  color:
Colors.black,
                  fontSize: 15,
                  fontWeight:
FontWeight.bold),
            ),
            SizedBox(
              child: Row(
                children: [
                  SizedBox(
                    child: Icon(
                      icons1,
                      color: value1 <= 25
? Colors.blue
: value1 >= 35
?
Colors.red
:
Colors.orangeAccent,
                    size: 30.0,
                  ),
                  width: 35,
                  height: 35,
                ),
                SizedBox(
                  child: Text(
                    value1.toStringAsFixed(2) + '\u00B0',
                    style: TextStyle(

```

This material is reserved for educational use only, not allowed for commercial use.


```

children: [
  Text (
    name,
    style: TextStyle(
      color: Colors.black,
      fontSize: 15,
      fontWeight:
FontWeight.bold),
  ),
  Text (
    value.toStringAsFixed(2)
+ '%',
    style: TextStyle(
      color: Colors.black,
      fontSize: 15,
      fontWeight:
FontWeight.bold),
  ),
],
),
Row(
  children: [
    SizedBox(
      child: Icon(
        icons,
        color: iconcolor,
        size: 30.0,
      ),
      width: 35,
      height: 35,
    ),
    Expanded(
      child: Column(
        children: [
          SizedBox(
            child: Column(
              mainAxisAlignment:
MainAxisAlignment.min,
              children: [
                Padding(
                  padding:
const EdgeInsets.symmetric(
                    vertical:
8.0),
                  child: Stack(
                    children: [
                      Container(
                        width:
width,
                        height:
height:
10,

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content **83** and cite the document when use.


```

@override
State<EntryScreen> createState() =>
_EntryScreenState();
}

class _EntryScreenState extends
State<EntryScreen>
with SingleTickerProviderStateMixin {
final formkey = GlobalKey<FormState>();
final auth = FirebaseAuth.instance;

RiveAsset selectedBottonNav = bottonNavs.first;
late AnimationController _animationController;
late Animation<double> animation;
late Animation<double> scaleAnimation;
late SMIBool isSideBarClosed;
bool isSideMenuClosed = true;

int _selectedIndex = 0;
void _onItemTapped(int index) {
setState(() {
_selectedIndex = index;
});
}

final List<Widget> _pages = [
const GraphDashboard(),
const UserHistoryScreen(),
];

@override
void initState() {
_animationController = AnimationController(
vsync: this,
duration: Duration(milliseconds: 200),
)..addListener(() {
setState(() {});
});
animation = Tween<double>(begin: 0, end:
1).animate(
CurvedAnimation(
parent: _animationController,
curve: Curves.fastOutSlowIn,
),
);
scaleAnimation = Tween<double>(begin: 1, end:
0.8).animate(
CurvedAnimation(
parent: _animationController,
curve: Curves.fastOutSlowIn,
),
);
}
}

```

This material is reserved for educational use only, not allowed for commercial use.

```

FirebaseMessaging.instance.getInitialMessage().then((message) {
  if (message != null) {}
});

FirebaseMessaging.onMessage.listen((message)
{
  if (message.notification != null) {}
});

FirebaseMessaging.onMessageOpenedApp.listen((message) {
  if (message.notification != null) {}
});

super.initState();
}

@override
void dispose() {
  _animationController.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    resizeToAvoidBottomInset: false,
    extendBody: true,
    backgroundColor: Color(0xff1d1e26),
    body: Stack(
      children: [
        AnimatedPositioned(
          duration: Duration(milliseconds:
200),
          curve: Curves.fastOutSlowIn,
          width: 288,
          left: isSideMenuClosed ? -288 : 0,
          height:
MediaQuery.of(context).size.height,
          child: SideMenu(),
        ),
        Transform(
          alignment: Alignment.center,
          transform: Matrix4.identity()
            ..setEntry(3, 2, 0.001)
            ..rotateY(animation.value - 30 *
animation.value * pi / 180),
          child: Transform.translate(
            offset: Offset(animation.value *
265, 0),

```

This material is reserved for educational use only, not allowed for commercial use.

```

        child: Transform.scale(
          scale: scaleAnimation.value,
          child: ClipRRect(
            borderRadius: BorderRadius.all(
              Radius.circular(!isSideMenuClosed ? 24 : 0)),
            child: _pages[_selectedIndex],
          ),
        ),
      ),
    ),
  ),
  AnimatedPositioned(
    duration: Duration(milliseconds:
200),
    curve: Curves.fastOutSlowIn,
    top: 16,
    left: isSideMenuClosed ? 338 : 240,
    child: MenuButton(
      riveOnInit: (artboard) {
        StateMachineController controller
= RiveUtils.getRiveController(
          artboard,
          stateMachineName: "State
Machine");
        isSideBarClosed =
controller.findSMT("isOpen") as SMIBool;
        isSideBarClosed.value = true;
      },
      press: () {
        isSideBarClosed.value
= !isSideBarClosed.value;
        if (isSideMenuClosed) {
          animationController.forward();
        } else {
          animationController.reverse();
        }
      },
      setState(() {
        isSideMenuClosed =
isSideBarClosed.value;
      }));
    ),
  ),
),
],
),
),
bottomNavigationBar: Transform.translate(
  offset: Offset(0, 200 * animation.value),
  child: SafeArea(
    child: Container(
      padding: const EdgeInsets.all(12.0),

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content **88** and cite the document when use.

```

        margin: const
EdgeInsets.symmetric(horizontal: 45.0, vertical:
15),
        decoration: const BoxDecoration(
            color: Color.fromARGB(192, 29,
30, 38),
            borderRadius:
BorderRadius.all(Radius.circular(24))),
        child: Row(
            mainAxisAlignment:
MainAxisAlignment.spaceAround,
            children: [
                ...List.generate(
                    bottonNavs.length,
                    (index) => GestureDetector(
                        onTap: () {
                            _onItemTapped(index);
                            bottonNavs[index].input!.change(true);
                            if (bottonNavs[index] !=
selectedBottonNav) {
                                setState(() {
                                    selectedBottonNav =
bottonNavs[index];
                                });
                                Future.delayed(const
Duration(seconds: 1), () {
                                    bottonNavs[index].input!.change(false);
                                });
                            },
                            child: Column(
                                mainAxisAlignment:
MainAxisSize.min,
                                children: [
                                    AnimatedBar(
                                        isActive:
bottonNavs[index] == selectedBottonNav),
                                    SizedBox(
                                        height: 44,
                                        width: 44,
                                        child: Opacity(
                                            opacity:
bottonNavs[index] == selectedBottonNav
? 1
: 0.5,
                                            child:
RiveAnimation.asset(
bottonNavs.first.src,

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content **89** and cite the document when use.

```
artboard:
  bottonNavs[index].artboard,
  onInit: (artboard)
{
  StateMachineController controller =
  RiveUtils.getRiveController(artboard,
  stateMachineName:
  bottonNavs[index].stateMachineName);
  bottonNavs[index].input =
  controller.findSMI("active") as SMIBool;
```

The `entry_point.dart` is the main component which is used with the bottom navigation bar to show the dashboard and control screen.

3.7.4.6.11 main.dart

```
import
'package:firebase_core/firebase_core.dart';
import
'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import
'package:projectiot/screen/LoginSession/login.dart';

Future<void> backgroundHandler(RemoteMessage
message) async {
  print(message.data.toString());
  print(message.notification!.title);
}

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  FirebaseMessaging.onBackgroundMessage(backgroundH
andler);
  await
  FirebaseMessaging.instance.setForegroundNotificat
ionPresentationOptions(
    alert: true,
    badge: true,
    sound: true,
  );
  await
  FirebaseMessaging.instance.requestPermission(
    alert: true,
    announcement: false,
    badge: true,
    carPlay: false,
    criticalAlert: false,
    provisional: false,
    sound: true,
  );
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    SystemChrome.setPreferredOrientations([
```

This material is reserved for educational use only, not allowed for commercial use.

```
DeviceOrientation.portraitUp,  
DeviceOrientation.portraitDown,  
]);  
return MaterialApp(  
  title: 'DeComplex',  
  theme: ThemeData(  
    primarySwatch: Colors.blue,  
  ),  
  home: const LoginScreen(),  
);  
}  
}
```

The main.dart is the significant part which is needed to be run firstly before the other script.



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

3.7.4.6.12 pubspec.yaml

```
name: projectiot
description: A new Flutter project.

# The following line prevents the package from
# being accidentally published to
# pub.dev using `flutter pub publish`. This is
# preferred for private packages.
publish_to: 'none' # Remove this line if you wish
# to publish to pub.dev

# The following defines the version and build
# number for your application.
# A version number is three numbers separated by
# dots, like 1.2.43
# followed by an optional build number separated
# by a +.
# Both the version and the builder number may be
# overridden in flutter
# build by specifying --build-name and --build-
# number, respectively.
# In Android, build-name is used as versionName
# while build-number used as versionCode.
# Read more about Android versioning at
# https://developer.android.com/studio/publish/vers
# ioning
# In iOS, build-name is used as
# CFBundleShortVersionString while build-number is
# used as CFBundleVersion.
# Read more about iOS versioning at
#
# https://developer.apple.com/library/archive/docum
# entation/General/Reference/InfoPlistKeyReference/
# Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major,
# minor, and patch parts
# of the product and file versions while build-
# number is used as the build suffix.
version: 1.0.0+1

environment:
  sdk: '>=2.19.0-264.0.dev <3.0.0'

# Dependencies specify other packages that your
# package needs in order to work.
# To automatically upgrade your package
# dependencies to the latest versions
# consider running `flutter pub upgrade --major-
# versions`. Alternatively,
# dependencies can be manually updated by
# changing the version numbers below to
```

This material is reserved for educational use only, not allowed for commercial use.

```

# the latest version available on pub.dev. To see
which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter

# The following adds the Cupertino Icons font
to your application.
# Use with the CupertinoIcons class for iOS
style icons.
cupertino_icons: ^1.0.2
rive: ^0.10.2
form_field_validator: ^1.1.0
shared_preferences: ^2.0.17
page_transition: ^2.0.9
firebase_core: ^2.4.1
firebase_auth: ^4.2.5
flutter_launcher_icons: ^0.11.0
firebase_database: ^10.1.1
firebase_messaging: ^14.6.0
flutter_local_notifications: ^14.0.0+2

dev_dependencies:
  flutter_test:
    sdk: flutter

flutter_icons:
  android: "launcher_icon"
  ios: true
  image_path: "assets/images/DeComplex_logo.png"
  min_sdk_android: 21
  web:
    generate: true
    image_path:
      "assets/images/DeComplex_logo.png"
    background_color: "#hexcode"
    theme_color: "#hexcode"
  windows:
    generate: true
    image_path:
      "assets/images/DeComplex_logo.png"
    icon_size: 48
  macos:
    generate: true
    image_path:
      "assets/images/DeComplex_logo.png"

# The "flutter_lints" package below contains a
set of recommended lints to

```

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use. 94

```
# encourage good coding practices. The lint set
provided by the package is
# activated in the `analysis_options.yaml` file
located at the root of your
# package. See that file for information about
deactivating specific lint
# rules and activating additional ones.
flutter_lints: ^2.0.0

# For information on the generic Dart part of
this file, see the
# following page:
https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter
packages.
flutter:

# The following line ensures that the Material
Icons font is
# included with your application, so that you
can use the icons in
# the material Icons class.
uses-material-design: true

assets:
  - assets/images/
  - assets/models/testpot/
  - assets/models/testpot/testpot.obj
  - assets/models/testpot/pot.glb
  - assets/riveAssets/
  - assets/riveAssets/icons.riv
  - assets/riveAssets/pot.riv
```

The pubspec.yaml is used to contain the necessary dependencies which are included in this mobile application.

CHAPTER 4

EVALUATION AND RESULT

4.1 Evaluation Criteria

From the possible criteria to measure plant growth introduced in Chapter 3 of the report, Plant height is chosen as the criteria for the measurement. With reason for not choosing the other four as criteria as follows

Biomass	Having to account for extra moisture can lead to error. And the team speculated that testing time are too short for any significant difference in biomass to be observed.
Leaf area	The species of the plant used in testing have many small leaves making counting them extremely difficult.
Root development	Require digging out the plant from the pot to measure the root. Can cause serious error.
Flowering and fruiting	The plant species does not produce either.

Table 4.1.1 Reason for not choosing the criteria

4.2 Testing Method

There will be five plants in the test. Each growing under different conditions as follows.

Auto	The plant will be placed inside the project workpiece and will be monitored and taken care of by automation.
Human Control	Human will take care of all three major conditions of the plant. (Water, Light, Temperature)
No light Control	Similar to control but without extra light control.
No Water Control	No water is provided to the plant.
No Temperature Control	No temperature control is provided.

Table 4.2.1 Explanation of each control environment

The first two conditions are tested to see the effectiveness of the project compared to human caretaker. The other three conditions are to see the impact each variable can have on the growth of the plant.

For Auto and Human control environment. The temperature, air humidity, and soil humidity are also recorded each time height measurement is taken.

4.3 Result

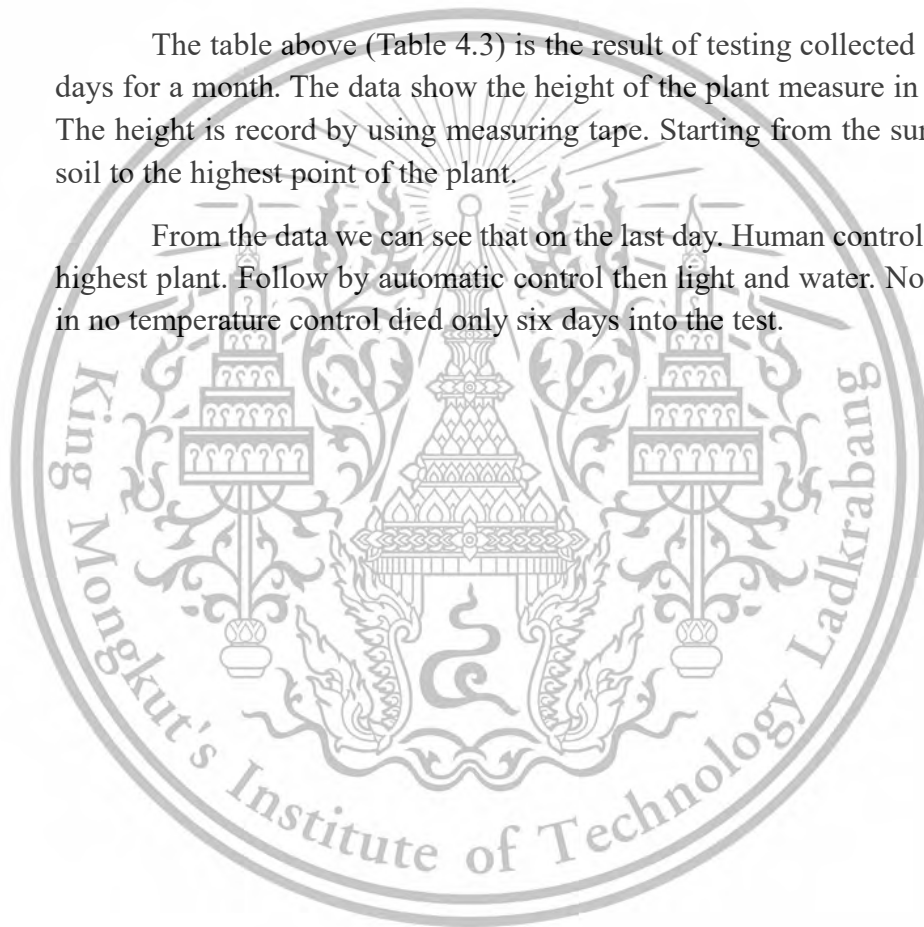
4.3.1 Height Table Data

	D0	D3	D6	D9	D12	D15	D18	D21	D24	D27	D30
Auto	100	109	111	130	153	169	189	210	238	246	273
Control	100	112	126	145	169	187	213	233	255	272	295
No light	100	105	113	128	139	152	163	180	191	201	210
No water	100	107	118	126	137	147	158	170	183	193	204
No Temp	100	103	Died	Died	Died	Died	Died	Died	Died	Died	Died

Table 4.3.1.1 Plant Height Table

The table above (Table 4.3) is the result of testing collected every three days for a month. The data show the height of the plant measure in millimeter. The height is record by using measuring tape. Starting from the surface of the soil to the highest point of the plant.

From the data we can see that on the last day. Human control pot has the highest plant. Follow by automatic control then light and water. Notably, plant in no temperature control died only six days into the test.



4.3.2 Temperature, Air and Soil Humidity Table Data

Auto	D0	D3	D6	D9	D12	D15	D18	D21	D24	D27	D30
Temperature	23.5	24	24.3	23.4	23.7	22.8	24.5	23.8	23.9	23.7	23.6
Humidity	55	60	62	57	59	52	65	57	57	58	57
Soil Humidity	680	724	702	744	720	697	730	712	685	728	702

Table 4.3.2.1 Auto environment data of Temperature, Air and Soil Humidity

Control	D0	D3	D6	D9	D12	D15	D18	D21	D24	D27	D30
Temperature	23.7	23.6	23.6	23.7	23.6	23.6	23.5	23.6	23.6	23.6	23.6
Humidity	62	61	62	61	61	60	63	61	62	61	60
Soil Humidity	653	713	680	665	728	702	687	660	718	698	675

Table 4.3.2.2 Control environment data of Temperature, Air and Soil Humidity

The two table above contain data of temperature, air humidity, and soil humidity reading at the time height measurement was taken for Auto (Table 4.4) and Control (Table 4.5)

Preliminary reading show that data from the auto environment fluctuate much more than human control environment. With temperature reading ranging from the lowest at 22.8°C to the highest at 24.5°C. A difference of 1.7°C. Compare to data collect from Human control environment with lowest at 23.5°C and highest at 23.7°C. A difference of only 0.2°C.

4.3.3 Height Graph

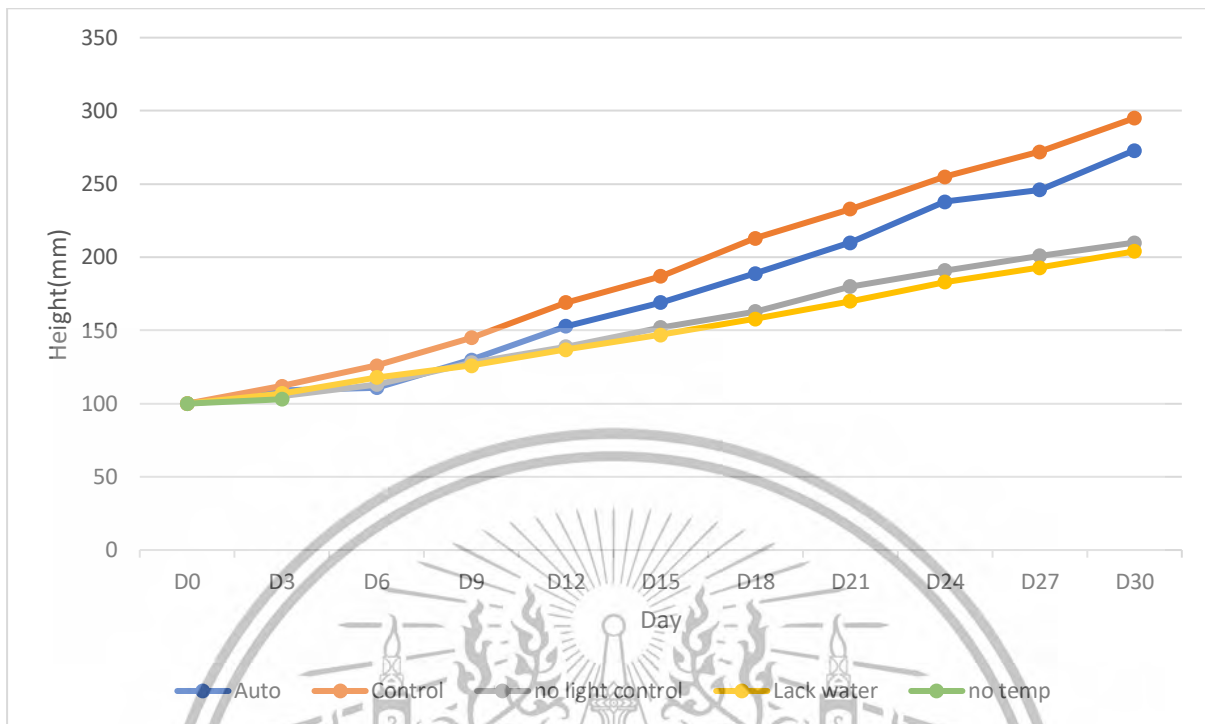


Figure 4. 1 Plant Height Graph

The figure above shows the scatter plot graph created from the data collected from the test.

From the graph. The trend is very consistent. With environment that control all three variables being ahead of all other. Other notable thing that can be observe is on the day six and nine of light and water control environment. Before day six. No water is ahead of no light control but from day nine onward. No light control overtakes no water for the rest of the test.

4.3.4 Temperature, Air and Soil Humidity Graph

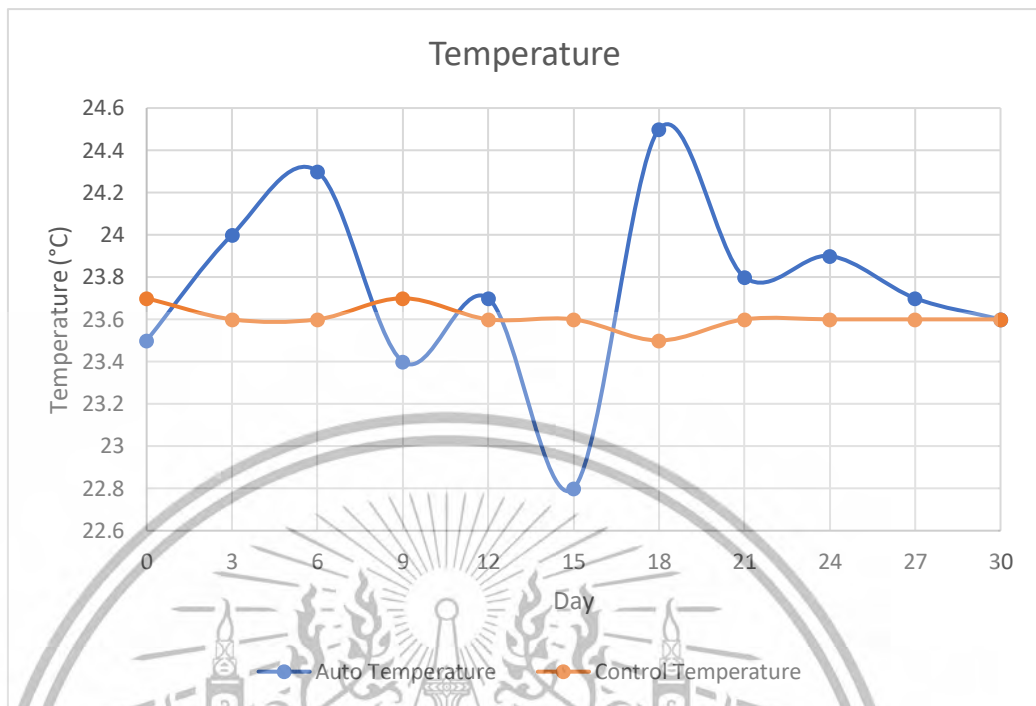


Figure 4. 2 Compare Temperature of auto and control

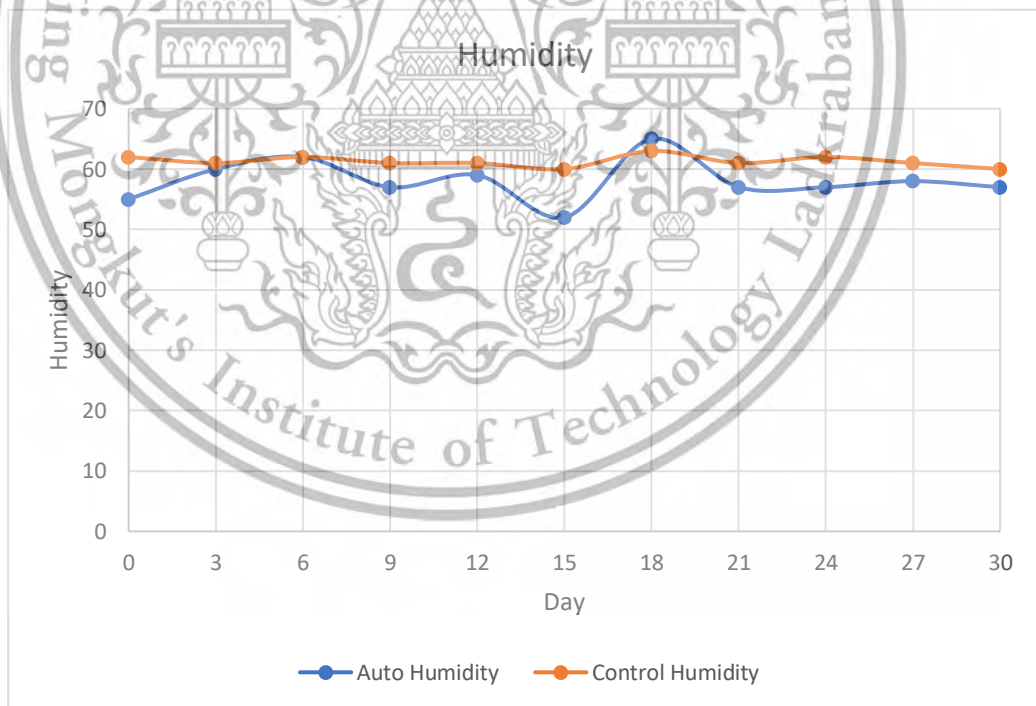


Figure 4. 3 Compare Air Humidity of auto and control

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

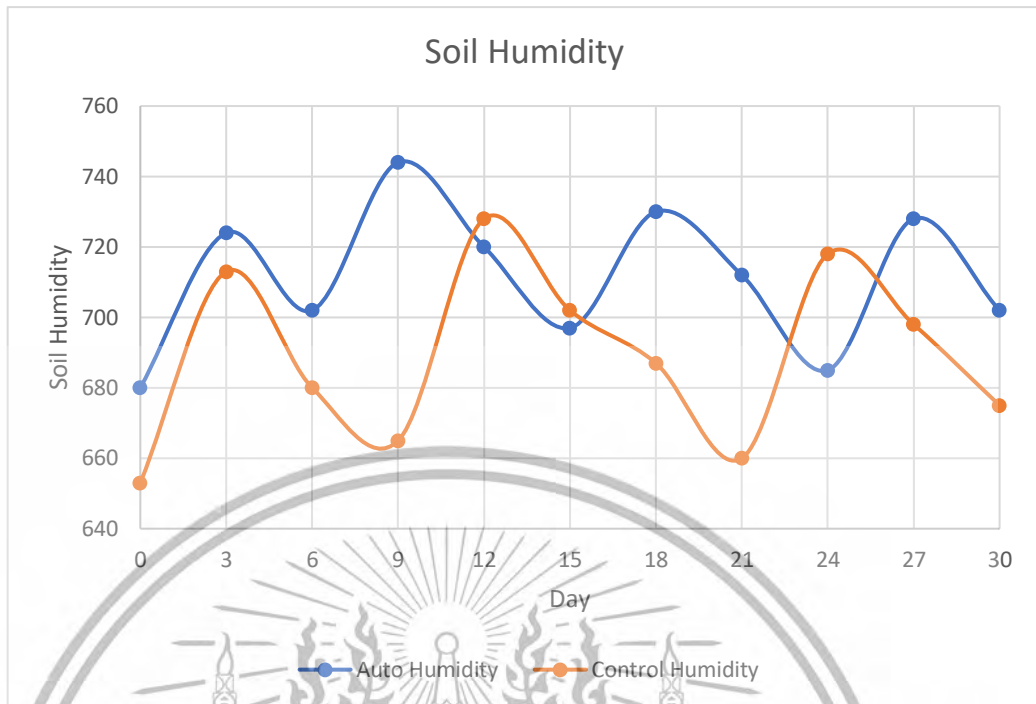


Figure 4. 4 Compare Soil Humidity of auto and control

The three figure above show graph created with data from the Temperature, Air and Soil humidity table. Each graph takes the Temperature, Humidity, and Soil humidity from both tables to compare the trend and difference between them.

The trend observes from these graph show that for both Temperature and Humidity. Control environment data are very stable while the auto show more fluctuation. The exception is found in Soil humidity where both auto and control environment fluctuate greatly each time the data is recorded.

4.4 Data Analyze

With the result from testing. An analyze is conducted to look for further information that can be gained.

4.4.1 Analyze – Plant Height

Auto	Show steady increase in plant height over the duration of the test.
Control	Also show consistent growth. On the last day of the test have plant with the highest height.
No light	Show a slower grow pattern than auto and control. At the end is slightly higher than no water.
No Water	Show a slower grow pattern than auto and control.
No Temperature	Plants are unable to grow and eventually die. Suggested that temperature plays a very important role in growth of the plant.

Figure 4. 5 Analyzing Plant Height

The overall assessment gather is that auto and control show the most growth rate of all condition in testing. With no light and no water showing slower growth. Proving that it is essential to control the environment to ensure that plant will have the highest growth rate possible.

4.4.2 Analyze – Temperature, Air and Soil Humidity

Auto	Temperature	Relatively stable. But still move around more than control.
	Humidity	Show some fluctuation when compare to control.
	Soil Humidity	Fluctuate a lot.
Control	Temperature	Very stable with minimal fluctuation.
	Humidity	Show minimal fluctuation. But is more stable compare to auto.
	Soil Humidity	Fluctuate a lot

Figure 4. 6 Analyzing Temperature, Air and Soil Humidity from auto and control environment

Comparing each factor. Both auto and control are very similar to each other in practice. Both temperature and humidity value are relatively similar with only slight variation observed. Only soil humidity of both shows high fluctuation with auto having more fluctuation than control.

Overall, there does not appear to be a significant difference in temperature, humidity, and soil humidity between auto and control environment. This suggests that the difference between both is not significant enough to impact the growth of the plant too drastically. Plant grow in both environments are likely to achieve the same result.

CHAPTER 5

CONCLUSION

In conclusion, the experiment conducted to test the hypothesis that an automatic control system for regulating water, temperature, humidity, and light for plant growth would be as effective as or more effective than traditional control methods while requiring less labor and manual intervention has provided insightful results.

The results after 30 days have confirmed the hypothesis, demonstrating that the automatic control system is indeed as effective as traditional control methods, with a marginal 5% increase in efficiency achieved by human control. Furthermore, the experiment revealed that when water, light, and temperature are not properly regulated, plant growth declined by approximately 40-45% in height.

5.1 Learning Point

5.1.1 IoT Protocol

The implementation of advanced IoT technologies, such as ESP32, Firebase, Node-RED, and Flutter, proved instrumental in the successful development of the automatic control system. The integration of these protocols enabled seamless data communication, real-time monitoring, and remote-control capabilities. This underscores the significance of leveraging IoT protocols in future research and applications related to plant growth. By harnessing the power of interconnected devices and data analytics, researchers can enhance the efficiency, accuracy, and scalability of automated control systems.

5.1.2 Cloud Control System

The incorporation of cloud-based technologies, exemplified by Node-RED and Flutter, demonstrated the immense potential of leveraging cloud computing in plant growth applications. The utilization of these platforms enabled centralized data storage, analysis, and control, fostering greater accessibility and flexibility. Cloud-based control systems offer the advantage of remote management and scalability, allowing researchers to monitor and adjust plant growth parameters from anywhere. The findings from this experiment encourage further exploration and development of cloud-based control systems, which can revolutionize agricultural practices and enable efficient resource utilization on a larger scale.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

5.1.3 Control Room

The creation of a controlled environment, characterized by a dimension of 50×50×60 cm, served as a dedicated space for conducting the experiment. However, the versatility and adaptability of the control room setup extend beyond plant growth experiments. The controlled environment can be repurposed for various applications, such as studying the effects of environmental factors on other organisms or simulating specific conditions for scientific research. The modular design and precise control parameters make the control room an invaluable resource for conducting controlled experiments and fostering innovative investigations in diverse scientific disciplines.



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

5.2 Future Development

For future development, several areas can be explored to enhance the automatic control system

5.2.1 Improved Sensor Technology

The accuracy and range of data acquired by the automatic control system can be significantly improved through the improvement in sensor technology. Higher accuracy and sensitivity sensors can be integrated to offer more precise data on environmental factors such as light intensity, air quality, and nutrient levels. Incorporating sensors to gauge variables like pH levels and soil conductivity can also provide important insights into the unique nutrient requirements and growth circumstances of various plant species.

5.2.2 Advanced Actuator Systems

The effectiveness and versatility of the automatic control system can be improved by upgrading the actuator systems. Advanced irrigation systems with adjustable flow rates, variable-speed fans for precise air circulation control, and smart humidifiers that maintain ideal humidity levels are a few examples of advanced actuators that can be used to create growth conditions for plants that are more targeted and responsive. Real-time alterations and fine-tuning of environmental factors in accordance with plant requirements can be made possible by integrating feedback loops between sensors and actuators.

5.2.3 Extended Experiment Duration

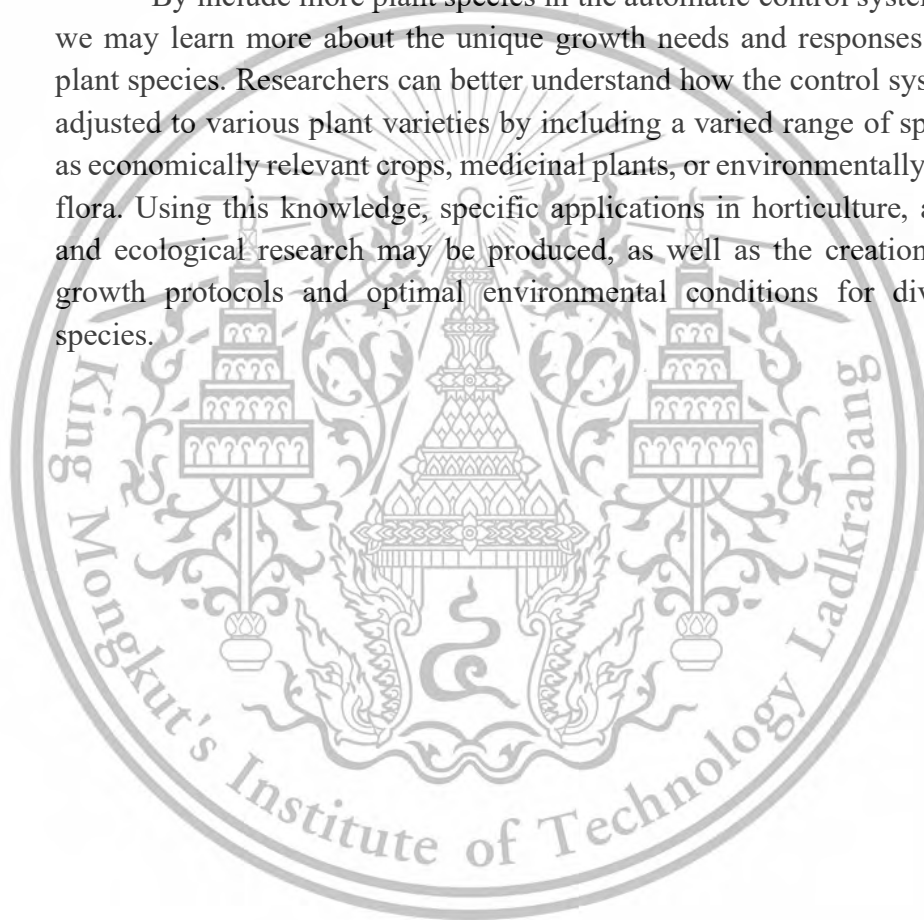
Extending the duration of experiments beyond 30 days allows for long-term monitoring of plant growth patterns, responses, and adaptations to the automatic control system. A longer experiment duration can provide insights into the system's stability, the plant's growth trajectory over time, and potential seasonal variations in growth patterns. Additionally, it allows for the observation of complex phenomena such as flowering, fruiting, and seed production, providing a more comprehensive understanding of plant life cycles within the controlled environment.

5.2.4 Iterative System Refinement

Conducting the experiment several times can aid in detecting potential system limitations improving control as well as lowering uncertainties or errors. Researchers can improve the automatic control system's overall performance, robustness, and reliability by continuously enhancing it. With each iteration, the insights gained from earlier tests can be expanded upon, resulting in a more optimized and effective control system design.

5.2.5 Plant Diversity and Specialty

By include more plant species in the automatic control system research, we may learn more about the unique growth needs and responses of various plant species. Researchers can better understand how the control system can be adjusted to various plant varieties by including a varied range of species, such as economically relevant crops, medicinal plants, or environmentally significant flora. Using this knowledge, specific applications in horticulture, agriculture, and ecological research may be produced, as well as the creation of unique growth protocols and optimal environmental conditions for diverse plant species.



References

- 5050 LED Datasheet: The Ultimate Guide - SIRS Electronics, Inc. (n.d.). *Understanding 5050 LED Strip Datasheets 5050 LED Datasheet: The Ultimate Guide*. Retrieved from SIRS-E: <https://sirs-e.com/general/5050-led-datasheet/>
- Agriculture Technology. (n.d.). *Agriculture Technology*. Retrieved from National Institute of Food and Agriculture: <https://www.nifa.usda.gov/topics/agriculture-technology>
- codeness.ly. (2021, March 16). *form_field_validator*. Retrieved from pub.dev: https://pub.dev/packages/form_field_validator
- dexterx.dev. (2023, May 11). *flutter_local_notifications*. Retrieved from pub.dev: https://pub.dev/packages/flutter_local_notifications
- esp32_datasheet_en.pdf. (2023, 01). *ESP32 Series Datasheet*. Retrieved from espressif: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- FC_FD_970528. (n.d.). *MINI FLOAT LEVEL SWITCH*. Retrieved from FineTek: [http://www.interflow-th.com/information_Product\(more\)/FINETEK-F\(C\)D_MINI_FLOAT_LEVEL_SWITCH.pdf](http://www.interflow-th.com/information_Product(more)/FINETEK-F(C)D_MINI_FLOAT_LEVEL_SWITCH.pdf)
- firebase.google.com. (2023, May 16). *firebase_auth*. Retrieved from pub.dev: https://pub.dev/packages/firebase_auth
- firebase.google.com. (2023, May 16). *firebase_core*. Retrieved from pub.dev: https://pub.dev/packages/firebase_core
- firebase.google.com. (2023, May 16). *firebase_database*. Retrieved from pub.dev: https://pub.dev/packages/firebase_database
- firebase.google.com. (2023, May 16). *firebase_messaging*. Retrieved from pub.dev: https://pub.dev/packages/firebase_messaging
- flutter.dev. (2023, May 8). *Shared preferences plugin*. Retrieved from pub.dev: https://pub.dev/packages/shared_preferences
- fluttercommunity.dev. (2023, April 19). *flutter_launcher_icons*. Retrieved from pub.dev: https://pub.dev/packages/flutter_launcher_icons
- Ks0034 keyestudio DHT11 Temperature and Humidity Sensor - Keyestudio Wiki. (2021, May 26). *Ks0034 keyestudio DHT11 Temperature and Humidity Sensor*. Retrieved from keyestudio: https://wiki.keyestudio.com/Ks0034_keyestudio_DHT11_Temperature_and_Humidity_Sensor
- Ks0049 keyestudio Soil Humidity Sensor - Keyestudio Wiki. (2021, January 7). *Ks0049 keyestudio Soil Humidity Sensor*. Retrieved from keyestudio: https://wiki.keyestudio.com/Ks0049_keyestudio_Soil_Humidity_Sensor

This material is reserved for educational use only, not allowed for commercial use.

Ks0098 keystudio TEMT6000 Ambient Light Sensor - Keystudio Wiki. (2021, January 7). *Ks0098 keystudio TEMT6000 Ambient Light Sensor*. Retrieved from keystudio: https://wiki.keystudio.com/Ks0098_keystudio_TEMT6000_Ambient_Light_Sensor

node-red-dashboard (node) - Node-RED. (2023, March 11). *node-red-dashboard*. Retrieved from Node-RED: <https://flows.nodered.org/node/node-red-dashboard>

raspberrypi-4-datasheet.pdf. (2019, June). *DATASHEET Raspberry Pi 4 Model B*. Retrieved from raspberrypi: <https://datasheets.raspberrypi.com/rpi4/raspberrypi-4-datasheet.pdf>

rive.app. (2023, April 13). *Rive Flutter*. Retrieved from pub.dev: <https://pub.dev/packages/rive>

Security Rules and Firebase Authentication | Firebase Security Rules. (2023, May 4). *Security Rules and Firebase Authentication*. Retrieved from Firebase: <https://firebase.google.com/docs/rules/rules-and-auth>

The Development of Agriculture. (n.d.). *The Development of Agriculture*. Retrieved from NATIONAL GEOGRAPHIC: <https://education.nationalgeographic.org/resource/development-agriculture/>

User Guide 4 Channel 5V Optical Isolated Relay Module. (n.d.). Retrieved from Handsontec: <https://www.handsontec.com/dataspecs/4Ch-relay.pdf>

yasinihan.com. (2022, June 21). *Flutter Page Transition Package*. Retrieved from pub.dev: https://pub.dev/packages/page_transition



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

APPENDICES



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

APPENDIX A: NODERED UI PAGE GRAPH

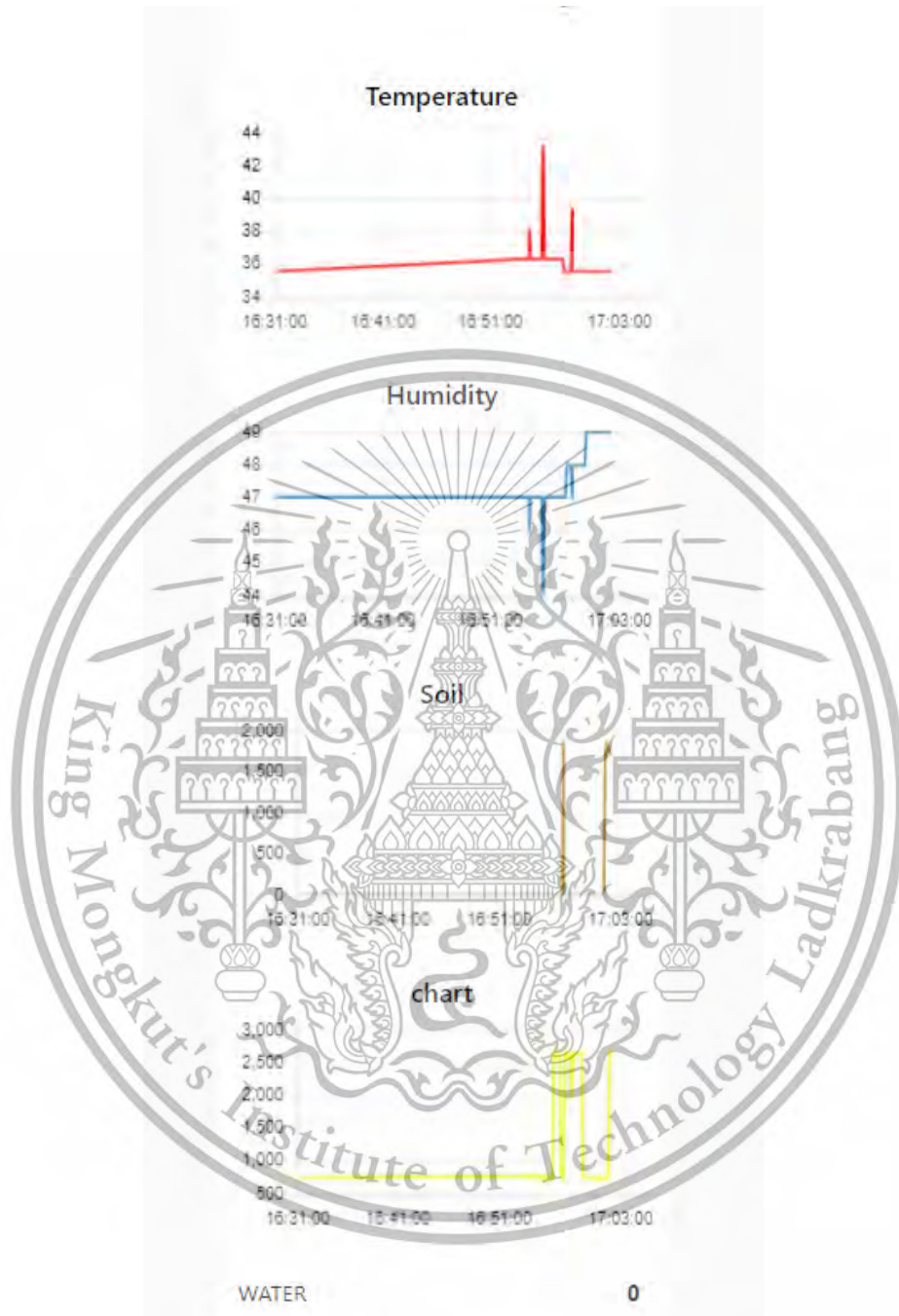


Figure 5. 1 Node-Red UI page graph section

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

APPENDIX B: SENSOR POSITION ON BREADBOARD

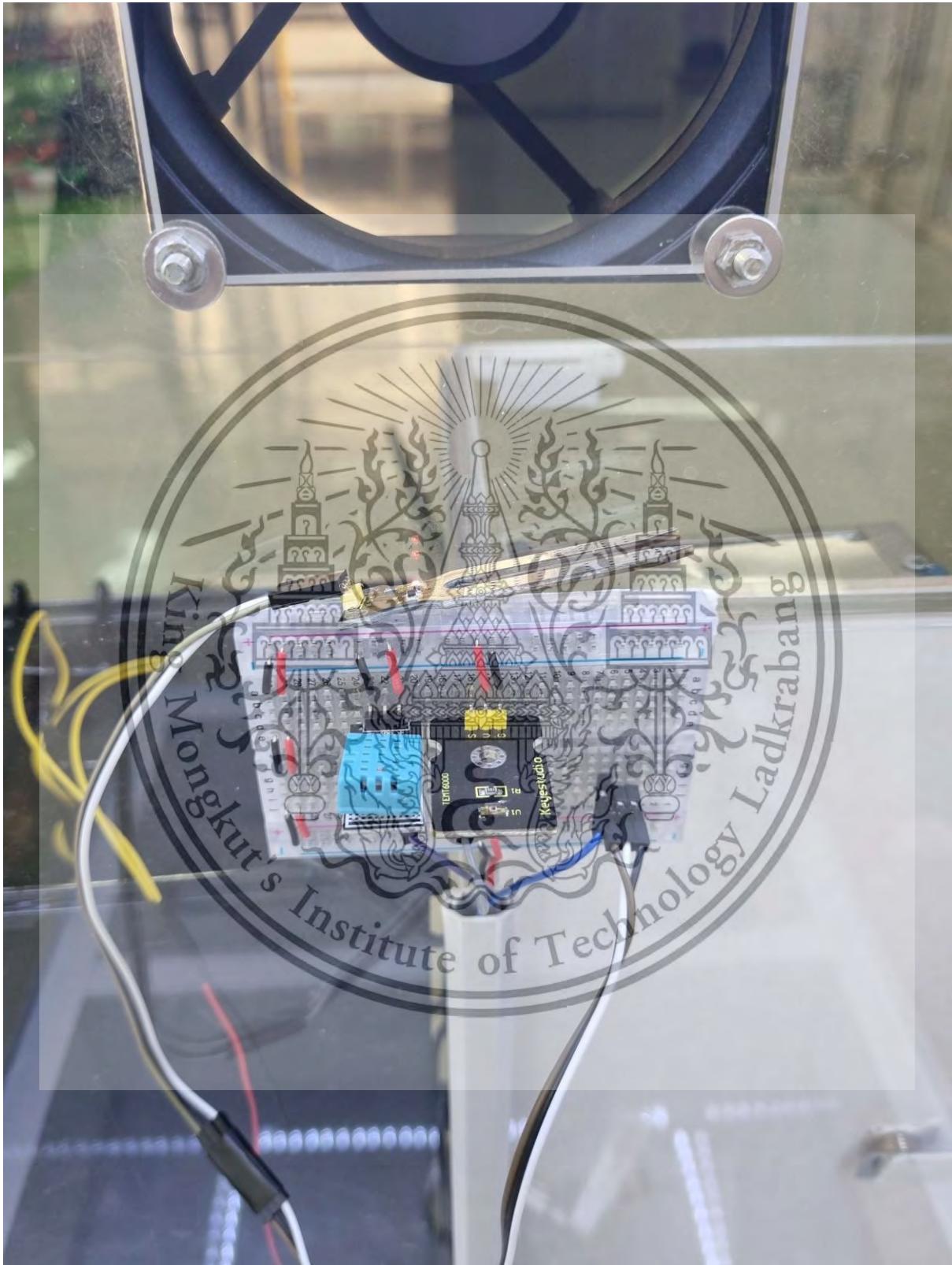


Figure 5. 2 Humidity, Temperature, and Light sensor install in the control box

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

APPENDIX C: ORIGINAL PROJECT SKETCH

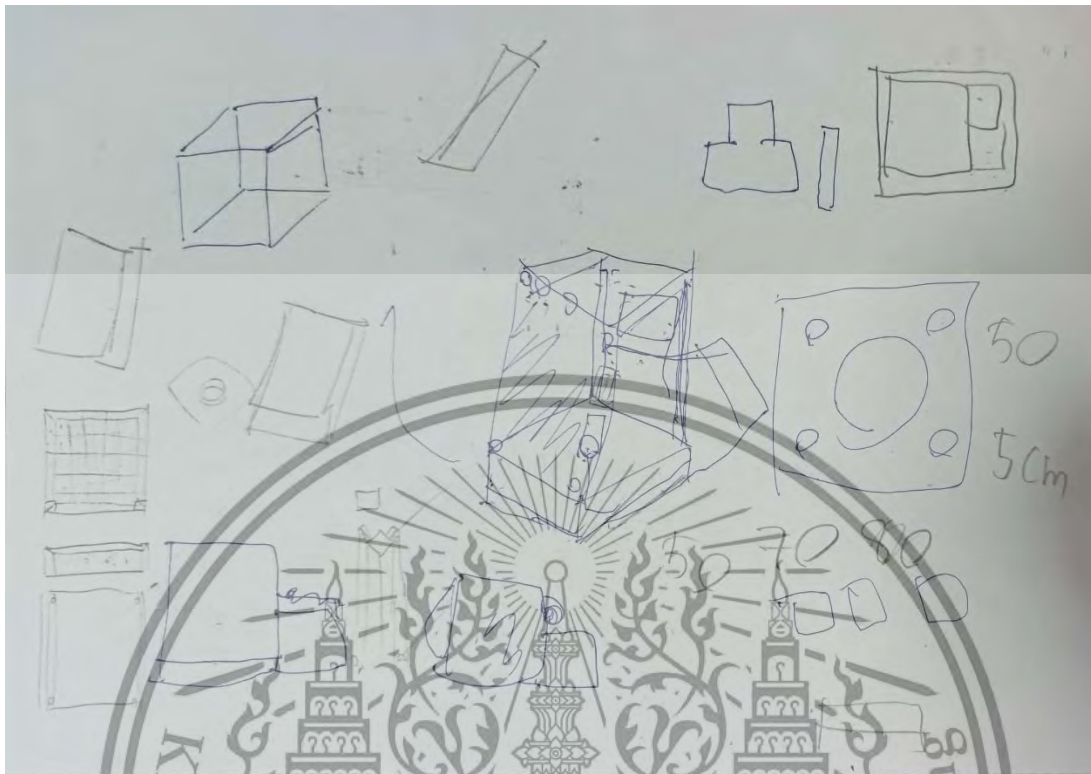


Figure 5. 3 First week planning sketch - Front

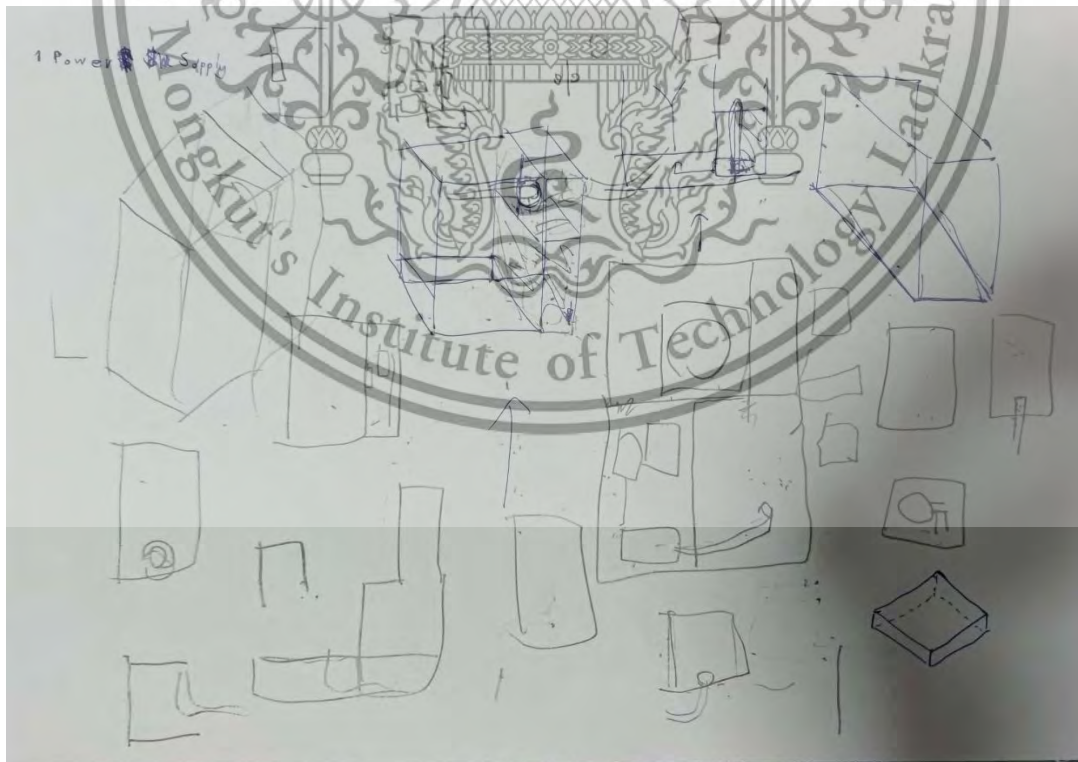


Figure 5. 4 First week planning sketch - Back

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

APPENDIX D: PROJECT 3D MODEL



Figure 5. 5 3D model of the latest version of the work piece

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

BIOGRAPHY



The author, Tri Thamwiwat, was born on June 19th, 2001 in Nonthaburi, Thailand. He is pursuing his fourth year of a bachelor's degree in Robotics and Artificial Intelligence at King Mongkut's Institute of Technology Ladkrabang and currently lives in Pakkred, Nonthaburi, Thailand. Before his time at the university, He has shown great interest in robotics and programming. Attending "Hours of code" hosted by Microsoft Thailand and joining robotic club in high school at Triam Udom Suksa. Being chosen as a school representative to attend a robotic competition.

During his time in university, He has taken part in extra training with topic related to industrial robotics. He also volunteered and became a teacher assistance helping out in microcontroller course.

He also demonstrates remarkable English skill. With experience in working with other while abroad thank to his experience as exchange student in Portugal (2018) and summer internship at Nagaoka institute of Technology (2022) working on thin film solar cell research and development. After returning from Japan. He also participates in Cooperative education. Taking internship at Sun plex engineering and software as a software engineer (2022).

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

BIOGRAPHY



The author, Weeraphat Wongbunnak, was born on May 6th, 2000, in Bangkok, Thailand. He is studying his fourth year of a bachelor's degree in Robotic and Artificial Intelligence Engineering at King Mongkut's Institute of Technology Ladkrabang (KMITL) and currently living in Samutprakarn, Bangphi, Thailand.

Weeraphat also joined company internship and Cooperative education program during his bachelor degree. He has begun his internship journey in summer period in 2022 at Fukuoka Institution of Technology (FIT) in Japan. Then, he joins Global Project-Based learning program (gBPL) which is a cooperative program between King Mongkut's Institute of Technology Ladkrabang (KMITL), Fukuoka Institution of Technology (FIT) and University Technology MARA (UiTM). In Autumn, he joined 108 Engineering co.ltd as an internee in software engineer position for 4 months.

BIOGRAPHY



The author, Worapon Yingyong, was born on June 7th, 2000, in Chon Buri, Thailand. He is pursuing his fourth year of a bachelor's degree in Robotics and Artificial Intelligence Engineering at King Mongkut's Institute of Technology Ladkrabang and currently living Bangkok City, Thailand. During his time at university, he has a summer internship in Fukuoka Institute of Technology, Japan for 1 month. Then, he has begun his internship since in the Sunplex Software and Engineering for 4 months.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content and cite the document when use.

BIOGRAPHY



The author, Yodkwan Theeraniti, was born on July 13th, 1998, in Bangkok, Thailand. He is studying in his fourth year of a bachelor's degree in Robotic and Artificial Intelligence Engineering at King Mongkut's Institute of Technology Ladkrabang (KMITL) and currently living in Wisetchaichan, Ang Thong City, Thailand.

Yodkwan also joined the company internship and Cooperative education program during his bachelor's degree. He has begun his internship journey in the summer period in 2022 at Fukuoka Institution of Technology (FIT) in Japan. Then, he joins the Global Project-Based learning program (gBPL) which is a cooperative program between King Mongkut's Institute of Technology Ladkrabang (KMITL), Fukuoka Institution of Technology (FIT), and Universiti Teknologi MARA (UiTM). In Autumn, he joined THE SUN PLEX ENGINEERING AND SOFTWARE CO., LTD as an internee in a software engineer position for 4 months.

Decomplex: Smart plant pot integrating with internet of things (IoT)

Tri Thamwivat, Weeraphat Wongbunnak, Worapon Yingyong, Yodkwan Theeraniti
Robotics and AI Department of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand

62011285@kmitl.ac.th, 62011293@kmitl.ac.th, 62011296@kmitl.ac.th, 62011297@kmitl.ac.th

Abstract— This thesis presents “Decomplex” The Smart Plant Pot Integrating with The Internet of Things, which aims to apply the knowledge gained by its authors to build a controlled system that can improve the agriculture industries. Planting the economic plant in Thailand is still not fully automated, and still uses human labor in watering and fertilizing. The “Decomplex” project has been created as a solution to provide an automated system for controlling the economic plant. Additionally, this project has the ability to control the actuators with a mobile application remotely. This project is made possible by integrating various hardware and software, including Firebase database, Flutter for mobile applications, Node-Red with raspberry pi, ESP32, sensors, and actuators. The mobile application can display temperature, light intensity, air and soil humidity, and also be used to control the actuators such as watering, giving light sources, and removing moisture. This project is a prototype of an automated system for large industries.

Keywords—IoT, Firebase, Flutter, Node-Red, Raspberry pi, ESP32, Temperature, Humidity

I. INTRODUCTION

Food is important to every living thing in the world because all living things need food to live. Each living things have their own methods to find food. Agriculture is one of the methods that human use for finding food supply. Bill and Melinda gate foundation is quoted as stating, "Investments in agriculture are the best weapons against hunger and poverty, and they have made life better for billions of people." According to the quote it shows how important agriculture by stating that better agriculture means better life. This paper represents the SMART PLANT POT INTEGRATING WITH THE INTERNET OF THING (DECOMPLEX) which is an agriculture monitoring system. Develop by Robotics and AI student from King Mongkut's Institute of Technology Ladkrabang (KMITL).

A. Agriculture Background

1) *Agriculture History*: Agriculture, food, and goods cultivated by farming are the main method to produce the world's food supply [1]. In long view of human history, human have nomadic hunter-gatherer lifestyles in their first period and then they continue to develop the tools to help them survive through the brutal period. Human have developed many tools that help them to survive and some of them have change their lifestyle forever such as agriculture. Agriculture first found about 12,000 years ago and it change human lifestyle from nomadic hunter-gatherer lifestyles to permanent settlement and farming (National Geographic, 2023). With the permanent settlement and farming lifestyle it means that the food supply can be farmed to meet the demand

and cause the human population to rocketed and it also created a new job like farmer as well. Therefore, agriculture have stay with human for a long time and it can change the way that human live by provide them higher opportunities to survive.

2) *Agriculture Technology*: Since the 2000, human population tend to grow up and demand more food to survive [2]. With the higher food demand. Human invented many methods to improve their food supply. To solve the insufficient of food supply human have use modern technologies to make higher crop productivity, reduce the budget, reduce impact on natural ecosystems, reduce chemical use and increase worker safety. To achieve those benefit human, have implement the modern technologies such as GPS technology, robotics system or even precision agriculture devices in their agriculture system which called modern agriculture (USDA,2023). From using those technologies, there are many products that use for modern agriculture such as greenhouse system and automatic agriculture system. To summaries, human pollution and food supply are direct variation and to maintain the food supply from direct variation state human have fused both agriculture and technologies together and called modern agriculture.

B. Problems

Growing of high-quality economic plant depend on many factors. Factors such as water, temperature, light, moisture in the soil and humidity. Plant in unstable environment can massively impact the quality and in the worst case it might cause the plant to die. Other than stable environment. Plant needs a daily care which require owner to water it every day and sometime the owner forgot to watering the plant and that give a negative impact to the plant. To solve those problems a system to look after the plants was create. A system which monitors the environment and automate the process to keep the plant in optimal growing condition. By integrating IoT to the product to give user more flexibility in monitoring and manage the control of the plant environment.

C. Research

1) *Internet of Things (IoT)*: Internet of Things or IoT is an object or device that have the capability to connect and exchange data with other device and system in the network. Usually refer to as Smart device. They are capable of collecting and processing data. Then communicate with other devices to be individually control.

2) *Micro Controller*: Micro Controller or MCU is a small computing device on a single integrated circuit chip. Design to be use in embedded application and automatic device. The

Forbidden to modify the content, and cite the document when use.

small size of MCU has led to them being integrated into many applications ranging from handheld power tools and toys to office machine, remote control, and medical device.

II. HYPOTHESIS

A. Green House

A greenhouse is a building made to control variables like temperature, humidity, light, and ventilation to provide the ideal environment for plant growth. Regardless of the weather outside, it offers a safe environment for plants to develop and thrive.

The most popular materials used to construct greenhouses are glass or transparent plastic, which permit light to enter and retain heat within the structure. In comparison to the outside world, the greenhouse's climate is warmer and more stable thanks to the trapped heat. The advantages of this regulated environment for plant growth include:

- Temperature control
- Protection from severe weather
- Better humidity control
- Pest and disease control
- Increased light exposure

B. Plant Growth

To evaluate plant growth, several parameters can be measured and assessed.

- Biomass: Calculate the plant's biomass, or overall weight. By gathering the plant, removing extra moisture, and weighing the plant material on a scale, this metric can be determined.
- Leaf area: Determine the size of the leaves, which may be a sign of their ability to absorb light. Using specialized tools like a leaf area meter or image analysis software.
- Root development: Carefully take the plant out of the growing medium and look at the roots to assess the root system. utilizing methods like washing the roots and utilizing a root scanner.
- Flowering and fruiting: Monitoring the production of flowers and fruits is vital because they serve as key markers of reproductive development. Assess the quantity of flowers or fruits that are produced as well as their quality (size, color, weight, etc.).

C. Project Hypothesis

An automatic control system that can regulate water, temperature, humidity, and light for plant growth will be as effective as or more effective than traditional control methods while requiring less labor and manual intervention. The hypothesis is drawn base on the following:

- Environmental parameters can be regulated precisely: Automatic system enables accurate adjustment of environmental factors. Compare to the conventional method that require manual changes may result in increased growth rate and higher yield.
- Consistency: Reducing swings and change that could adversely affect plant development. The automatic control creates a consistent environment for plant growth.

- Time Efficient: Automatic system minimizes the need for manual intervention and supervision. This may result in significant labor time save. Freeing up worker for other crucial tasks.
- Data gathering and analysis: Automatic system are capable of gathering real-time data on plant responses and environmental variables. The information can be analyzed and adapted to improve the growth of the plant.

III. METHODOLOGY

A. Gantt chart planning

The Gantt chart is used by the team to plan the step of development while working on the project. The chart can be divided into four main phases.

- 1) *Planning*: The first step of the project. Set up the scope and goals of the project. Rough schedule is also draft here to get a rough idea for what must be done to complete the project and who would be responsible for what part.
- 2) *Research*: Research theory and technology related to the project. Prototype and testing of equipment is also done here before work on the main project begin.
- 3) *Development*: Designing, Building, and testing the main project are done in this phase of work.
- 4) *Paper*: Finalizing the project. Writing research paper and preparation for presentation.

B. Plant Research

Asparagus fern is the plant of choice that the team has choose to be grow and use in testing. Asparagus fern (*Asparagus setaceus*), is a native South African plant but well known in Thailand as a decorative plant. It is a multipurpose plant that can be used on various occasions for a decorative purpose, especially in ceremonies and events. Its charming appeal makes it one of the best choices for indoor decoration, as it can survive without requiring direct sunlight and can grow continuously to 3 meters tall. The following is the optimal condition to for Asparagus fern to grow.

- Suitable environment: Indoor with stable temperature and humidity level.
- Temperature: 18 to 24 degrees Celsius. Extreme temperature fluctuation can negatively affect the plant.
- Soil: Medium soil is a well-drained and not immensely soggy soil that prevents root from rotting.
- Watering: Frequent watering to keep the soil moist but not too much which can cause root rot.
- Lighting: Does not require direct sun light but still a moderate amount of light to survive.
- Humidity: Medium to high. Avoid placing the plant in dry environment.

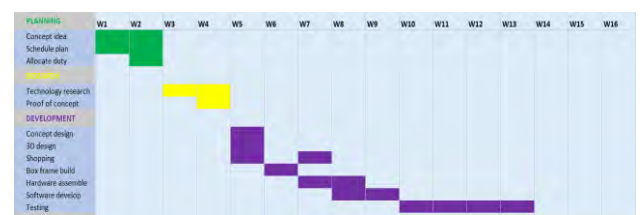


Fig 3.1 Gantt chart

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

C. Technology Research

With parameters and condition to control to ensure plant growth. Including humidity, light, and water. Research was conducted to find equipment needed to read and collect data, communicate the data to other part of the project in order to automate the process of controlling the growth condition of the plant.

1) Micro Controller

a) *Raspberry Pi*: A small single-board computers. Developed by the Raspberry Pi Foundation in association with Broadcom [3]. Originally develop as a teaching tool for computer science. Its versatility and performance see it adopt into many practical applications. The version we use in the project is Raspberry Pi 4 (Model B) which feature the ability to connect the board to external monitor. Which greatly help simplify the development process. The main purpose for the Raspberry Pi board in this project is to communicate with the database on the internet for further used. The detail of the software will be explained in its own section.

b) *ESP32*: A small microcontroller developed by Espressif Systems. [4] Acting as a central hub for reading data collected by set of sensors in the work piece. Its small size combines with low power usage and excellence performance make it perfect for its role.

2) Sensors

a) *Humidity & Temperature sensor (DHT11)*: Use to measure air humidity and temperature. The measurement range for humidity is 20 to 90%RH and 0 to 50 degrees Celsius for temperature. Both within use range of expected environment in the project. It is also highly accurate with minimal deviation for both value [5].

b) *Ambient light sensor (TEMT6000)*: Detect incoming light. The higher the light intensity. The higher the reading output. Can only detect light in human visible spectrum which is enough for the project [6].

c) *Water float switch (EP-4510)*: Use to detect water level in storage tank and send a warning when water level is low [7].

d) *Soil humidity sensor (Ks0049)*: Install in plant pot to check soil humidity. Dry soil give low output reading while humid soil will give higher reading [8].

3) Actuator

a) *Power supply*: Convert main electricity line high voltage to lower level that other component in the project can use. The specification use in Decomplex give output of 12V 15A.

b) *Relay Module (4 Channel)*: Act as a switch to control operation of other component [9]. The relay itself is control by signal from ESP32.

c) *LED strip*: Provide light [10] necessary for photosynthesis to the plant.

d) *Water pump*: Install in water tank to move water use for watering the plant. The pump use is a submerge type. Giving it high power and allow moving water for longer distance.

e) *Fan*: Use to move humid air out of the control environment. By doing so it also help reduce the temperature.

4) Designing

a) *Sketch*: Draw by hand to define general look and size of the project. Position of the circuit box and water tank relative to the control box is also considered here.

b) *Autodesk Inventor*: Use to create 3D model of the project to see the overall look of the final design and check for possible overlap of component before physical assembly begin.

c) *Drawing file*: Created from 3D model with Autodesk Inventor. Use as a reference by all member in the team to help in assembly.

5) Hardware Development

a) *Material selection*: Each part of the project require different material. Chosen base on the role it is expected to do. For the base, aluminium is chosen for it durability and for the structure of the control box. Acrylic is chosen for it water resistance, and it transparency to let the light through.

b) *Circuit box*: Micro controller, sensor, and actuator that will be place in circuit box are pre arrange before assembly begin to check the size needed and also to see potential wiring route.

c) *Water storage*: Another layer of plastic bucket is place in the acrylic box for safety in case of water leak. It also provide place to install the water pump and water level sensor.

6) *System Design*: Wire line are measure and cut to fit within the circuit box. Wiring diagram are drawn and use as reference to connect all component together.

7) Software Development

a) *ESP32*: The code within is used to send data collected with sensor to raspberry pi. The communication is possible by using MQTT protocol by publishing sensors topics. The code is divided into four main sections of Wifi setup, MQTT reconnection, subscribe topics, and publishing topics.

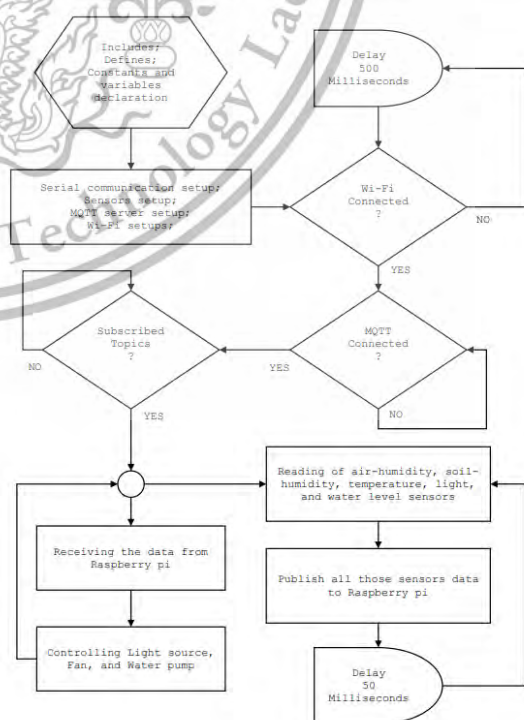


Fig 3.2 ESP32 connection flowchart

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

b) *Firestore*: A mobile and web application development platform that provides backend services and tools [11]. Including real-time database, cloud storage, authentication, hosting, and more. Decomplex use authentication tool to create sign-in method, email verification, password reset, email change, and SMS service verification [12]. And Real-time database tool to collect data gathered from sensors in the work piece [13].

c) *Raspberry Pi*: Act as a server for sending and receiving data to real-time database. Node-Red [14] is applied in raspberry pi for communicating with ESP32 using MQTT protocol, and real-time database.

d) *Flutter*: An open source software UI toolkit for building application for web, mobile, and desktop platforms. Flutter allows developer to build their styles and fast app that run smoothly on various platform. Making it an efficient tool for cross-platform application development. In Decomplex, Flutter is use to create and develop the mobile application which allow user to monitor and control the environment of the work piece.

IV. EVALUATION AND RESULT

A. Evaluation criteria

There are five main way that can be use to evaluate plant growth which are Height, Biomass, Leaf area, Root development, and flowering & fruiting. Because of the plant chosen in testing and difficulty in using some method. Height measurement is chosen as the way to evaluate plant growth.

B. Testing Method

Five environment is setup to test the theory that automatic environment control is equivalent or more efficient than manual control. And to see the impact of each variable on plant growth.

C. Test Result

Two sets of data are recorded. First is the height of plant growth in all five environments. The data is recorded every three days for a month. Second set of data collected is temperature, humidity and soil humidity reading of plant in automatic and manual control environment. This set of data is collected to compare the effectiveness of automatic system control of the environment in comparison to traditional manual control.

1) *Height data analysis*: By plotting the data collected into graph. Two interesting observation can be made. One, Plant with no temperature control die sometimes around day three to six. This indicated that temperature is a very important variable that should be prioritise. Two, sometimes around day nine. Plant in no water environment start to fall behind all other pot despite showing greater growth in earlier data. This show that while plant could continue growing without water. It still play an important role in plant growth.

TABLE I. TEST ENVIRONMENT CONDITION

Auto	The plant will be place inside the project workpiece and will be monitored and taken care of by automation.
Human Control	Human will take care of all three major condition of the plant. (Water, Light, Temperature)
No light Control	Similar to control but without extra light control.
No Water Control	No water is provided to the plant.
No Temperature Control	No temperature control is provided.

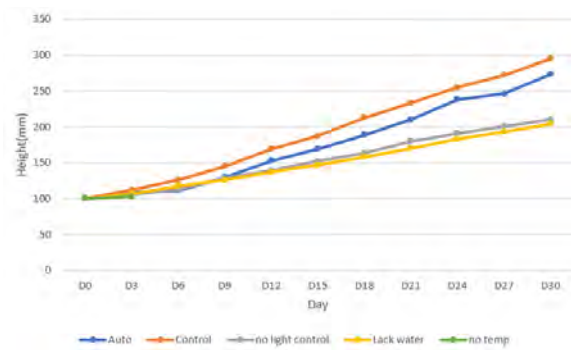


Figure 4.1 Height data graph

2) *Temperature, Air and Soil humidity analysis*: Comparing each variable with the other environment. For temperature and Air humidity, Data from both environment show similar value with automatic showing more fluctuation between each collection day but these are not too drastic. This leave soil humidity in which both environment show great fluctuation in data. The conclusion drawn from data show that both automatic and manual control are comparable, performance wise.

V. CONCLUSION

In conclusion, the experiment conducted to test the hypothesis that an automatic control system for regulating water, temperature, humidity, and light for plant growth would be as effective as or more effective than traditional control methods while requiring less labor and manual intervention has provided insightful results.

The results after 30 days have confirmed the hypothesis, demonstrating that the automatic control system is indeed as effective as traditional control methods, with a marginal 5% increase in efficiency achieved by human control. Furthermore, the experiment revealed that when water, light, and temperature are not properly regulated, plant growth declined by approximately 40-45% in height.

A. Learning Point

1) *IoT Protocol*: The implementation of advanced IoT technologies, such as ESP32, Firestore, Node-RED, and Flutter, proved instrumental in the successful development of the automatic control system. The integration of these protocols enabled seamless data communication, real-time monitoring, and remote-control capabilities. This underscores the significance of leveraging IoT protocols in future research and applications related to plant growth. By harnessing the power of interconnected devices and data analytics, researchers can enhance the efficiency, accuracy, and scalability of automated control systems.

2) *Cloud Control System*: The incorporation of cloud-based technologies, exemplified by Node-RED and Flutter, demonstrated the immense potential of leveraging cloud computing in plant growth applications. The utilization of these platforms enabled centralized data storage, analysis, and control, fostering greater accessibility and flexibility. Cloud-based control systems offer the advantage of remote management and scalability, allowing researchers to monitor and adjust plant growth parameters from anywhere. The

TABLE II. TEMPERATURE, AIR AND SOIL HUMIDITY DATA IN AUTOMATIC CONTROL ENVIRONMENT

Auto	D0	D3	D6	D9	D12	D15	D18	D21	D24	D27	D30
Temperature	23.5	24	24.3	23.4	23.7	22.8	24.5	23.8	23.9	23.7	23.6
Humidity	55	60	62	57	59	52	65	57	57	58	57
Soil Humidity	680	724	702	744	720	697	730	712	685	728	702

TABLE III. TEMPERATURE, AIR AND SOIL HUMIDITY DATA IN MANUAL CONTROL ENVIRONMENT

Control	D0	D3	D6	D9	D12	D15	D18	D21	D24	D27	D30
Temperature	23.7	23.6	23.6	23.7	23.6	23.6	23.5	23.6	23.6	23.6	23.6
Humidity	62	61	62	61	61	60	63	61	62	61	60
Soil Humidity	653	713	680	665	728	702	687	660	718	698	675

findings from this experiment encourage further exploration and development of cloud-based control systems, which can revolutionize agricultural practices and enable efficient resource utilization on a larger scale.

3) *Control Room*: The creation of a controlled environment, characterized by a dimension of 50×50×60 cm, served as a dedicated space for conducting the experiment. However, the versatility and adaptability of the control room setup extend beyond plant growth experiments. The controlled environment can be repurposed for various applications, such as studying the effects of environmental factors on other organisms or simulating specific conditions for scientific research. The modular design and precise control parameters make the control room an invaluable resource for conducting controlled experiments and fostering innovative investigations in diverse scientific disciplines.

B. Future Development

1) *Improved sensor technology*: The accuracy and range of data acquired by the automatic control system can be significantly improved through the improvement in sensor technology. Higher accuracy and sensitivity sensors can be integrated to offer more precise data on environmental factors such as light intensity, air quality, and nutrient levels. Incorporating sensors to gauge variables like pH levels and soil conductivity can also provide important insights into the unique nutrient requirements and growth circumstances of various plant species.

2) *Advanced actuator systems*: The effectiveness and versatility of the automatic control system can be improved by upgrading the actuator systems. Advanced irrigation systems with adjustable flow rates, variable-speed fans for precise air circulation control, and smart humidifiers that maintain ideal humidity levels are a few examples of advanced actuators that can be used to create growth conditions for plants that are more targeted and responsive. Real-time alterations and fine-tuning of environmental factors in accordance with plant requirements can be made possible by integrating feedback loops between sensors and actuators.

3) *Extended experiment duration*: Extending the duration of experiments beyond 30 days allows for long-term monitoring of plant growth patterns, responses, and adaptations to the automatic control system. A longer

experiment duration can provide insights into the system's stability, the plant's growth trajectory over time, and potential seasonal variations in growth patterns. Additionally, it allows for the observation of complex phenomena such as flowering, fruiting, and seed production, providing a more comprehensive understanding of plant life cycles within the controlled environment.

4) *Iterative system refinement*: Conducting the experiment several times can aid in detecting potential system limitations improving control as well as lowering uncertainties or errors. Researchers can improve the automatic control system's overall performance, robustness, and reliability by continuously enhancing it. With each iteration, the insights gained from earlier tests can be expanded upon, resulting in a more optimized and effective control system design.

5) *Plant diversity and specialty*: By include more plant species in the automatic control system research, we may learn more about the unique growth needs and responses of various plant species. Researchers can better understand how the control system can be adjusted to various plant varieties by including a varied range of species, such as economically relevant crops, medicinal plants, or environmentally significant flora. Using this knowledge, specific applications in horticulture, agriculture, and ecological research may be produced, as well as the creation of unique growth protocols and optimal environmental conditions for diverse plant species.

ACKNOWLEDGMENTS

The completion of this project could not have been possible without the help of the Wongbunnak family. Additionally, we would like to give exclusive thanks to Wongbunnak family for the workplace, supplies, equipment, and advice. The advice from Wongbunnak family significantly enhanced our project development progress and made the project approach our objective.

We express our deepest thanks to the Robotics and AI engineering that provided us with the budget to complete this project. Finally, we are indebted to Dr. Pitikhate Sooraksa, the project's chief consultant. With his permission and advice, it allow us to be more ambitious to present the magnificent project.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

REFERENCES

- [1] The Development of Agriculture, "The Development of Agriculture," [Online]. Available: <https://education.nationalgeographic.org/resource/development-agriculture/>.
- [2] Agriculture Technology, "Agriculture Technology," [Online]. Available: <https://www.nifa.usda.gov/topics/agriculture-technology>.
- [3] raspberry-pi-4-datasheet.pdf, "DATASHEET Raspberry Pi 4 Model B," June 2019. [Online]. Available: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>.
- [4] esp32_datasheet_en.pdf, "ESP32 Series Datasheet," 01 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [5] Ks0034 keyestudio DHT11 Temperature and Humidity Sensor - Keyestudio Wiki, "Ks0034 keyestudio DHT11 Temperature and Humidity Sensor," 26 May 2021. [Online]. Available: https://wiki.keyestudio.com/Ks0034_keyestudio_DHT11_Temperature_and_Humidity_Sensor.
- [6] Ks0098 keyestudio TEMENT6000 Ambient Light Sensor - Keyestudio Wiki, "Ks0098 keyestudio TEMENT6000 Ambient Light Sensor," 7 January 2021. [Online]. Available: https://wiki.keyestudio.com/Ks0098_keyestudio_TEMENT6000_Ambient_Light_Sensor.
- [7] FC_FD_970528, "MINI FLOAT LEVEL SWITCH," [Online]. Available: [http://www.interflow-](http://www.interflow-th.com/information_Product(more)/FINETEK-F(C)D_MINI_FLOAT_LEVEL_SWITCH.pdf)
- [8] Ks0049 keyestudio Soil Humidity Sensor - Keyestudio Wiki, "Ks0049 keyestudio Soil Humidity Sensor," 7 January 2021. [Online]. Available: https://wiki.keyestudio.com/Ks0049_keyestudio_Soil_Humidity_Sensor.
- [9] "User Guide 4 Channel 5V Optical Isolated Relay Module," [Online]. Available: <https://www.handsontec.com/dataspecs/4Ch-relay.pdf>.
- [1] 5050 LED Datasheet: The Ultimate Guide - SIRS Electronics, Inc., "Understanding 5050 LED Strip Datasheets 5050 LED Datasheet: The Ultimate Guide," [Online]. Available: <https://sirs-e.com/general/5050-led-datasheet/>.
- [1] Security Rules and Firebase Authentication | Firebase Security Rules, "Security Rules and Firebase Authentication," 4 May 2023. [Online]. Available: <https://firebase.google.com/docs/rules/rules-and-auth>.
- [1] firebase.google.com, "firebase_auth," 16 May 2023. [Online]. Available: https://pub.dev/packages/firebase_auth.
- [1] firebase.google.com, "firebase_database," 16 May 2023. [Online]. Available: https://pub.dev/packages/firebase_database.
- [1] node-red-dashboard (node) - Node-RED, "node-red-dashboard," 11 March 2023. [Online]. Available: <https://flows.nodered.org/node/node-red-dashboard>.



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.