A PROTOTYPE OF INDOOR HYDROPONICS FARMING FRAMEWORK

BY

LEE SONG YAO

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)
COMMUNICATIONS AND NETWORKING
Faculty of Information and Communication Technology
(Kampar Campus)

OCTOBER 2021

REPORT STATUS DECLARATION FORM

Title: A PROTOTYPE OF INDOOR HYDROPONICS FARMING

Academic Session: OCTOBER 2021

I LEE SONG YAO

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
1079b, Jalan Raja Muda Musa,
36000 Teluk Intan, Perak

Ts Dr Goh Hock Guan
Supervisor's name

Date: 26 November 2021

Date: 26/11/2021

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

Date: 26 November 2021

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that LEE SONG YAO (ID No: 19ACB05829) has completed this final year project entitled "A PROTOTYPE OF INDOOR HYDROPONICS FARMING FRAMEWORK" under the supervision of Ts Dr Goh Hock Guan (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

LEE SONG YAO

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**A PROTOTYPE OF INDOOR HYDROPONICS FARMING FRAMEWORK**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :

Name : LEE SONG YAO

Date : 26 November 2021

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Ts Dr Goh Hock Guan for his insightful comments and suggestions which gave me a lot of help in the process of the project development. His suggestion had help me a lot in completing this project without any delay. My appreciation also goes out to my family and friends for their encouragement and support all through my studies.

ABSTRACT

This project is an indoor hydroponic farming framework for academic purpose. It works by using a single board computer as the main controller to command and receive data from sensors and actuators. The indoor hydroponic farming framework contains the main controller, sensors, and actuators which will help to maintain the health of the crops within the entire life cycle until the time to harvest. The data collected will be publish to the cloud service provider dashboard for storage purpose and to display in real-time to the user while a camera is placed within the farming framework which will stream the live feed of the farm into a streaming platform then embedded into the cloud service provider dashboard for live viewing of the farm at any location. The indoor hydroponic farming framework also being developed in an isolated environment which will attempt to reduce dependencies to the weather differ to the major traditional soil-based farming which is highly dependent on the weather. With the sensors, actuators, and the camera being placed, the indoor hydroponic farming framework aims to be having least human interaction as possible when the crops are growing until the time of harvest. In short, this project proposed an indoor hydroponic farming framework which is developed to be less dependent on weather by isolating from the outdoor environment, real-time monitoring of farming framework status by reading data from sensors then transmit through MQTT protocol with live feed of the farm embedded at the same SaaS cloud service dashboard.

TABLE OF CONTENTS

REPORT STATUS DECLARATION FORM .... **I**

DECLARATION OF ORIGINALITY .... **III**

ACKNOWLEDGEMENTS .... **IV**

ABSTRACT .... **V**

LIST OF TABLES .... **VIII**

LIST OF FIGURES .... **IX**

LIST OF ABBREVIATIONS .... **XI**

CHAPTER 1 INTRODUCTION **1**

1.1 PROBLEM STATEMENT AND MOTIVATION 1
1.2 PROJECT OBJECTIVES 2
1.3 PROJECT SCOPE 3
1.4 IMPACT, SIGNIFICANCE AND CONTRIBUTION 4
1.5 ORGANISATION OF THE REPORT 4

CHAPTER 2 LITERATURE REVIEW **4**

2.1 REVIEW OF THE TECHNOLOGIES 4
2.1.1 Hardware platform 4
2.1.2 Firmware/OS 6
2.1.3 Cloud Services 8
2.1.4 Programming Language 9
2.1.5 Summary of the Technologies Review 10
2.2 REVIEW OF THE EXISTING SYSTEMS/APPLICATIONS 12
2.2.1 Hommons: Hydroponic Management and Monitoring System for an IOT Based NFT Farm Using Web Technology 12
2.2.2 IOT Hydroponics Management System 15
2.2.3 Automation of Hydroponics Green House Farming using IOT 17
2.2.4 IoT based Hydroponic Farm 19
2.2.5 Internet of things (IoT) applied to an urban garden 20
2.2.6 Summary of Existing Systems/Applications 22

CHAPTER 3 SYSTEM METHODOLOGY **23**

3.1 SYSTEM DEVELOPMENT MODEL 23
3.1.1 Waterfall System Development Model 23
3.1.2 Spiral System Development Model 24
3.1.3 Scrum System Development Model 25
3.1.4 Kanban System Development Model 26
3.1.5 Selected Model 27
3.2 SYSTEM REQUIREMENT 28
3.2.1 Hardware 28
3.2.2 Software 30
3.3 FUNCTIONAL REQUIREMENT 32
3.4 PROJECT MILESTONE 33
3.5 ESTIMATED COST 34
3.6 CONCLUDING REMARK 35

CHAPTER 4 SYSTEM DESIGN **36**

4.1 SYSTEM ARCHITECTURE 36
4.2 FUNCTIONAL MODULES IN THE SYSTEM 37
4.3 SYSTEM FLOW 38
4.4 GUI DESIGN 39
4.5 CONCLUDING REMARK 40

CHAPTER 5 SYSTEM IMPLEMENTATION 41

5.1 HARDWARE SETUP 41
5.2 SOFTWARE SETUP 44
5.3 SETTING AND CONFIGURATION 45
5.4 SYSTEM OPERATION 46
5.5 CONCLUDING REMARK 50

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION 51

6.1 SYSTEM TESTING AND PERFORMANCE METRICS 51
6.2 TESTING SETUP AND RESULT 52
6.3 PROJECT CHALLENGES 58
6.4 SWOT 59
6.5 OBJECTIVES EVALUATION 60
6.6 CONCLUDING REMARK 61

CHAPTER 7 CONCLUSION & RECOMMENDATION 62

7.1 CONCLUSION 62
7.2 RECOMMENDATION 63

BIBLIOGRAPHY 64

APPENDIX A A-1

APPENDIX B B-1

APPENDIX C C-1

LIST OF TABLES

Table Number	Title	Page
Table 2.2.6.1	Summary of the existing system.	22
Table 3.2.1.1	specification of the main controller Raspberry Pi 3B+.	28
Table 3.2.1.2	specification of the 5MP IR Adjustable Pi Camera.	28
Table 3.2.1.3	specification of the 5V 8 channel relay module.	28
Table 3.2.1.4	specification of the DHT22 humidity & temperature sensor module.	28
Table 3.2.1.5	specification of the Keyestudio TDS sensor kit.	29
Table 3.2.1.6	specification of the ADC MCP3008.	29
Table 3.3.1	functional requirement of the system.	32
Table 3.4.1	gantt chart of the system development for FYP 1.	33
Table 3.4.2	gantt chart of the system development for FYP 2.	33
Table 3.5.1	estimated cost to setup the system.	34
Table 6.2.1	comparison of thermometer with DHT22.	52
Table 6.2.2	comparison of TDS/EC Meter with TDS sensor.	52
Table 6.2.3	growth of crops.	54-57
Table 6.4.1	SWOT table.	59

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1.1.1	Raspberry Pi 3B+	4
Figure 2.1.1.2	Arduino Uno Rev 3	5
Figure 2.1.1.3	Micro:bit V2	5
Figure 2.1.2.1	Raspberry Pi OS connected from a RealVNC viewer	6
Figure 2.1.2.2	Arduino IDE and tools to update bootloader	7
Figure 2.2.1.1	web management page for monitoring.	12
Figure 2.2.1.2	pre-set value for automation monitoring.	12
Figure 2.2.1.3	connection from the solar panel to the hydroponic farm.	13
Figure 2.2.1.4	setup of the hydroponic farm.	13
Figure 2.2.2.1	component used in this project.	15
Figure 2.2.2.2	setup of the hydroponic farm.	15
Figure 2.2.2.3	partial hardware prototype	16
Figure 2.2.3.1	component used in this project.	17
Figure 2.2.3.2	prototype of the monitoring and supply control system	17
Figure 2.2.4.1	block diagram of this project.	19
Figure 2.2.5.1	Physical architecture of the embedded system.	20
Figure 3.1.1.1	figure representative of Waterfall development model.	23
Figure 3.1.2.1	figure representative of Spiral development model.	24
Figure 3.1.3.1	figure representative of Scrum development model.	25
Figure 3.1.4.1	figure representative of Kanban development model.	26
Figure 3.1.5.1	Kanban Board which visualize the task.	27
Figure 3.2.2.1	desktop interface of the Raspberry Pi OS.	30
Figure 3.2.2.2	main controller connected through RealVNC Viewer.	31
Figure 4.1.1	hardware arrangement and connection.	36
Figure 4.3.1	system data publishing flowchart.	38
Figure 4.3.2	live feed flowchart.	38
Figure 4.3.3	automated grow light flowchart using Crontab.	38

Figure 4.4.1	Ubidots Dashboard.	39
Figure 5.1.1	connection from the main controller Raspberry Pi header to breadbox.	41
Figure 5.1.2	grow light & fans connected to the relay module.	43
Figure 5.1.3	grow light & fans wire which are modified.	43
Figure 5.2.1	menu of Raspberry Pi Imager.	44
Figure 5.3.1	raspi-config menu through SSH.	45
Figure 5.4.1	command to stream live feed to youtube live through FFMPEG.	46
Figure 5.4.2	process of the Ubidots MQTT python script is executed.	47
Figure 5.4.3	crontab for the growlight automation.	47
Figure 5.4.4	youtube live interface.	47
Figure 5.4.5	HTML Canvas editor of Ubidots Dashboard.	48
Figure 5.4.6	main controller is registered into the Ubidots services.	48
Figure 5.4.7	selection to define variable and place widget.	49
Figure 5.4.8	interface of the Ubidots Dashboard.	49
Figure 6.2.1	YouTube Live previous live stream.	53

LIST OF ABBREVIATIONS

ADC	Analog Digital Converter
CSI	Camera Serial Interface
DWC	Deep Water Culture
EC	Electricity Conductivity
FPS	Frame Per Second
GUI	Graphical User Interface
PIP	Pip Installs Packages
RDP	Remote Desktop Protocol
RTMP	Real-Time Messaging Protocol
SBC	Single Board Computer
SPI	Serial Peripheral Interface
TDS	Total Dissolved Solids
URL	Uniform Resource Locator
VNC	Virtual Network Computing

CHAPTER 1 INTRODUCTION

This section of the proposal will cover the problems and the motivations alongside with the scope of the research which will be conducted to solve the problems stated.

1.1 Problem Statement and Motivation

As the government first announced the Movement Control Order (MCO), the strict Standard Operating Procedure (SOP) travel restriction require all citizen to stay within their housing premises and away from the workplace premises, the supply chain which used to bring food from production to consumer are affected as workers are not allowed to reach their workplace which causes the issues to the balanced supply chain circulation, which cause at least 1 millions ringgits of lost per day for at least 2300 farmers and is forced to discard up to 200 metric tonnes of vegetables per day as senior lecturer of Food Science and Technology Programme, Faculty of Applied Sciences, UiTM, Ts. ChM. Dr. Raseetha Vani Siva Manikam pointed out at *New Strait Times* (MD SAAD, 2021). Thus, the idea of self-sustainability farms started to emerge as an idea among the people living in the country to ensure the availability of fresh vegetables and food security for the near future (Wong, 2020). But, even though the idea sparks, the process of preparing farms to be able to sustain daily consumptions require intensive labour and care while most of the people have little time to took care and monitor the indoor farm due to the need of working from home or searching for work to uphold daily lives.

The indoor hydroponic farming requires at least six factors to grow which is surrounding temperature, surrounding humidity, nutrient value, nutrient temperature, light, pH, oxygen content, etc. There are a number of solutions which aid in the growing of indoor hydroponic farm but most of them are enterprise-oriented and require a special allocated spaces to deploy. The motivation of this project is to create a consumer-oriented indoor hydroponic farm framework which is isolated from the unpredictable weather and equipped with complete sensors & actuators with a camera to observe the indoor hydroponic farm remotely.

1.2 Project Objectives

1. To create a fully automated indoor hydroponic farm framework.
   - Tends the crops within the farm automatically with the least human interaction possible.
2. To design the indoor hydroponic farming framework with the IoT technologies.
   - Using Single Board Computer (SBC) Raspberry Pi 3B+ as the main controller to collect data from sensors, publish data online, and control actuators to modify the status of the farm base on analysing data collected from the sensors.
3. To maintain the start-up of the farming framework to be easy in sourcing and setting up.
   - The framework of the farm is build using everyday material which should be found as locally as possible, and the setup should be straight forward for stress-free deployment in any other location.

1.3 Project Scope

This proposal will be concentrating on developing automated indoor isolated hydroponics farming which the actors can be controlled completely by SBC such as Raspberry PI. The proposed framework will be housing and maintaining the necessary necessity required by the choice of plant which is *Water Spinach*. The water spinach is selected due to its fast growth, adaptive to hot weather and is beginner friendly.

A few factors will be taken care through different sensors which is:

- Temperature,
- Humidity,
- TDS, &
- EC.

Due to the nature of farm is to be isolated from the nature habitat, there will be other factors which is important to maintain the health of the water spinach throughout the entire project. Such as:

- Lighting, &
- Air movement

Additionally, a Raspberry Pi Night-Vision Camera will be placed within the framework will project images or live feed of the indoor isolated hydroponics farm remotely and an air stone to ensure the roots does not suffocate by receiving sufficient oxygen from the nutrient solution.

1.4 Impact, significance and contribution

Since the pandemic hits the country, the dependency of food supply through freight is affected as work force require manual transportation using actual drivers. Given that the current pandemic is going to be staying for a prolong period, there are chances that the citizens are unable to obtain fresh goods as needed to stay within their premise to reduce virus transmission as much as possible. Besides, there are also a lot of chances that the freight might be affected due to lockdown in various location which might be the concentrated location for agriculture activities.

Therefore, an indoor hydroponic farm which require less water, almost completely automated, and will generate fresh goods is one of the best options to have in this uncertain period.

1.5 Organisation of the Report

This proposal is organized into seven (7) chapters. The first chapter (Chapter 1) will be the introductory chapter, the introductory chapter mentions the problem statement, motivation, project objectives, project scope, and the expected contribution. Chapter 2 reviews five (5) of the existing technologies which is developed by the previous researcher, and the summary of existing technologies. Chapter 3 describe the methodology and the steps taken to develop the system from scratch. Chapter 4 compiles the preliminary works, tasks required & completed, with the result of implementing proposed system. Chapter 5 describe the implementation of the entire software in every aspect. Chapter 6 reviews the developed system in terms of performance and usability. Chapter 7 concludes the proposed system deployment and provide recommendation for future references.

CHAPTER 2 LITERATURE REVIEW

2.1 Review of the Technologies

2.1.1 Hardware platform

There are a few different hardware platforms which can be used to act as the main controller to retrieve data from sensors and control the actuators.

One of the platforms is the Pi 3B+ is a single board computer which is capable to run a full-fledge GNU/Linux OS without any issues with the support of ARM optimized OS developed primarily for the credit card size computer. With the existence of the GPIO header pins, and the camera port, the Raspberry Pi 3B+ brings out a huge number of possibilities to create project without breaking the bank & motivates user to eventually develop methods to prototyping without soldering.

Raspberry Pi 3B+ is equipped with the Broadcom BCM2837B0 Cortex-A53 64-bit SoC @ 1.4 GHz processor, 1GB LPDDR2 SDRAM, 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2 BLE, Gigabit Ethernet over USB 2.0, Extended 40-pin GPIO header, Full-size HDMI, 4 USB 2.0 ports, CSI camera port, DSI display port, 4-pole stereo output & composite video port, Micro SD port, and 5V/2.5A Micro USB DC power input (Raspberry Pi 3 Model B+, 2021).

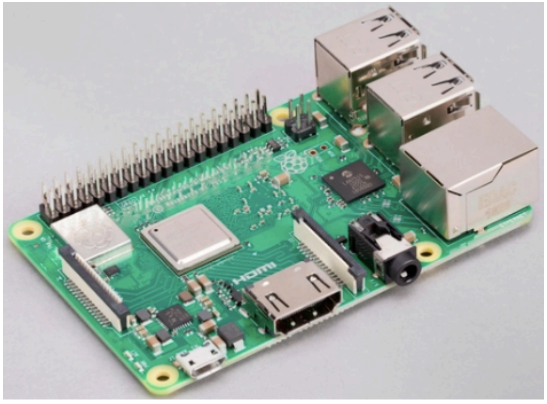


Figure 2.1.1.1 shows a Raspberry Pi 3B+.

The next platform which is used by existing systems are Arduino Uno Rev 3. The Arduino Uno is a microcontroller which equipped with analog inputs with digital I/O pins allows the user to connected both analog & digital sensors without any additional converter such as MCP3008 or ADC1115. The Arduino Uno works by flashing the programs which is needed in modified C++ code specifically developed for the microcontroller using a computer which had the Arduino IDE installed.

Arduino Uno Rev 3 is equipped with the ATmega328P @ 16 MHz microcontroller, 2 KB SRAM, 1 KB EEPROM, 32 KB flash memory, 14 digital I/O pins (6 PWM output supported), 6 analog input pins, USB - B for connection or power supply, and 9 -12V 2.1 mm centre positive barrel jack connector for power supply (Arduino Uno Rev3 | Arduino Official Store, 2021).

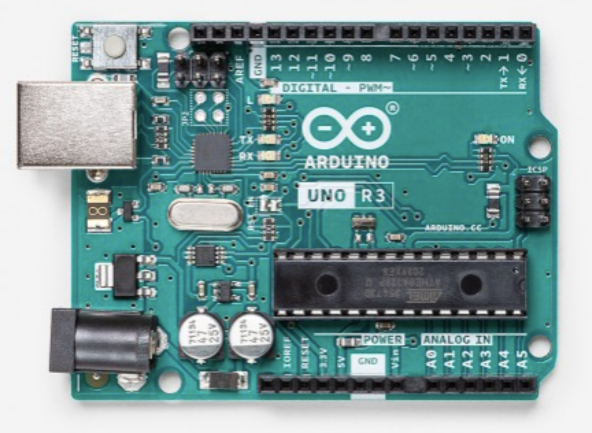


Figure 2.1.1.2 shows an Arduino Uno Rev 3.

The other hardware platform other than the two above are the Micro:bit V2, another microcontroller which equipped with analog inputs and digital I/O to allow analog input without converter. The micro:bit works by connecting the board to the computer and transfer the hex file into the micro:bit board like a USB flash drive.

The micro:bit V2 is equipped with nRF52833 ARM Cortex-M4 @ 64 MHz microcontroller, 128 KB SRAM, 512 KB flash memory, 25 digital pins & 4 dedicated GPIO with PWM support, ST LSM303 motion sensor, MEMS Mic, Piezo Buzzer, LED indicator, 2.4 GHz micro:bit radio, 5.1 BLE, 5V micro USB port for connection and power supply, and 3V battery port (Arduino Uno Rev3 | Arduino Official Store, 2021).

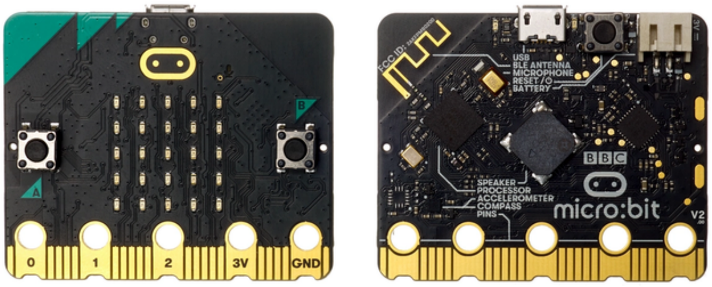


Figure 2.1.1.3 shows an Micro:bit V2.

2.1.2 Firmware/OS

The Raspberry Pi hardware relies on the Raspberry Pi OS to operate. Raspberry Pi OS which released on May 7th, 2021 with the kernel version of 5.10 is the current optimized full-fledged GNU/Linux OS modified for the Raspberry Pi computers to serve as the interface for the user to interact with the Pi computers. The OS has most of the feature includes in the famous desktop distro and the software package can be manage through APT. The OS can be download through the Raspberry Pi official webpage and burn to the sd card using dd or diskutil. There are also a Raspberry Pi Imager which can be used to burn the OS to sd card in a much more user-friendly method.

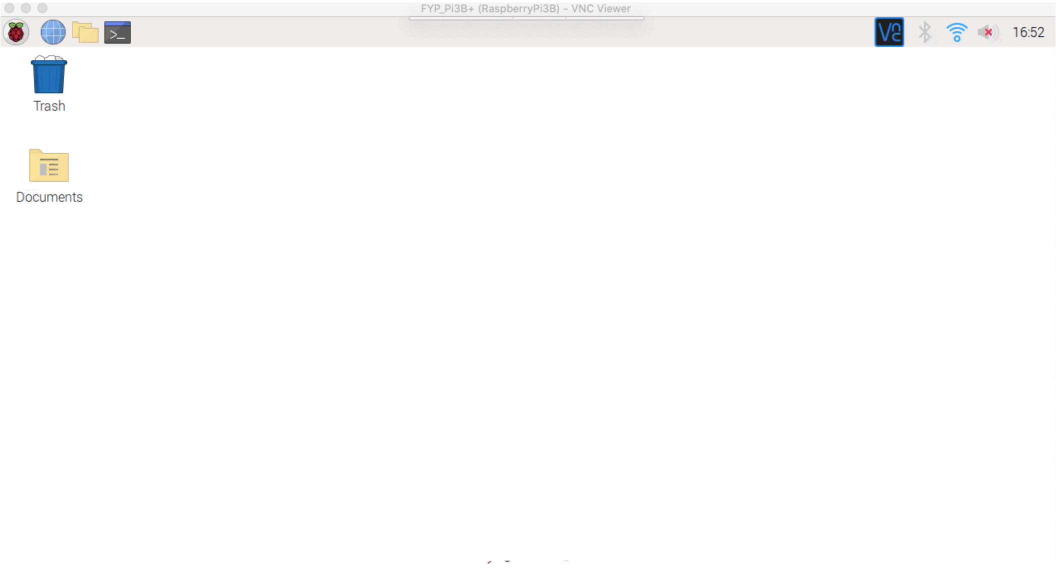
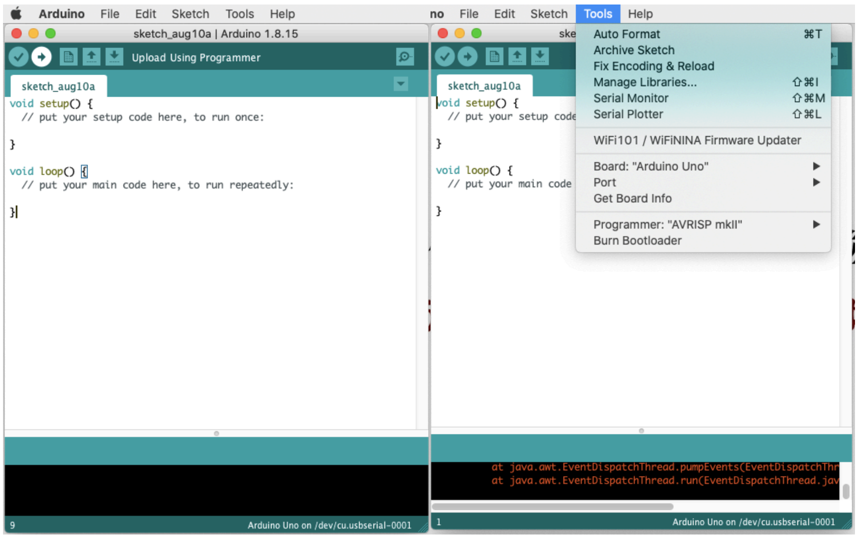


Figure 2.1.2.1 shows a Raspberry Pi OS connected from a RealVNC viewer.

Differ to the Raspberry Pi, the Arduino hardware does not have an OS to operate on as they are considered microcontroller. The code wrote on the Arduino IDE sketchbook will upload the content of the sketchbook into the flash memory on the Arduino hardware. The Arduino hardware comes with a bootloader which will be called whenever there is a reset on the hardware. Whenever the bootloader is called, the bootloader will load the program from the flash memory and perform actions programmed regardless of if the hardware is connected to Arduino IDE or not. The bootloader will be provided by the Arduino when shipped, but there are methods to modify the bootloader to suit special usages.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThr
at java.awt.EventDispatchThread.run(EventDispatchThread.jav
```

Figure 2.1.2.2 shows the Arduino IDE and tools to update bootloader.

Micro:bit works by placing the hex file into the flash memory like using a USB flash drive regardless if the file is the coded program or the firmware. The firmware of micro:bit is much more simpler and direct as the board had limited functions and is smaller in sizes. When the board is powered on, the firmware will load the program hex file which is placed in the flash memory.

2.1.3 Cloud Services

Cloud services are beginning to be a trend thanks to the help of the better network speed & coverage compared to the previous decade. In this era, the fibre internet connects the server to the users more efficiency and provide more possibilities on what service can be provided to the users which lead to the development of as-a-service which provide different level of cloud services to the user.

Besides as-a-service provided to the users, there are also another type of cloud services which is the publish-subscribe known as the MQTT which works by having two (2) types of network entities, the broker, and the client. The broker serves as the server which received all published messages from clients and transmit them to the clients subscribed to the same topic on the same broker URL.

Virtual Network Computing (VNC) is another cloud services which provide their user the ability to remote access their computing devices in remote locations. The Remote Framebuffer protocol is developed to passes the screen from a device to another regardless of which OS is used in both clients. VNC usually works in local area network, but some companies developed cloud services which the user can log into any client in anywhere as long as the clients is connected to the internet.

2.1.4 Programming Language

Python programming language is largely used by lots of projects for it ease to import, code, and execute. Code readability is also a major focus by using significant indentation to help producing clear & logical code. Python's designer vision of small core language with large standard library helps shaping python into a highly extensible language which allow user to use any library required or develop new library for the language when there is a need for it. Besides, large amount of library is available on the internet which help in interfacing sensors and most of the python libraries are available through package installer for python (pip) which install the dependencies into the OS without needing to download individual library from source and install the library manually.

C++ is also a fairly popular language and is used widely with a lot of variations to suit different purposes. In this proposal, the variation of C++ to be reviewed is the modified C++ for the Arduino hardware which include addition special methods and functions. The Arduino IDE is used to write, contain the library, and upload the code to the flash memory. The library of Arduino can be installed using the built library manager or import from .zip folder downloaded from other sources.

MicroPython is another programming language which is developed for the microcontroller to execute on bare metal as most of the standard python library consume too many resources when running on microcontroller. MicroPython aims to be as compatible as the standard Python yet to be compact to fit into 256k of flash memory & 16k of RAM. Besides, MicroPython also having an advantage which allows the user to call different language such as C, C++, BASIC, and assembly (Garrett, 2021).

2.1.5 Summary of the Technologies Review

In the current era, the network technologies development had improved drastically to the point where a cloud service can provide services identical to physical on-premises server with just the need of high-speed fibre internet.

There are a few hardware platforms which had been used in the past as the main controller for some projects. In this proposal, Raspberry Pi 3B+, Arduino Uno Rev3, and Micro:bit V2 is reviewed. Each of the platform had its pros and cons when being compared. The Raspberry Pi 3B+ is a single board computer with higher performance where a full-fledge GNU/Linux OS can be run on without any issues. Besides, it also equipped with HDMI to display the desktop environment and audio jack to output audio. The downside of the Raspberry Pi 3B+ is it is the most expensive among the comparison and require analog to digital convertor. The Arduino Uno Rev 3 is a microcontroller which is popular among the hobbyist to be used as the controller for the project. Arduino Uno Rev 3 is equipped with analog input pins which can be used to retrieve analog data directly. The downside of the Arduino Uno rev 3 is that the performance is lower compared to the Pi 3B+ and a computer is required to modify & upload the program written to the flash memory. The micro:bit V2 is another microcontroller which is popular for education purposes as the starter board to learn coding. Similar to the Arduino Uno, the micro:bit is equipped with both analog and digital pins. Differ from the Arduino Uno, the micro:bit V2 does not required any specific IDE to write and upload to the flash memory.

Firmware is developed to be run in compact and limited hardware especially microcontroller which have only KBs of resources so most of the firmware had limited and specific functions. In the other hand, GNU/Linux OS required a lot more resources estimated in GBs. GNU/Linux OS provide desktop environment to the user and program can be written and run within the single board computer with just a display and IO devices attached.

As the times goes, the technologies which can be used within a small to project increased significantly to the extent that cloud services could replace the local servers to receive data and display them in real-time to the users. As-a-service provide platforms to the user to use APIs to upload data for information processing such as real-time display, compile data into database and export list to local devices. MQTT provides technologies to transmit data into the internet and allows one or multiple authorized users with correct topic subscribed to received data transmitted by the brokers. VNC is originated as a local remote desktop service similar to RDP for Windows OS. Some companies such as RealVNC develop cloud services on the original VNC to where any client which is configured and is connected to the internet will be able to be reached by the owner regardless of the location and types of networks.

Python is one of the popular programming languages where it can be used in personal small project up to difficult data science projects. With its vast library which can be easily download from PIP, readability from the help of significant indentation, the language can be as modular and as extensible as the user wants it to be. C++ is one of the popular languages which had been around for decades for its fast execution by its compiler language nature. With its fast execution speed, lots of programmers and developers develop libraries and modified functions for the used in microcontroller such as Arduino hardware and such. MicroPython is a language which is developed to be running on bare metal microcontroller hardware which had small resources in around KBs but can be as compatible to the standard Python. MicroPython had its limit such as loss of standard high performance requirement libraries as the language is written to runs on microcontroller tight resources.

2.2 Review of the Existing Systems/Applications

2.2.1 Hommons: Hydroponic Management and Monitoring System for an IOT Based NFT Farm Using Web Technology

In this paper, the researcher's objective is to utilize the unused spaces in the urban area of the country due to conversion of agricultural land into industrial and settlement usage. The project involves the usage of hydroponic techniques to minimize the area required. The Nutrient Film Technique (NFT) is chosen by the researchers due to it usability even when placed vertically to increase the number of crops can be planted at the same time. A microcontroller (Arduino Uno) which is connected to sensors & actuators with esp8266 WiFi module is used to monitor the status of the crops and send the data to the microcomputer (Raspberry Pi 2B) for broadcast into webserver for display. Beside the monitoring system, there are also pre-set monitoring value which have the pH and the PPM of a few different vegetables (Crisnapati, et al., 2017).

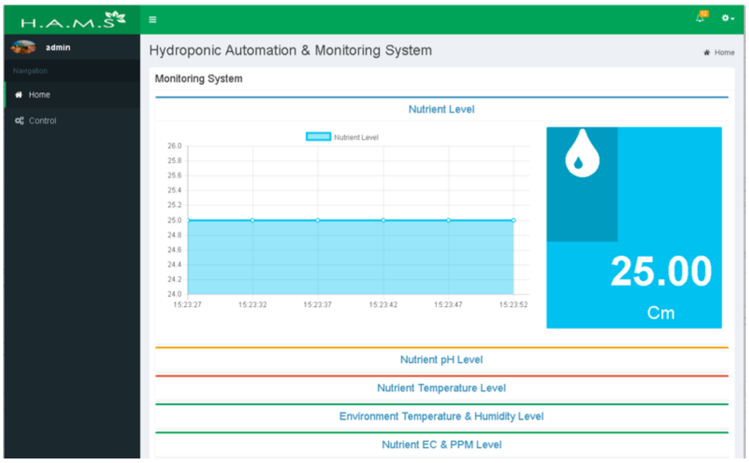
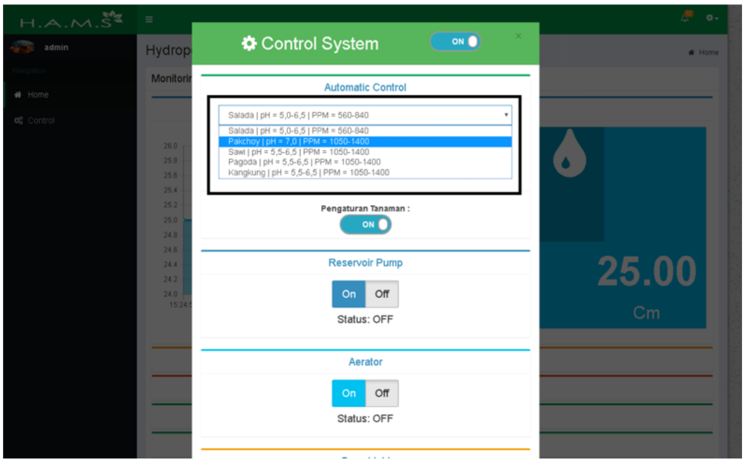


Figure 2.2.1.1 shows the web management page for monitoring.



Salada | pH = 5,0-6,5 | PPM = 560-840
Salada | pH = 5,0-6,5 | PPM = 560-840
Pakchoy | pH = 7,0 | PPM = 1050-1400
Sawi | pH = 5,5-6,5 | PPM = 1050-1400
Pagoda | pH = 5,5-6,5 | PPM = 1050-1400
Kangkung | pH = 5,5-6,5 | PPM = 1050-1400

Status: OFF

Status: OFF

Figure 2.2.1.2 shows the pre-set value for automation monitoring.

The farm draws power from the solar panel to reduce the dependency of the farm to the building it was placed and thus immune to any power outage of surrounding building. Besides, the by setting the farm vertically which hold the growing spot on the top and medium of the rack and the nutrient place, water tub and the reservoir is placed at the bottom of the rack. The water tub and the reservoir are also used as the base to hold the farm rack steadily. By setting up the farm in this method, the limited spaces of urban country can be utilized completely.

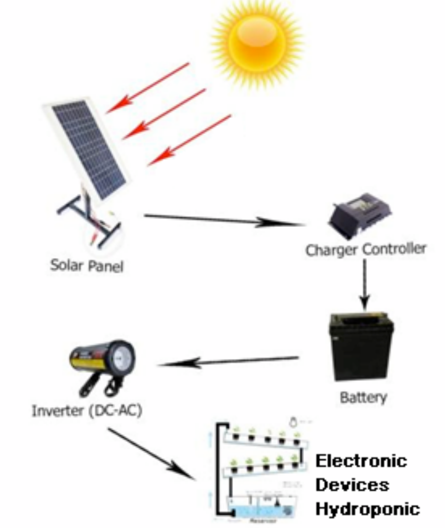


Figure 2.2.1.3 shows the connection from the solar panel to the hydroponic farm.

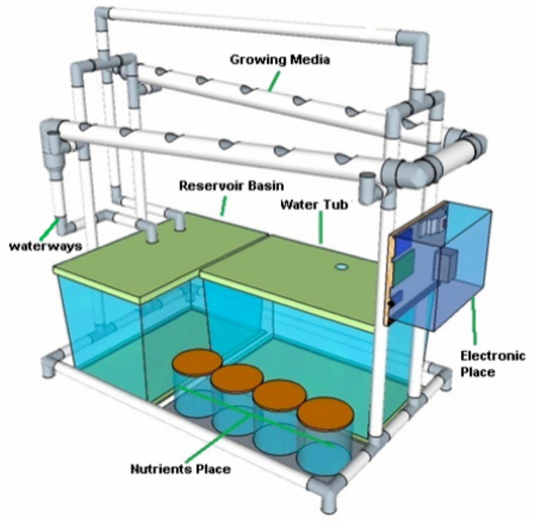


Figure 2.2.1.4 shows the setup of the hydroponic farm.

There are a few limitations on this project as the controls are still being maintained by the conventional human labour. Besides, the monitoring system does not support video feed, the farm does not have any power supply beside the solar panel & the small battery attached around the hydroponic farm and the farm scalability cost will be high due to the dependency to raspberry pi and the solar panel will be needed to purchase to meet the required power supply for the operation of farms.

The video feeds might be able to produce by using camera module connected to either Single Board Computer and added to the web management page. For the power supply, an Uninterruptible Power Supply (UPS) can be used to replace the battery for a long-lasting power supply if the weather is not suitable for power generation.

2.2.2 IOT Hydroponics Management System

In this paper, the researchers proposed an integrated system based on Internet of Things (IoT) devices for hydroponics garden monitoring and management. The integrated system which is placed on a microcomputer (Raspberry PI 3B) will constantly monitor the pH, water level, air temperature and relative humidity while providing automation on irrigation of water and nutrient solution intake. The data collected will be upload to cloud-based database for storage, analysis and sharing to user worldwide (Aliac & Maravillas, 2018).

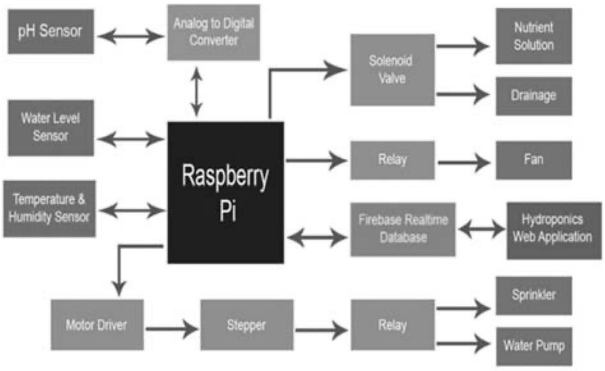
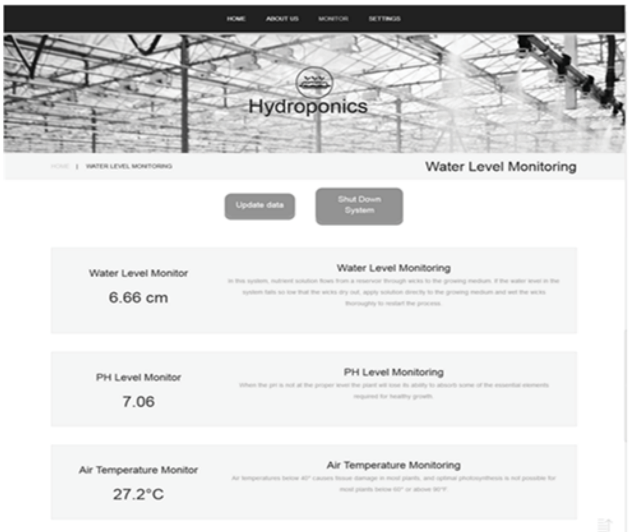


Figure 2.2.2.1 shows the component used in this project.

This project has a better and complete set of monitoring sensors and actors which include temperature & humidity sensor, pH sensor, water level sensor and even have sprinkler and fan to manage the relative humidity and air temperature. The Google firebase is used to store the data transfer from the microcomputer (Raspberry PI 3B) and show the value into a remote-control management and monitoring website. The user can then control the water pump to increase the water level, drainage to remove the water, fan to manage the temperature and sprinkler to manage humidity.



Water Level Monitoring

Water Level Monitor

6.66 cm

Water Level Monitoring

In this system, nutrient solution flows from a reservoir through wicks to the growing medium. If the water level in the system falls so low that the wicks dry out, apply solution directly to the growing medium and wet the wicks thoroughly to restart the process.

PH Level Monitor

7.06

PH Level Monitoring

When the pH is not at the proper level the plant will lose its ability to absorb some of the essential elements required for healthy growth.

Air Temperature Monitor

27.2°C

Air Temperature Monitoring

Air temperatures below 40° causes tissue damage in most plants, and optimal photosynthesis is not possible for most plants below 60° or above 90°F.

Figure 2.2.2.2 shows the setup of the hydroponic farm.

This project had some limitation which is when the people is away from the farm, the user will have to rely on the sensors value to determine the status of the garden. The other problem is that the garden IoT devices relies on the power outlet heavily and any outages will cause the entire garden integrated system to be unavailable. Besides, the garden does not have the ability to perform scalability which limit the size of the garden.

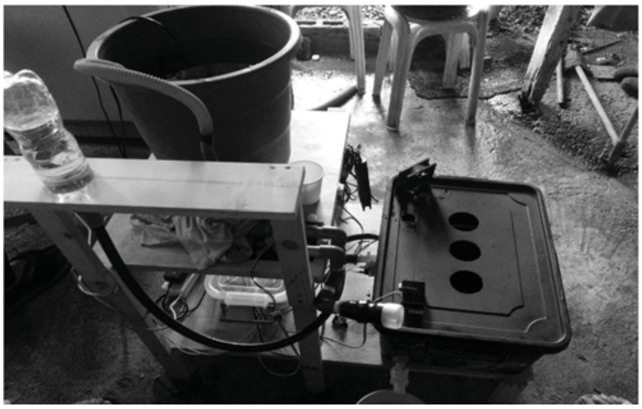


Figure 2.2.2.3 Shows the partial hardware prototype

To overcome the situation, a camera can be fitted at the IoT devices and transmit either video feed or images to the database for viewing. For the power outages preparation, Uninterruptible Power Supply (UPS) can be used to provide backup power to the component which require higher voltage and some batteries can be used to supply power for sensors. The garden integrated systems heavily rely on microcomputer (Raspberry Pi 3B) on the connection of sensors and transferring the data into the firebase database.

2.2.3 Automation of Hydroponics Green House Farming using IOT

In this paper, the researchers utilize the Nutrient Film Technique (NFT) hydroponic method and various IoT devices to create an automated hydroponic green house. The Raspberry Pi is selected as the processing devices to perform the monitoring through temperature sensor, humidity sensor, pH sensor and pressure sensor. Besides, the IoT devices also tasked to transfer the data into cloud and send the value in an easily readable format into the phone application (Saraswathi , et al., 2018).

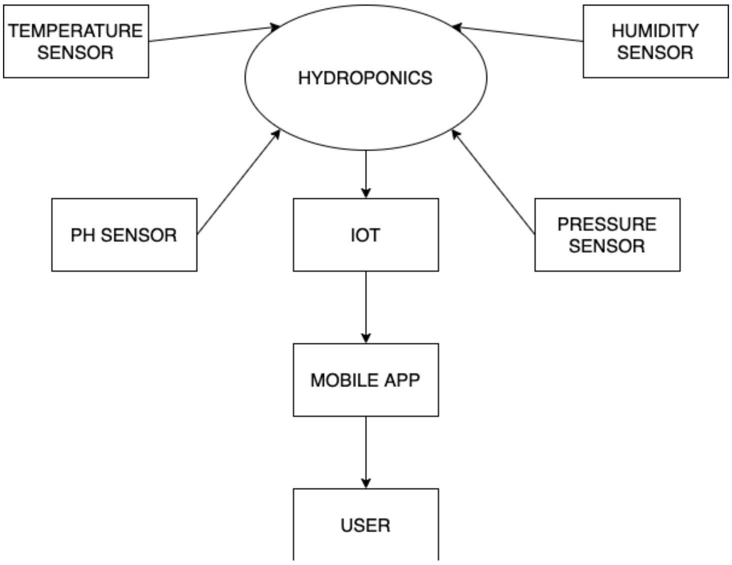


Figure 2.2.3.1 Shows the component used in this project.

The project gave benefits of having a low-cost but sufficient hydroponic farm which serve the purpose of monitoring and supplying the nutrient to the NFT farm without much human interactions. The temperature, humidity, pH and pressure sensors are there to monitor the most sensitive factors which affects the growth of the crops to produce quality yields. The microcomputer also serves as the server to upload the data into cloud at the same time to reduce unnecessary devices wasting around.

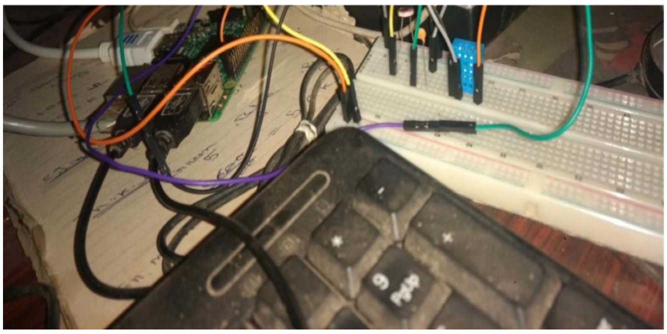


Figure 2.2.3.2 Shows the prototype of the monitoring and supply control system.

The project aims for low-cost approach and thus there are few limits the project might not be able to do. Such as the probability of power outages, monitoring the farm status on graphical feeds and the scalability of the hydroponic farm.

Even though the cost might be too much for the project, an Uninterruptible Power Supply (UPS) and some batteries are recommended in an unexpected event where the power of the location might be gone. For the graphical monitoring, a camera can be added directly to the Raspberry Pi to provide video feed or image to show the status of the farm without being present at the location.

2.2.4 IoT based Hydroponic Farm

In this paper, the researchers intend to solve the issues of uncontrolled environment on traditional farming method. Besides, the factors which downfall the quality of yields such as growth of weeds and the need of spaces increases the difficulty and cost to achieve the desired outcome. The paper introduces the usage of microcontroller to overcome the intensive labour & controlling the environment of the farm. Hydroponic farming, which is independent to weather, light and is soilless is proposed to reduce the maintenance needed (Bakhtar, et al., 2018).

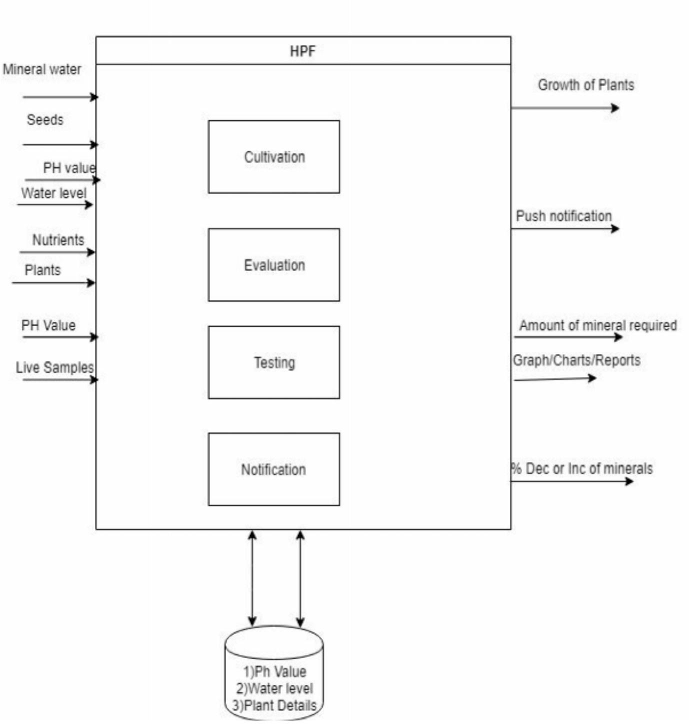


Figure 2.2.4.1 Shows the block diagram of this project.

The smart farm in this project features a vertical approach with various to monitor all sorts of status of the farm from the status of the crops to the nutrient usage. The vertical approach reduces the land spaces needed to produce same number of yields compare to traditional land farming method while the status of the farm is controlled by the microcontroller. The microcontroller controls the temperature and humidity on the environment parts and liquid level in the pipes on the nutrient and supply control.

The project researchers aim to have an approach of having different reports generated on different type of crops planted and the monitoring of pH and nutrient for different type of crops. Pesticides and Insecticides control is also one of the research areas planned to be done in the future.

2.2.5 Internet of things (IoT) applied to an urban garden

In this paper, the researchers divide the entire proposed method in three (3) part, which is Mechanical Subsystem (MS), Electronic Subsystem (ES) and IT Subsystem (ITS). The MS runs on a wheel-motor fitted G-rail with the actuators to enable automation on each station of the urban farm meanwhile the ES consist of sensors to measures the status of the garden such as environment temperature & humidity, CO2, luminosity, soil temperature & humidity and crops proximities. Besides, there are also actuators to perform automated actions such as measure and watering the garden. The main controller to control all of the sensors and actuators are also attached in this subsystem. For the ITS, it took care of the network connectivity to the vast internet through various protocols and communication tools. One of the functions it can provide is enabling the user to monitor the status of the farm through the web and apps (Gabriela, et al., 2018).

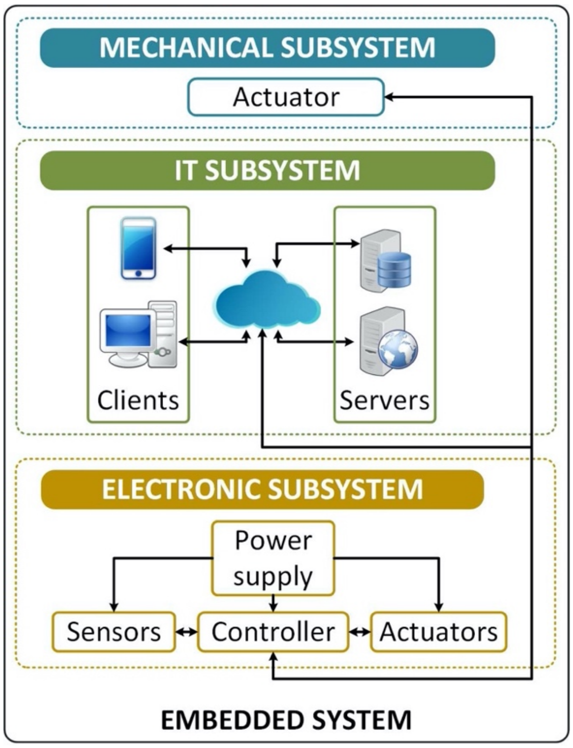


Figure 2.2.5.1 Shows the Physical architecture of the embedded system.

The proposed method is considered a complete and fully automated system as the environment climate, medium conditions and the proximity of the crops is monitored, while the irrigation is automated to irrigate the crops when needed. Besides, there are an app to shows the current conditions of the urban farm to the user without going near the farm.

The limitation of this proposed method is that the user can only know the condition of the farm based on values transfer by the IoT devices back to the database. Besides, the climate condition of the farm is not completely control and is vulnerable to the weathers such as the need of sunlight. Given the need of sunlight by the system, we can also assume that the system power source might be far away and there are no backup power source mentions.

To overcome the limitation mentioned above, there are a few methods which can be used, such as having a camera which will be connect to the microcontroller to send the either images periodically or video feed to the user. For the climate vulnerable, full spectrum led grow light can be used to replace the need for natural light and thus make the environment more controllable. For the power sources issues, backup battery with higher capacity can be used to store the excessive energy and a few solar thermal panels for energy generation if the farm is in outdoor conditions or a special amorphous silicon photovoltaic device can be used to generate energy from indoor light sources.

2.2.6 Summary of Existing Systems/Applications

Existing System	Advantages	Disadvantages	Critical Comments
Solar-powered smart farming management system (Crisnapati, et al., 2017)	- independent power source (solar & UPS) - crops profile preset is available	- required more component to works (adruino with esp8266 & pi 2B)	This project aims to provide monitoring & automated system which depend on solar power & batteries to supply power.
IOT Hydroponics Management System (Aliac & Maravillas, 2018)	- more sensors and actuators are available - data is shared to the public	- does not have live feed to view the farm from other location - hard-coded & hard to customize	This project collect data from numerous sensors & actuators then publish the data to public for easier & efficient management.
Automated IoT NFT monitoring system (Saraswathi , et al., 2018)	- low-cost project - sufficient for basic monitoring	- monitoring only available on mobile	This project aims to provide the most basic monitoring on the lowest cost.
IoT based Hydroponic Farm (Bakhtar, et al., 2018)	- is built for different hydroponic setup in the same framework	- amount of data being monitored is slightly lesser than other system	This study aims to study the framework which suitable for different types of hydroponics.
Internet of things (IoT) applied to an urban garden (Gabriela, et al., 2018)	- data is accessible in chart form - a variety of different sensors available	- possibilities of being affected by weather	This project aims to develop a monitoring & automated system with data stored into database and display to the web app and client app.

Table 2.2.6.1 Shows the summary of the existing system.

CHAPTER 3 SYSTEM METHODOLOGY

3.1 System Development Model

3.1.1 Waterfall System Development Model

Waterfall development model is the simplest and straight forward linear development model. The feature of the Waterfall model is that the process is arrange in stages and only when the current stage is completely completed, the process will move to the next stage. The downside of this development model is that there will be no re-evaluation as the stages which is done will not be modified again. The entire project must be planned according to the requirement before starting the project and will be on strict control without any new ideas to stick into (Shiklo, 2021).

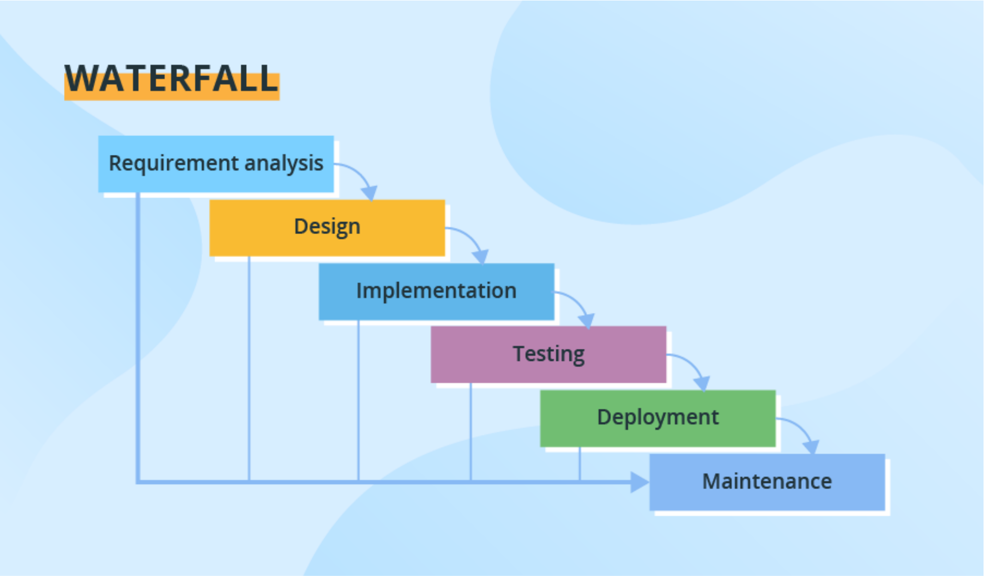


Figure 3.1.1.1 Shows the figure representative of Waterfall development model.

3.1.2 Spiral System Development Model

In the Spiral development model, there are four (4) main activities which is planning, risk analysis, prototyping, and evaluations while the risk assessment is the main focus of the development project. The difference in this development model is that client involvement is high at the project planning, review, and evaluation. This development model is suitable for large, complicated, innovative, and/or research & development projects for it high focus on risk assessment & client involvement (Shiklo, 2021).

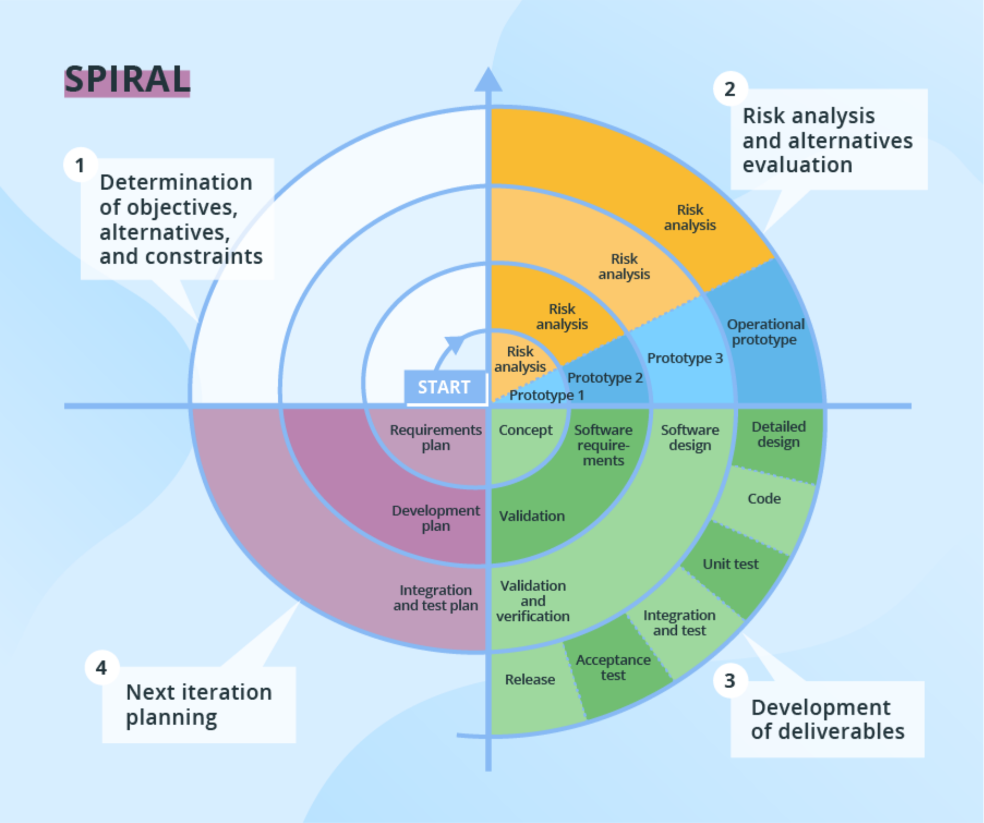


Figure 3.1.2.1 Shows the figure representative of Spiral development model.

3.1.3 Scrum System Development Model

The Scrum development model is similar to the Waterfall development model which require each stage to complete before going into next stages, but the difference in this development model is that the process is breakdown into segments, a sprint, which allow for some planning, development, & a review where the outcome is evaluate before making any amendments and proceed into the second sprint with the similar process (Shiklo, 2021).

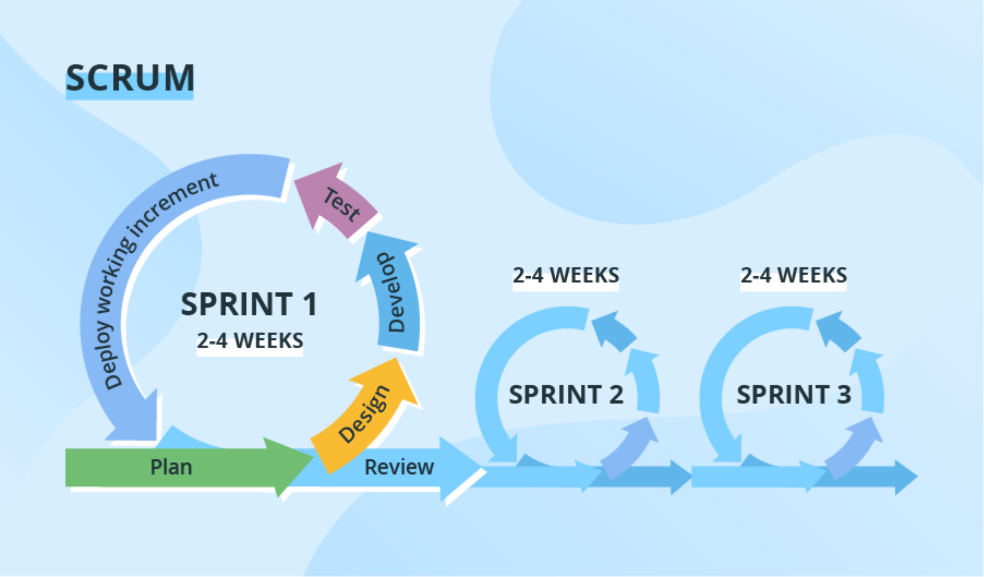


Figure 3.1.3.1 Shows the figure representative of Scrum development model.

3.1.4 Kanban System Development Model

Kanban is an evolutionary development model which differentiate from most of the development model by disposing the stages and breaking down the development into extremely short sprints. The purpose of Kanban development model is to visualize planning & transparency to determine priority of task. Without the presence of iterations stages, the progress can be check at any time and new request can be placed whenever required which encourage enhancement of the output (Shiklo, 2021).

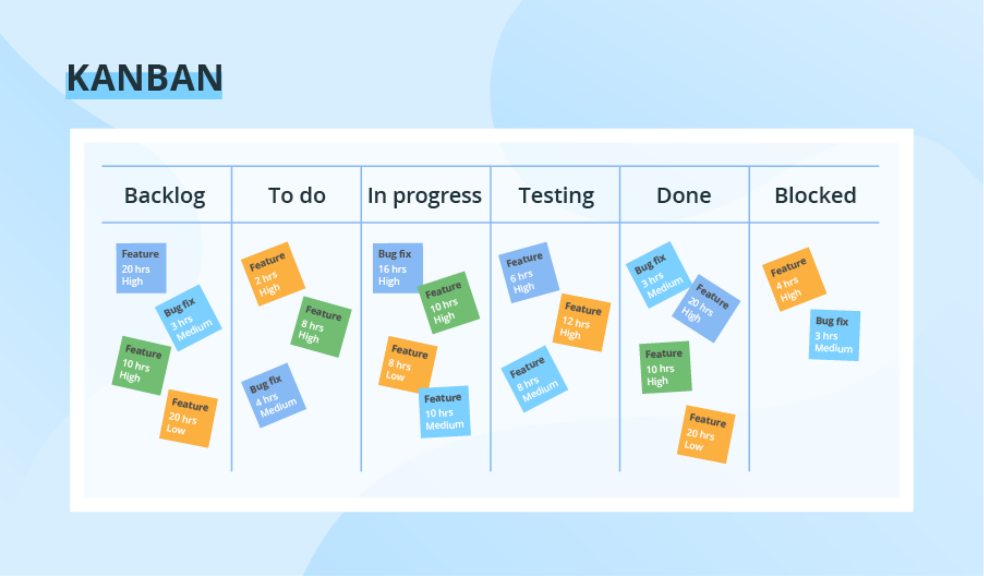
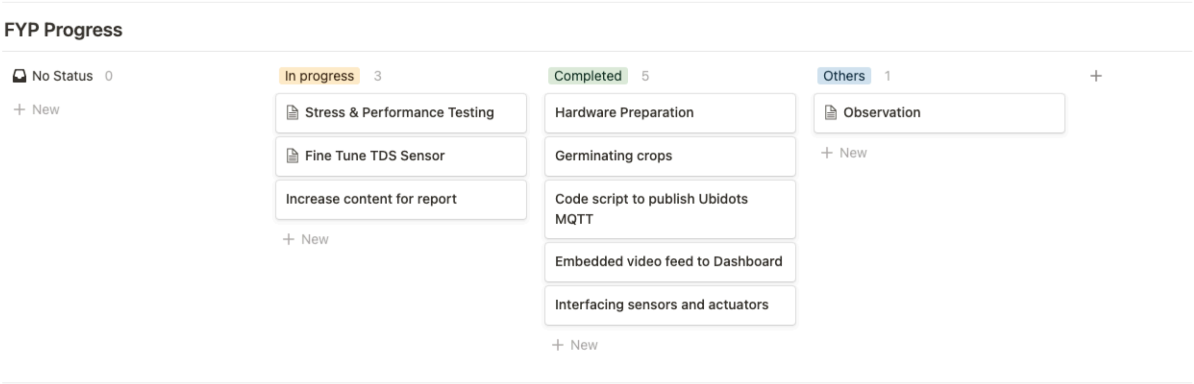


Figure 3.1.4.1 Shows the figure representative of Kanban development model.

3.1.5 Selected Model

The Kanban Development Model is selected for the development of this project due to its planning visualization features. Besides, without the need to differentiate stages which leads to no separate planning stages, new requests or demands can be add in at any time without any issues.



FYP Progress

No Status 0	In progress 3	Completed 5	Others 1
	Stress & Performance Testing	Hardware Preparation	Observation
	Fine Tune TDS Sensor	Germinating crops	
	Increase content for report	Code script to publish Ubidots MQTT	
		Embedded video feed to Dashboard	
		Interfacing sensors and actuators	

Figure 3.1.5.1 Shows the Kanban Board which visualize the task.

3.2 System Requirement

3.2.1 Hardware

Processor	Broadcom BCM2837B0, Cortex-A53 64-bit @ 1.4GHz
Memory (RAM)	1GB Low-Powered DDR2 SDRAM
Storage (ROM)	Micro SD Card (recommended 16 GB to 32 GB)
OS	Raspbian 10 Buster
Connectivity	• 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac WLAN • Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
Onboard port	• Extended 40-pin GPIO header • MIPI CSI camera port

Table 3.2.1.1 Shows the specification of the main controller Raspberry Pi 3B+.

Total Resolution	5 MP	
Charge-coupled device size	¼ inches	
Aperture	f/1.8	
Focal Length	3.6 mm	
Input	3.3V	
Output	Infrared LED module x2	

Table 3.2.1.2 Shows the specification of the 5MP IR Adjustable Pi Camera.

Maximum Current	10A	
Maximum Voltage	250V AC or 30V DC	
Input Voltage	5V	

Table 3.2.1.3 Shows the specification of the 5V 8 channel relay module.

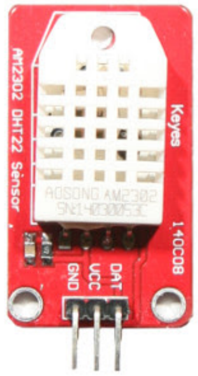
Input Voltage	3.3V – 6V	
Sensing element	Polymer capacitor	
Operating range	Humidity: 0-100%RH Temperature: -40~80 °C	
Accuracy	Humidity: +-5%RH Temperature: <+-0.5 °C	
Long-term Stability	+-0.5%RH/year	

Table 3.2.1.4 Shows the specification of the DHT22 humidity & temperature sensor module.

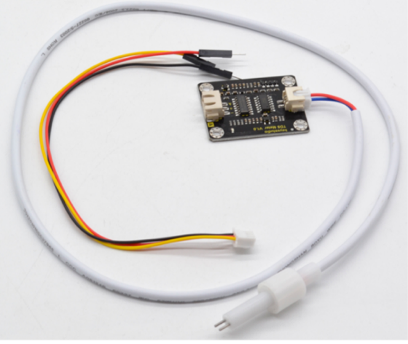
Input Voltage	3.3V – 5.5V	
Output Voltage	0V – 2.3V	
Working Current	3mA – 6mA	
TDS Measurement Range	0ppm – 1000ppm	
TDS Measurement Accuracy	± 10%	
Module Interface	XH2.54-3P	
Electrode Interface	XH2.54-2P	

Table 3.2.1.5 Shows the specification of the Keyestudio TDS sensor kit.

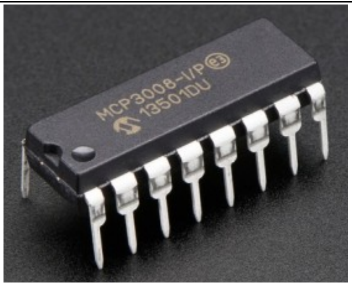
Input Voltage	2.7V – 5.5V	
Max. sampling rate/sec	• 200k @ 5V • 75k @ 2,7V	
Analogue Input(s)	8	
Interface	SPI	

Table 3.2.1.6 Shows the specification of the ADC MCP3008.

3.2.2 Software

Raspberry Pi OS

Raspberry Pi OS is the required operating system which is used by the user to interact with the hardware. The GUI version of the OS is selected for ease of usage within the operation of the system. With the GUI installed, software such as Thorny, which is a code editor, and SenseHat Emu, an emulate sensehat module, can be installed for development purposes. Besides, having a GUI also reduce the need of additional computing devices to develop the system and making modifications on the scripts as the main controller is capable of running a full-fledged OS.

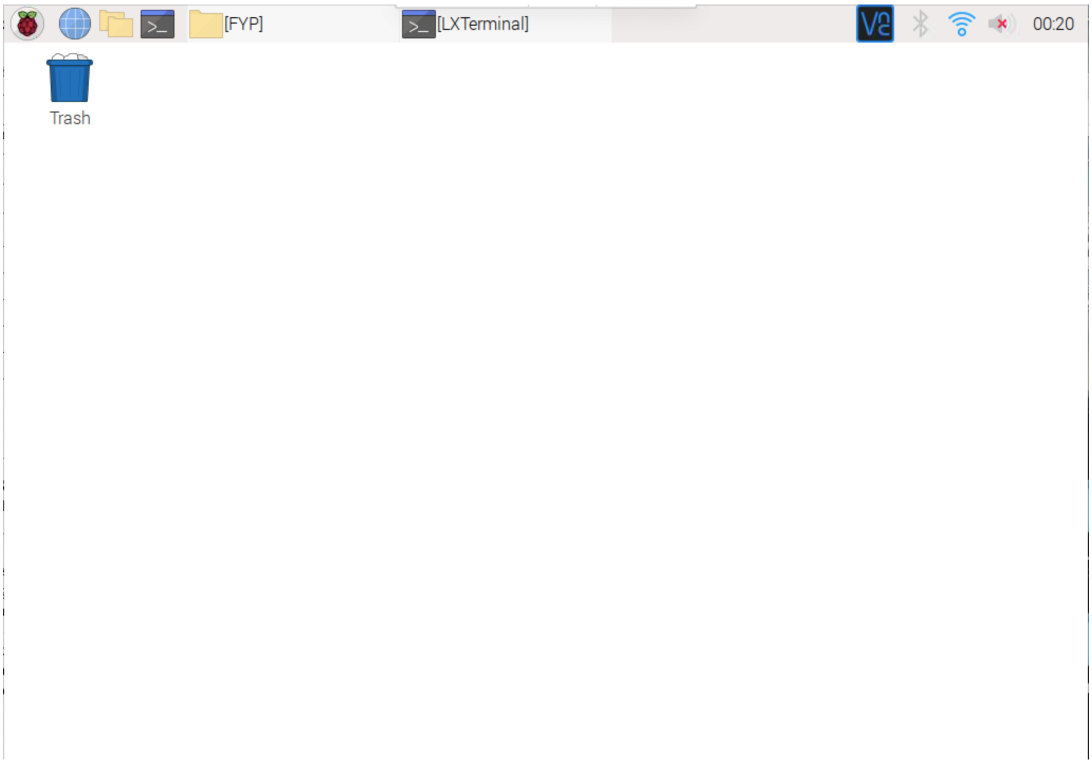


Figure 3.2.2.1 Shows the desktop interface of the Raspberry Pi OS.

Python 3

Python is a multi-function programming language which is versatile in different environment with the help of corresponding libraries developed. Python is also platform independent & OS independent programming language which can be run on different OS and can be edited using the most basic text editor without any problem.

Ubidots MQTT

MQTT is a lightweight message transport protocol which publish the data to broker and allow the client to subscribe to the topic which is published into. In this system, the broker is the cloud provider Ubidots which integrate MQTT into their services to allow connections from different embedded devices into the cloud services.

Ubidots Dashboard (SaaS)

Ubidots dashboard is the Software-as-a-Service service which provided by the Ubidots to serve as a platform to hold and display the data collected from embedded devices to the user.

FFmpeg

FFmpeg is an open-source stream software which stream any video input which is given to the destination. The streaming does not restrict to the local services, but also allow streaming to online platform such as YouTube by stream key provided by YouTube.

YouTube Live (SaaS)

YouTube is considered one of the largest streaming services which allow user to upload their video onto the platform. Since 2008, YouTube provided live streaming services which allow the user to stream video without any duration limitation as long as the devices continue to stream without any disconnection occurs.

RealVNC

VNC is a service where the user can connect and control their devices which are not around them physically through Remote Frame Buffer protocol (RFB). The main purpose of VNC is to connect to faraway devices without getting to the location.

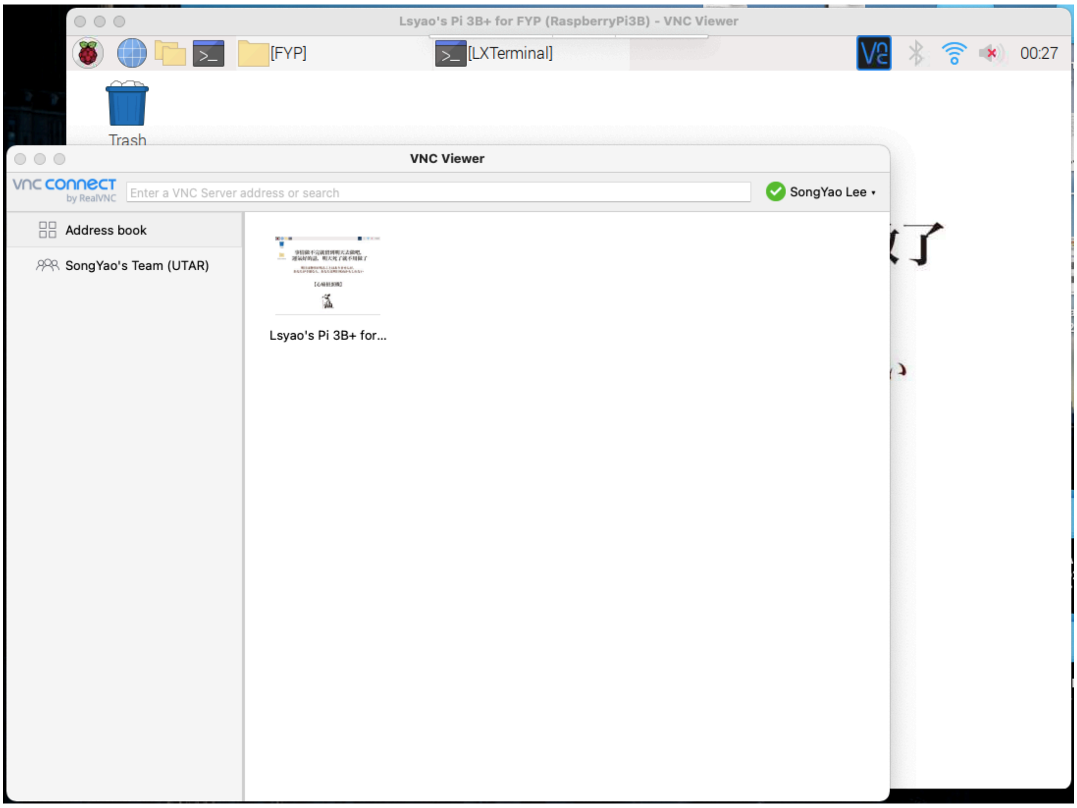


Figure 3.2.2.2 Shows the main controller connected through RealVNC Viewer.

3.3 Functional Requirement

No.	**Description**
FR1	Main controller should collect & publish the data collected from the sensors once every hour.
FR2	Main controller should stream the video feed all the time.
FR3	Main controller should turn the grow light on & off on fixed hour.
FR4	Dashboard should display all data collected including live video feed.
FR5	Dashboard should save all data collected for download.

Table 3.3.1 Shows the functional requirement of the system.

3.4 Project Milestone

Task name	Start date	End date	Progress	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14
FYP 1	**07/06/2021**	**29/08/2021**	100%	■	■	■	■	■	■	■	■	■	■	■	■		
Infrastructure setup & farming preparation	**07/06/2021**	**04/07/2021**	100%	■	■	■	■										
Awaiting material to reach	07/06/2021	19/06/2021	100%	■	■												
Preparing crops seed to germinate	14/06/2021	27/06/2021	100%		■	■											
Set up indoor isolated farming infrastructure	21/06/2021	04/07/2021	100%			■	■										
Attach sensors & actuators to the framework	21/06/2021	04/07/2021	100%			■	■										
Connection & Coding	**05/07/2021**	**01/08/2021**	100%					■	■	■	■						
Connect sensors & actuators to the main controller	05/07/2021	11/07/2021	100%					■									
Code & calibrate script for sensors & actuators	12/07/2021	18/07/2021	100%						■								
Cronjob the script for automated actuators	19/07/2021	25/07/2021	100%							■							
Code the publishing script to Ubidots	26/07/2021	01/08/2021	100%								■						
Prototype 1	**02/08/2021**	**15/08/2021**	100%									■	■				
Stress test the publishing script	02/08/2021	08/08/2021	100%									■					
Fine tuning sensors calibration	09/08/2021	15/08/2021	100%										■				
Prototype 2	**16/08/2021**	**29/08/2021**	100%											■	■		
Stress test the fine-tuned publishing script	16/08/2021	29/08/2021	100%											■	■		
Fine tune for whenever needed	16/08/2021	29/08/2021	100%											■	■		

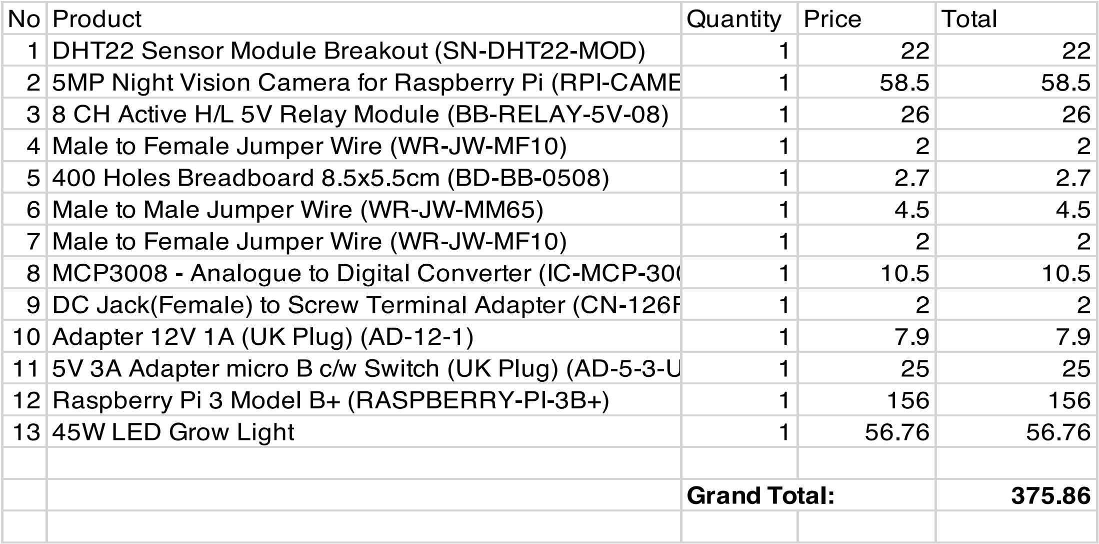
Table 3.4.1 Shows the gantt chart of the system development for FYP 1.

Task name	Start date	End date	Progress	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14
FYP 2	**18/10/2021**	**05/12/2021**	100%	■	■	■	■	■	■	■							
First Batch of farming	**18/10/2021**	**21/11/2021**	100%	■	■	■	■	■									
Germination period	18/10/2021	24/10/2021	100%	■													
Plantation period	25/10/2021	21/11/2021	100%		■	■	■	■									
Backup Batch for farming	**01/11/2021**	**05/12/2021**	100%			■	■	■	■	■							
Germination period	01/11/2021	07/11/2021	100%			■											
Plantation period	08/11/2021	05/12/2021	100%				■	■	■	■							
Report & Presentation preparation	**08/11/2021**	**28/11/2021**	100%				■	■	■								
Adding content & amendment	08/11/2021	28/11/2021	100%				■	■	■								

Table 3.4.2 Shows the gantt chart of the system development for FYP 2.

3.5 Estimated Cost

The following is the items which are hard to purchase from the local market as most of them are electronics which have specific purposes and are unusually seen in daily life. The rest of the items which are not listed in are items which can be salvaged from broken daily appliances or can be source cheaply locally. The cost of the items for commercialisation are based on the cost of items with corporate discount which are not known & is always depends on case-by-case scenario.



No	Product	Quantity	Price	Total
1	DHT22 Sensor Module Breakout (SN-DHT22-MOD)	1	22	22
2	5MP Night Vision Camera for Raspberry Pi (RPI-CAME	1	58.5	58.5
3	8 CH Active H/L 5V Relay Module (BB-RELAY-5V-08)	1	26	26
4	Male to Female Jumper Wire (WR-JW-MF10)	1	2	2
5	400 Holes Breadboard 8.5x5.5cm (BD-BB-0508)	1	2.7	2.7
6	Male to Male Jumper Wire (WR-JW-MM65)	1	4.5	4.5
7	Male to Female Jumper Wire (WR-JW-MF10)	1	2	2
8	MCP3008 - Analogue to Digital Converter (IC-MCP-30	1	10.5	10.5
9	DC Jack(Female) to Screw Terminal Adapter (CN-126F	1	2	2
10	Adapter 12V 1A (UK Plug) (AD-12-1)	1	7.9	7.9
11	5V 3A Adapter micro B c/w Switch (UK Plug) (AD-5-3-U	1	25	25
12	Raspberry Pi 3 Model B+ (RASPBERRY-PI-3B+)	1	156	156
13	45W LED Grow Light	1	56.76	56.76
		Grand Total:		**375.86**

Figure 3.5.1 Shows the estimated cost to setup the system.

3.6 Concluding Remark

This chapter gives an overview of the technology which is being used to make the system a reality. The system development model is discuss first followed by the chosen system development model which is use in this project. The system requirement on the hardware and software is then listed followed by the functional requirements, project milestones, & the estimated cost.

CHAPTER 4 SYSTEM DESIGN

4.1 System Architecture

In this project, the temperature & humidity is handled by DHT22 breakout sensor module, the TDS value is handled by the TDS sensor connected to an analogue converter MCP3008, and the actuator are handled by an 8-channel relay module. There are also a Pi Camera to show a feed of the indoor hydroponic farm from the inside.

Python is selected to use as the scripting language for this project for its ease to code properties. Besides, python libraries from Adafruit, which is used for DHT22 & MCP3008 is used in this project.

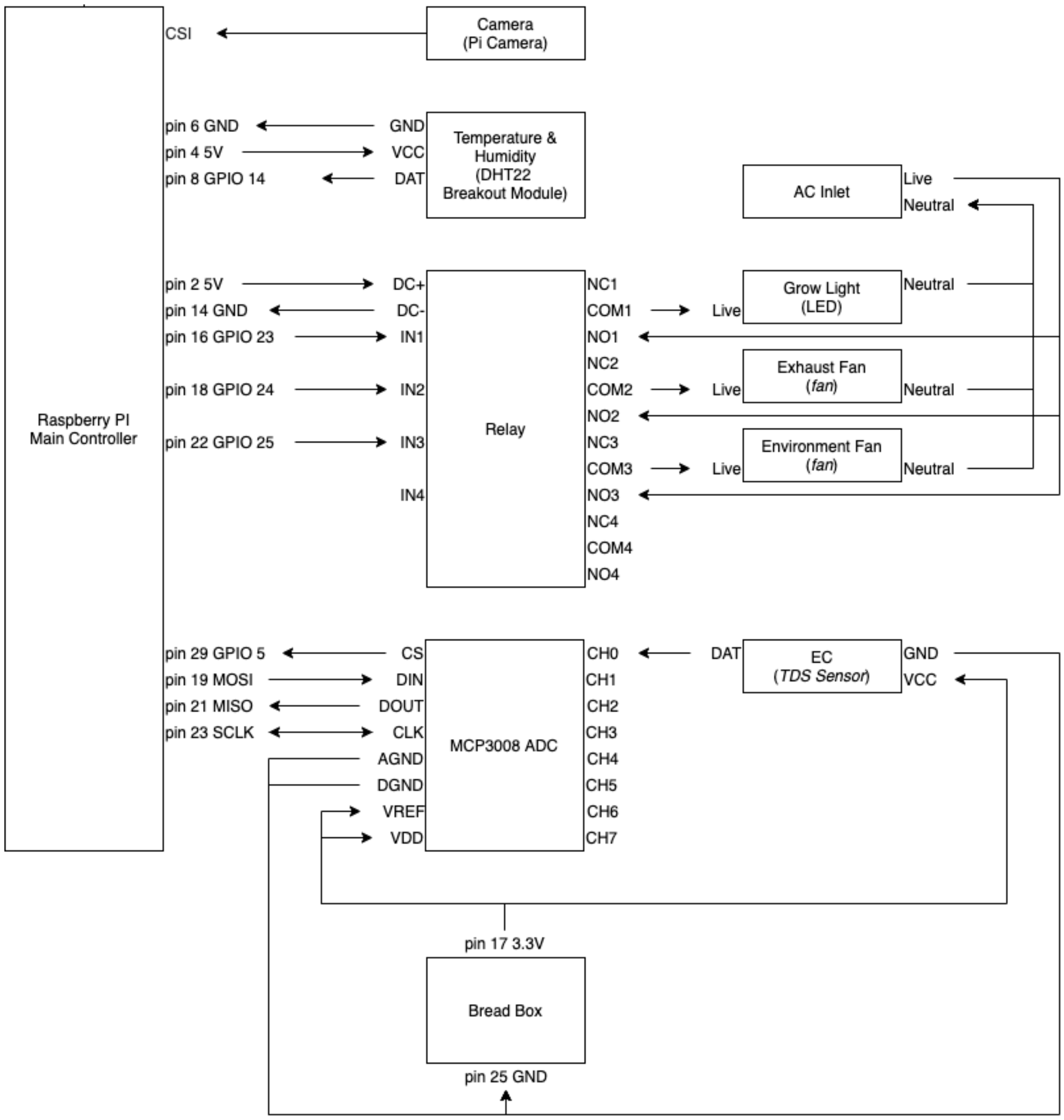


Figure 4.1.1 Shows the hardware arrangement and connection.

4.2 Functional Modules in the System

- Module 1 (Data Collection):
    - This module will receive the value read from the sensors and transfer to the SaaS Ubidots Dashboard through Ubidots MQTT API.
- Module 2 (Live Stream):
    - Retrieve the video feed using *raspivid* command and stream the video feed to YouTube Live using FFmpeg. To ensure the live feed reach YouTube, the YouTube stream key is used to specify the destination.
- Module 3 (Automation Control):
    - The grow light is set to be turn on for a specific amount of time to simulate the sunlight. Crontab is used to turn the grow light which attached through the 8-channel relay module on & off at specified time which is 7 am and 10 pm.
- Module 4 (Display on Dashboard):
    - The SaaS Ubidots Dashboard will receive the data from the main controller through Ubidots MQTT API and display the value on the Dashboard. The live feed will be embedded on the blank HTML canvas using YouTube embedded functions.
- Module 5 (Data Download for Backup):
    - All data will be kept on the Ubidots databases and is available for download at any moment.

4.3 System Flow

The main controller, a Raspberry Pi 3B+, will retrieve the data from the sensors, temperature & humidity from DHT22, and TDS from TDS sensor connected to MCP3008.

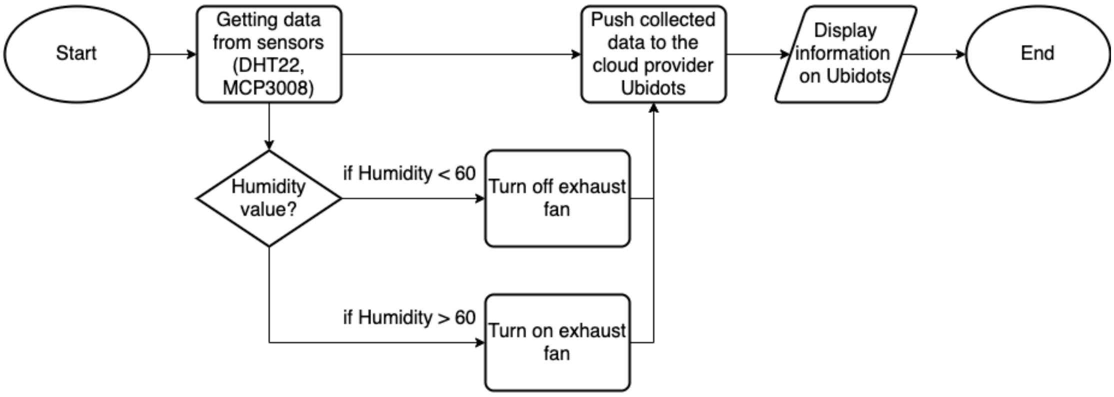

Start
Getting data from sensors (DHT22, MCP3008)
Push collected data to the cloud provider Ubidots
Display information on Ubidots
End
Humidity value?
if Humidity < 60
Turn off exhaust fan
if Humidity > 60
Turn on exhaust fan

Figure 4.3.1 Shows the system data publishing flowchart.

The main controller will then publish the data to the cloud provider to allow the user to understand the status of the farm on real-time.

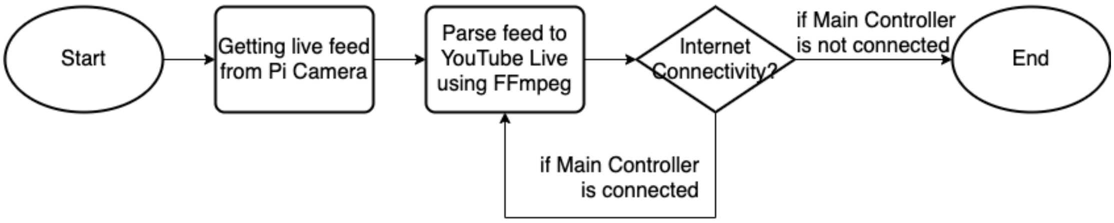

Start
Getting live feed from Pi Camera
Parse feed to YouTube Live using FFmpeg
Internet Connectivity?
if Main Controller is not connected
End
if Main Controller is connected

Figure 4.3.2 Shows the live feed flowchart.

For the grow light, an automation is setup using crontab which is featured in almost all GNU/Linux OSes. Time for the grow light to turn on & off is fixed to simulate the sunlight duration at the field.

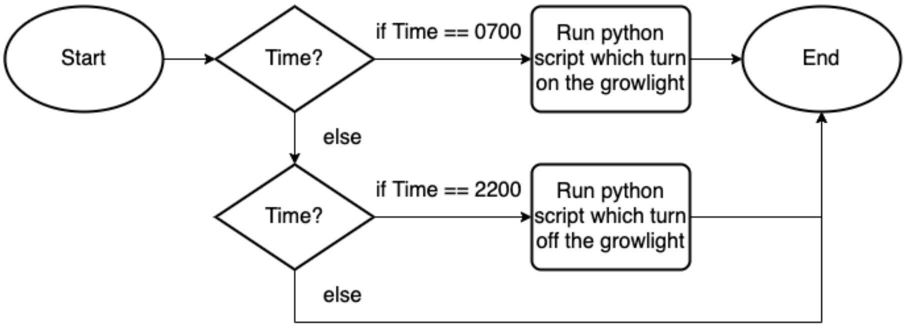

Start
Time?
if Time == 0700
Run python script which turn on the growlight
End
else
Time?
if Time == 2200
Run python script which turn off the growlight
else

Figure 4.3.3 Shows the automated grow light flowchart using Crontab.

4.4 GUI Design

The only GUI which will be presented is the SaaS Ubidots Dashboard which is used to hold the widget for data display and the live feed. The widget are resize and stack to each other to have an organized view.

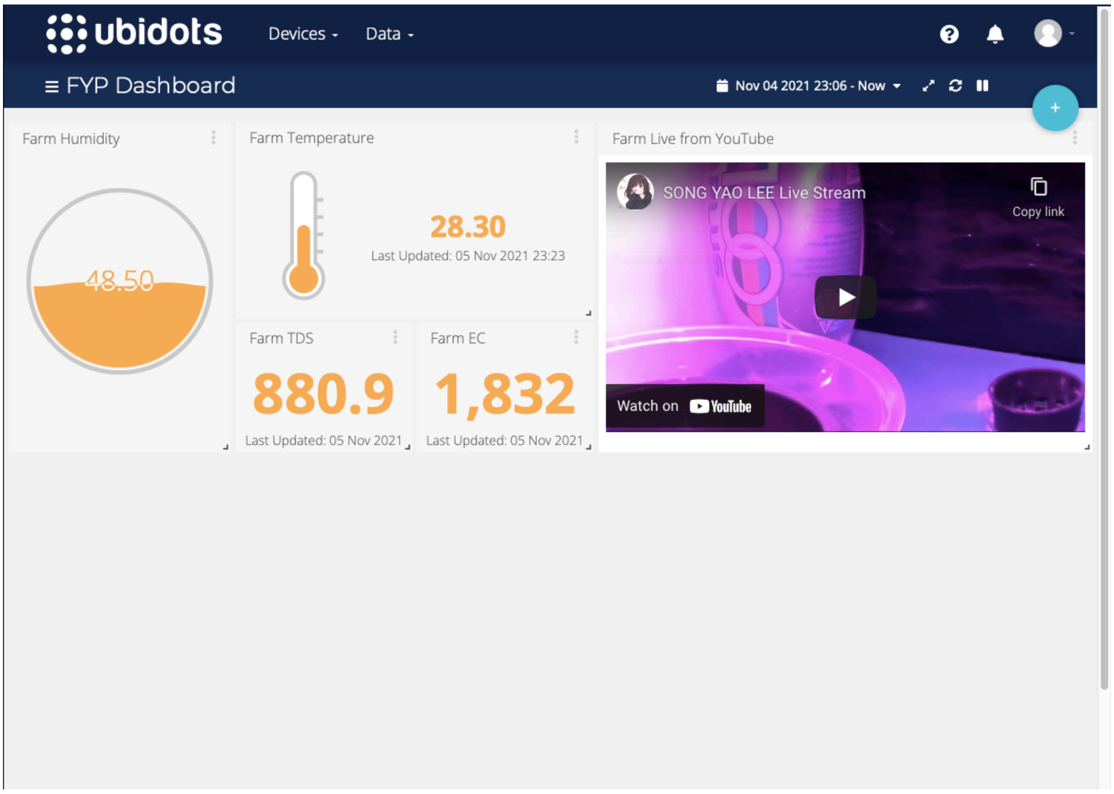


Figure 4.4.1 Shows the Ubidots Dashboard.

4.5 Concluding Remark

This chapter discuss about the technical design of the system which is developed in this project. The architecture is first listed with the items which will be used in detail, then follow by the functional modules which define the function of the system. The system flow is then listed by using flowcharts represent the actions taken by the system in its operation. The GUI Design of the Dashboard is then discussed as the ending of this chapter.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Hardware Setup

In this system, most of the connections used are mainly focus on prototyping which reduce the need of soldering to achieve modularity in case any of the parts actually fail or does not function properly.

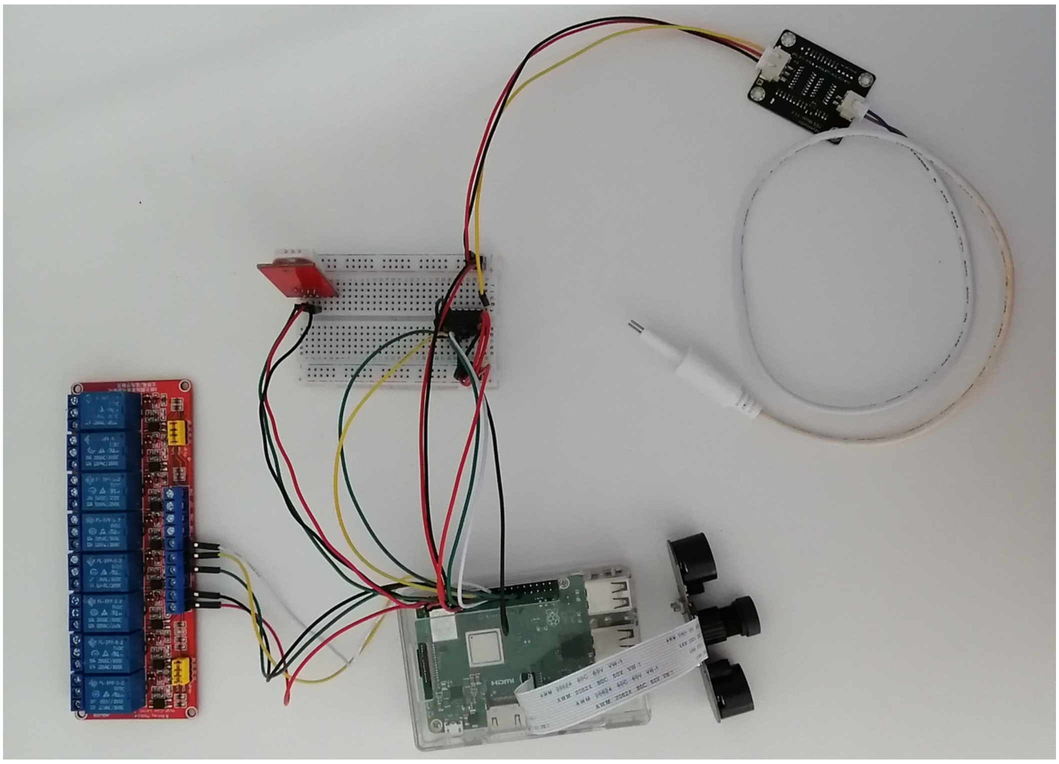


Figure 5.1.1 Shows the connection from the main controller Raspberry Pi header to breadbox.

The main controller Raspberry Pi 3B+ read value from the sensors & controls the actuators using its header pins. The sensors are concentrated to the breadboard for easier management while the actuators are connected to the main controller through the a 8-way relay module. The Pi Camera is connected to the CSI port which are provided on the Raspberry Pi 3B+. The sensors which is attached are the:

- DHT22
- MCP3008
- TDS Sensor

The DHT22 is a multipurpose sensor which can read the temperature and the humidity at the same time with the help of Adafruit python library and is able to transmit the data through digital pins. The pin 4 (5V), pin 6 (GND), and pin 8 (GPIO 14) is chosen to connect the DHT22 to the main controller through the breadbox.

The MCP3008 is an ADC which is used to convert analog reader to digital value as the Raspberry Pi does not have any analog pins. In this system, Adafruit circuitpython library is used to ensure that the reading is consistent and accurate. MCP3008 require four (4) output data pins, two (2) power pins, and two (2) ground pins. As the main controller does not have much power & ground pins available, the breadboard is used to multiply the available power & ground slots through the vertically placed slots at the outer layer. The SPI is used for the MCP3008 which took the pin 19 (MOSI), pin 21 (MISO), pin 23 (SCLK), and pin 29 (GPIO 5).

The TDS sensor is then connected through the MCP3008 to main controller. As the TDS sensor is an analog sensor, the MCP3008 will read the value transmitted from the TDS sensor and convert to digital value before transmitting to the main controller.

The Pi camera which connected through the CSI will be registered automatically and can be used without any additional setup.

For the actuators, there are slightly more part which need to be done as the actuators are mostly work independently on the electric socket thus modification are required to be done in order to connect the actuators to the relay module. The relay works by cutting the live wire of the actuators and place the wire connected to the power supply at the NO (Normally Open) ports and the wire to the actuators at the COM (Common Contact) port. The pin required relies on the amount of actuators with addition which is power (5V) and ground (GND). In this system, three (3) actuators are used thus the pin 16 (GPIO 23), pin 18 (GPIO 24), and pin 22 (GPIO 25). The relay module can be control without the need of additional python library.

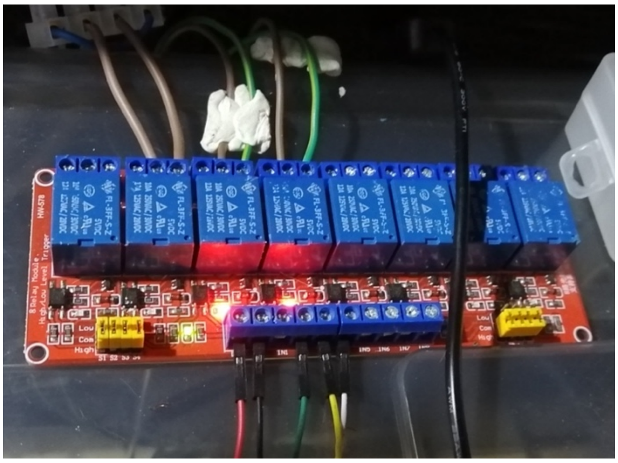


Figure 5.1.2 Shows the grow light & fans connected to the relay module.

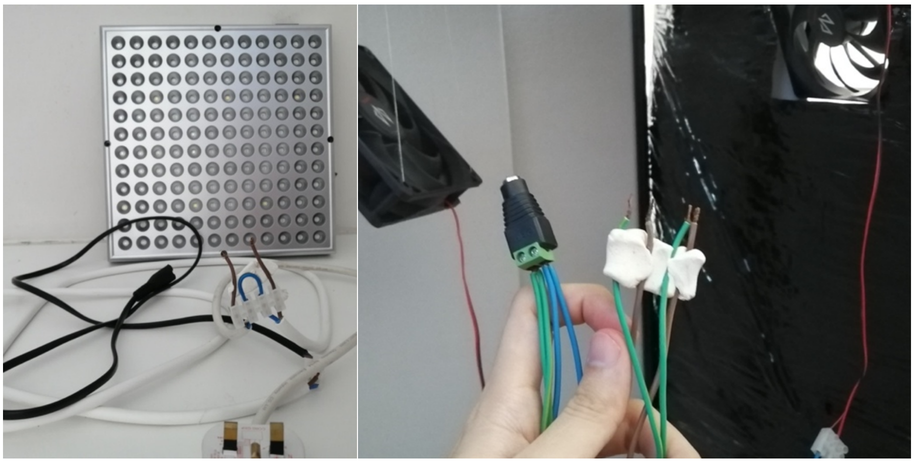


Figure 5.1.3 Shows the grow light & fans wire which are modified.

5.2 Software Setup

The setup of the software is separated into three sections:

- OS
- Dependencies
- Library

The Operating System needed to be flashed onto SD card which is going to hold the OS and any modification made. The easier method can be done by using the Raspberry Pi Imager provided by the Raspberry Pi foundation. Just select the OS & the SD card in the menu and press write to flash the OS onto the SD card.

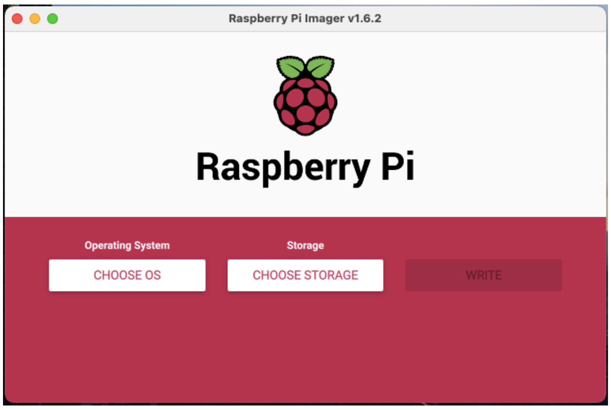


Figure 5.2.1 Shows the menu of Raspberry Pi Imager.

For the dependencies, ffmpeg, python3 and pip3 will be the main focus to ensure the system can work correctly. Two (2) commands are required to install the required dependencies which will allow easier installation of libraries:

```
sudo apt-get install python3-dev python3-pip ffmpeg
sudo python3 -m pip install --upgrade pip setuptools wheel
```

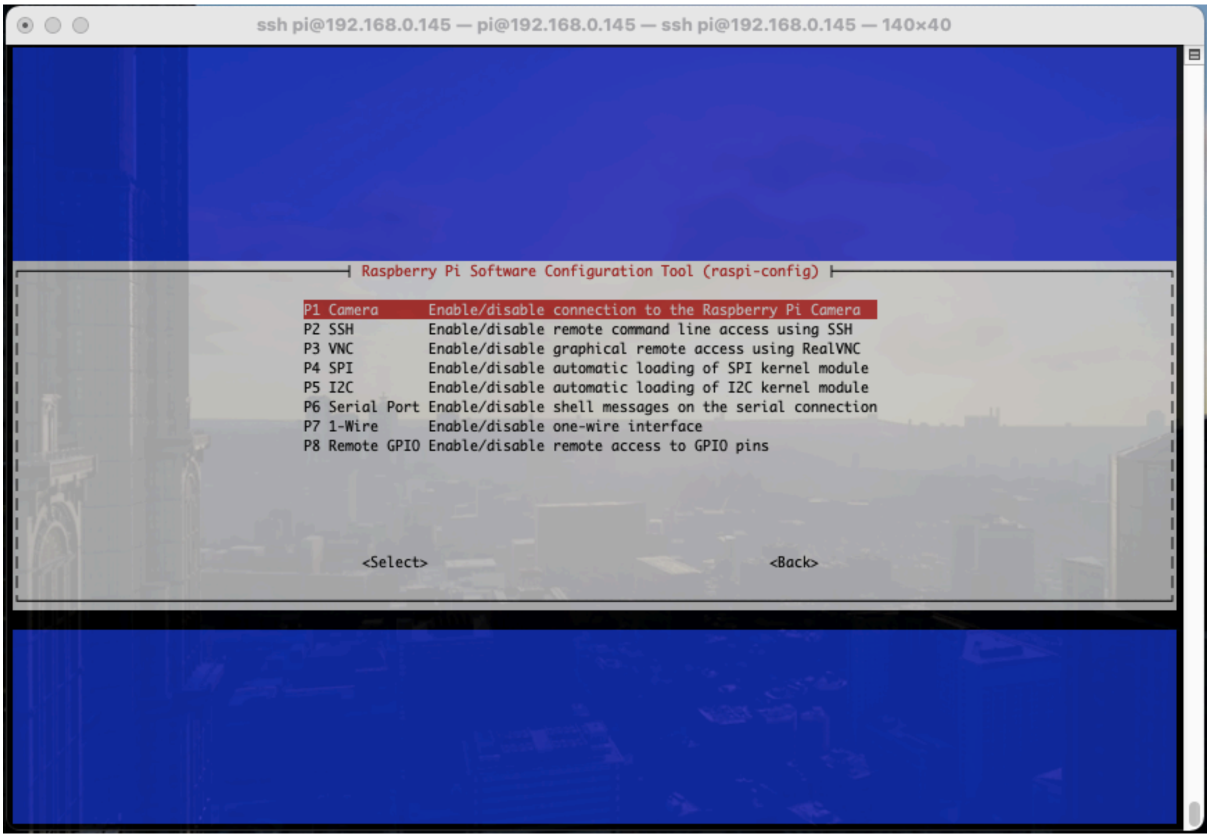
With the pip3 on the system, vast python library can be installed with just a short command. In this system, adafruit libraries are used for two of the sensors which is the temperature/humidity sensor}}}}} DHT22 and the ADC MCP3008. To install the libraries, just search within *https://pypi.org* for library required. Two (2) library which is used in this system is listed:

```
sudo pip3 install Adafruit_DHT
sudo pip3 install adafruit-circuitpython-mcp3xxx
```

5.3 Setting and Configuration

After configuring the OS and the dependencies, the Raspberry Pi OS require manual enable of interfaces for security purposes. The interfaces can be easily enabled in the *raspi-config* menu. There are four (4) interface which is required:

- Camera
- SSH
- VNC
- SPI



ssh pi@192.168.0.145 — pi@192.168.0.145 — ssh pi@192.168.0.145 — 140×40

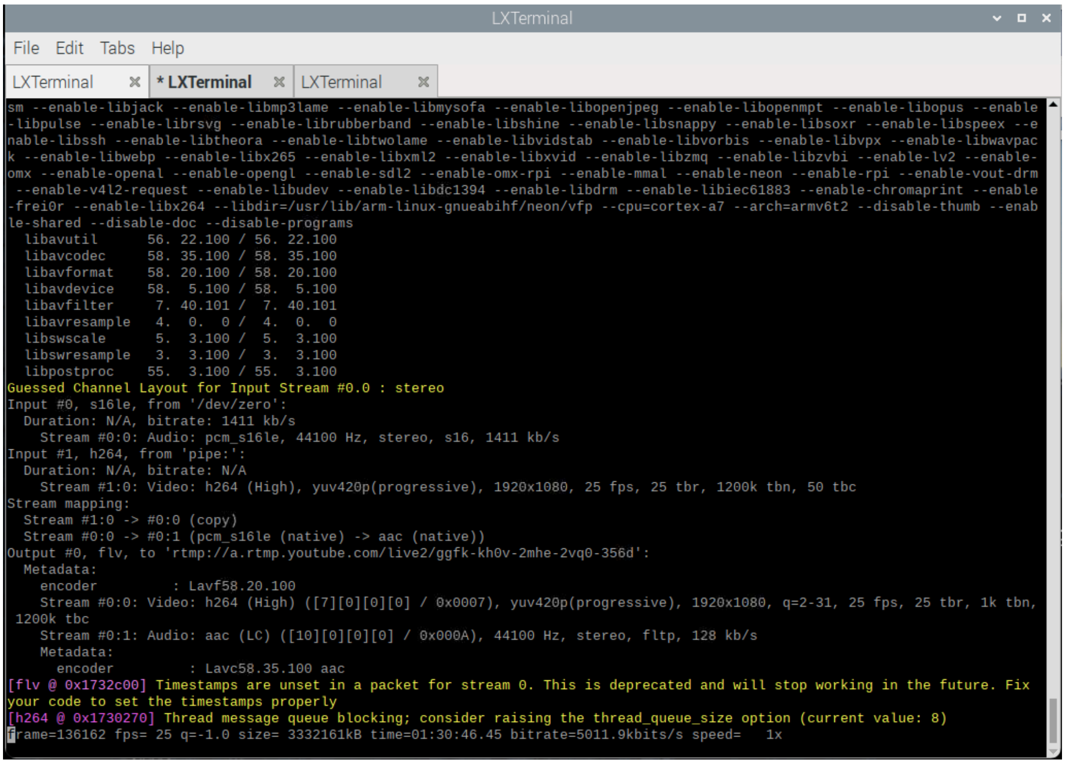
```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins


       <Select>                                   <Back>
```

Figure 5.3.1 Shows the raspi-config menu through SSH.

The camera needed to be enabled so that the CSI is available for live streaming usage. SSH is enabled when there are needed for some simple modification from mobile devices. The VNC is enabled so that the Pi can be placed as headless while maintaining the availability of the desktop for using GUI application. SPI is enabled so that the ADC MCP3008 can be used.

5.4 System Operation

The system is run on a command which capture the live feed & stream through FFMPEG to YouTube, a python script which transmit data collected from sensor to Ubidots Dashboard through MQTT, and a Crontab which automated the grow light.



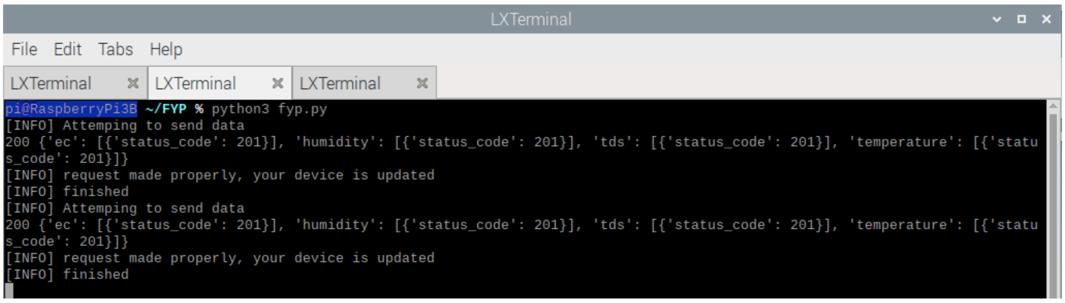
```
sm --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable
-libpulse --enable-librsvg --enable-librubberband --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --e
nable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpac
k --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable-
omx --enable-openal --enable-opengl --enable-sdl2 --enable-omx-rpi --enable-mmal --enable-neon --enable-rpi --enable-vout-drm
 --enable-v4l2-request --enable-libudev --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --enable
-frei0r --enable-libx264 --libdir=/usr/lib/arm-linux-gnueabihf/neon/vfp --cpu=cortex-a7 --arch=armv6t2 --disable-thumb --enab
le-shared --disable-doc --disable-programs
  libavutil      56. 22.100 / 56. 22.100
  libavcodec     58. 35.100 / 58. 35.100
  libavformat    58. 20.100 / 58. 20.100
  libavdevice    58.  5.100 / 58.  5.100
  libavfilter     7. 40.101 /  7. 40.101
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  3.100 /  5.  3.100
  libswresample   3.  3.100 /  3.  3.100
  libpostproc    55.  3.100 / 55.  3.100
Guessed Channel Layout for Input Stream #0.0 : stereo
Input #0, s16le, from '/dev/zero':
  Duration: N/A, bitrate: 1411 kb/s
    Stream #0:0: Audio: pcm_s16le, 44100 Hz, stereo, s16, 1411 kb/s
Input #1, h264, from 'pipe:':
  Duration: N/A, bitrate: N/A
    Stream #1:0: Video: h264 (High), yuv420p(progressive), 1920x1080, 25 fps, 25 tbr, 1200k tbn, 50 tbc
Stream mapping:
  Stream #1:0 -> #0:0 (copy)
  Stream #0:0 -> #0:1 (pcm_s16le (native) -> aac (native))
Output #0, flv, to 'rtmp://a.rtmp.youtube.com/live2/ggfk-kh0v-2mhe-2vq0-356d':
  Metadata:
    encoder         : Lavf58.20.100
    Stream #0:0: Video: h264 (High) ([7][0][0][0] / 0x0007), yuv420p(progressive), 1920x1080, q=2-31, 25 fps, 25 tbr, 1k tbn,
 1200k tbc
    Stream #0:1: Audio: aac (LC) ([10][0][0][0] / 0x000A), 44100 Hz, stereo, fltp, 128 kb/s
    Metadata:
      encoder         : Lavc58.35.100 aac
[flv @ 0x1732c00] Timestamps are unset in a packet for stream 0. This is deprecated and will stop working in the future. Fix
your code to set the timestamps properly
[h264 @ 0x1730270] Thread message queue blocking; consider raising the thread_queue_size option (current value: 8)
frame=136162 fps= 25 q=-1.0 size= 3332161kB time=01:30:46.45 bitrate=5011.9kbits/s speed=   1x
```

Figure 5.4.1 Shows the command to stream live feed to youtube live through FFMPEG.

The command which is used to stream live feed is a combination of two piece of software, which is *raspivid* and *ffmpeg*. the raspivid will set the video parameter & the orientation of video while ffmpeg will receive the video to stream through rtmp.

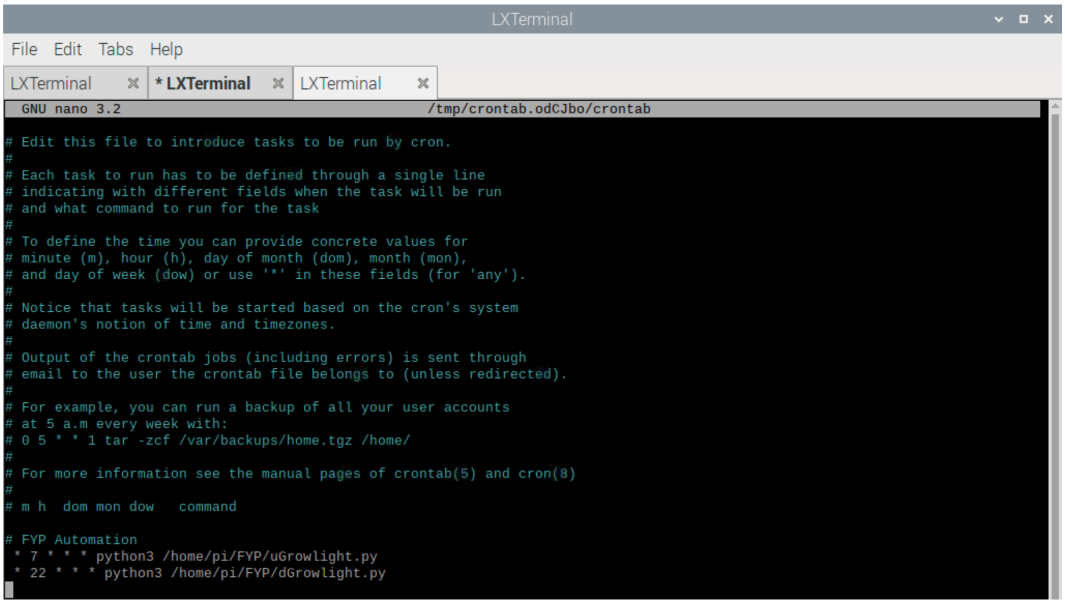
```
raspivid -o - -t 0 -vf -hf -fps 30 -b 6000000 | ffmpeg -re -ar
44100 -ac 2 -acodec pcm_s16le -f s16le -ac 2 -i /dev/zero -f h264
-i - -vcodec copy -acodec aac -ab 128k -g 50 -strict experimental
-f flv rtmp://a.rtmp.youtube.com/live2/stream-key-here
```

The raspivid parameters which need to be taken note is the *-vf* & *-hf* which flip the output vertically horizontally as the camera is placed upside down while the -fps limit the max fps to transmit. In the FFmpeg, the *-re* allow to read the video feed at the native resolution, the *-f h264* which force the format to be in MP4, and the *rtmp://a.rtmp.youtube.com/live2/ggfk-kh0v-2mhe-2vq0-356d* which use the stream key at the end of the URL to point the video feed to the destination YouTube Live. The rest of the command is used for audio transmission which is non-existence in this project but still required to transmit to YouTube Live.



```
pi@RaspberryPi3B ~/FYP % python3 fyp.py
[INFO] Attemping to send data
200 {'ec': [{'status_code': 201}], 'humidity': [{'status_code': 201}], 'tds': [{'status_code': 201}], 'temperature': [{'statu
s_code': 201}]}
[INFO] request made properly, your device is updated
[INFO] finished
[INFO] Attemping to send data
200 {'ec': [{'status_code': 201}], 'humidity': [{'status_code': 201}], 'tds': [{'status_code': 201}], 'temperature': [{'statu
s_code': 201}]}
[INFO] request made properly, your device is updated
[INFO] finished
```

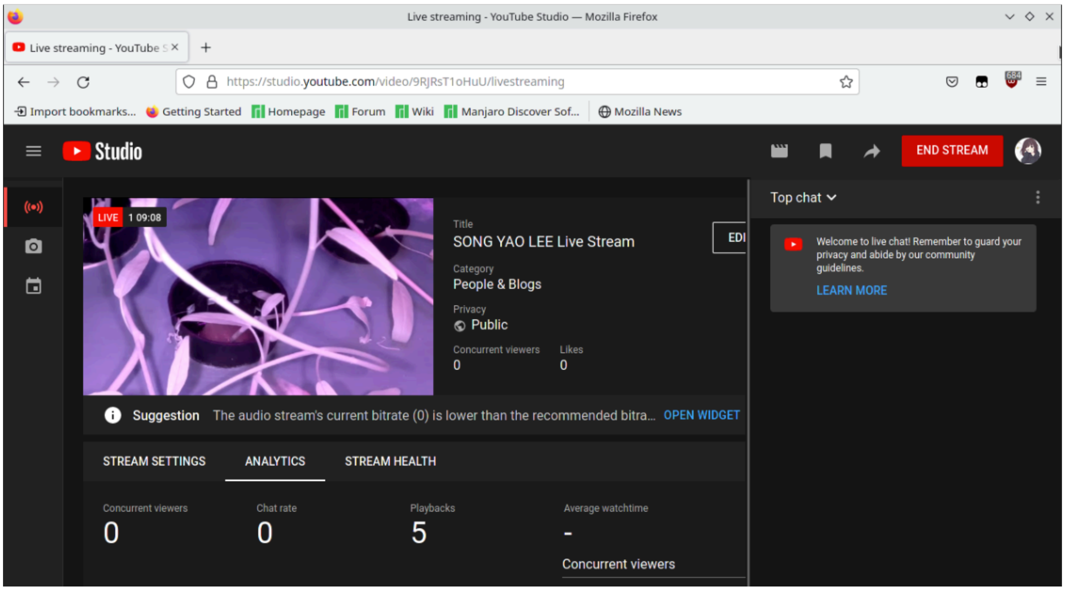
Figure 5.4.2 Shows the process of the Ubidots MQTT python script is executed.



```
  GNU nano 3.2                              /tmp/crontab.odCJbo/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

# FYP Automation
 * 7 * * * python3 /home/pi/FYP/uGrowlight.py
 * 22 * * * python3 /home/pi/FYP/dGrowlight.py
```

Figure 5.4.3 Shows the crontab for the growlight automation.

For the live feed, youtube is chosen as the processing and storing hub to display on the Ubisots dashboard.



Title
SONG YAO LEE Live Stream

Category
People & Blogs

Privacy
Public

Concurrent viewers 0
Likes 0

Suggestion The audio stream's current bitrate (0) is lower than the recommended bitra...

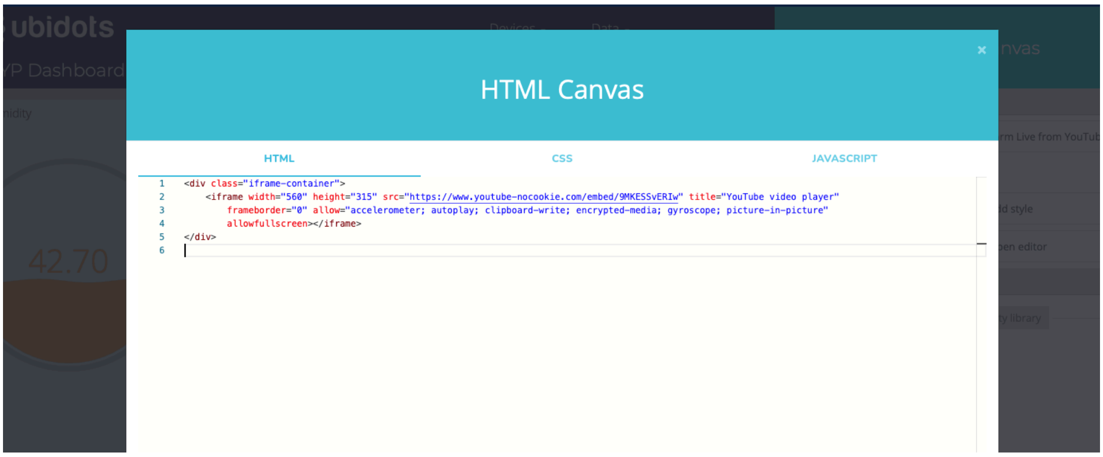
Welcome to live chat! Remember to guard your privacy and abide by our community guidelines.

Concurrent viewers 0
Chat rate 0
Playbacks 5
Average watchtime -

Concurrent viewers

Figure 5.4.4 Shows the youtube live interface.

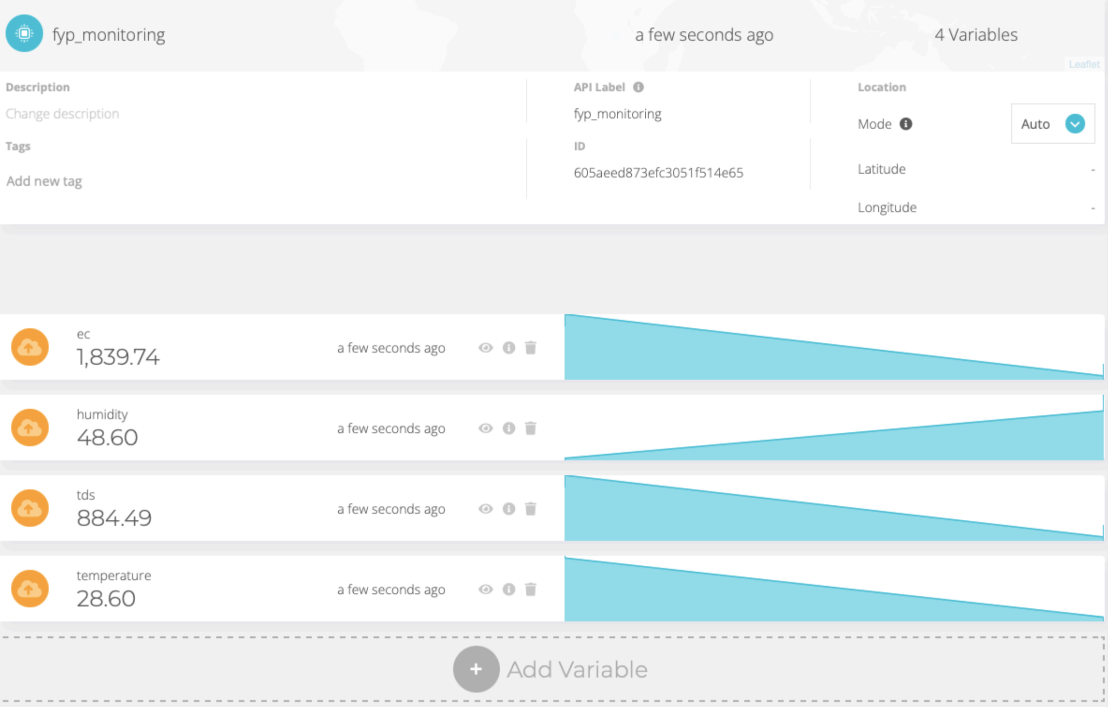
The live feed is then embedded into the HTML Canvas in Ubidots dashboard by using the youtube embedded functions.



```
<div class="iframe-container">
    <iframe width="560" height="315" src="https://www.youtube-nocookie.com/embed/9MKESSvERIw" title="YouTube video player"
        frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
        allowfullscreen></iframe>
</div>
```

Figure 5.4.5 Shows the HTML Canvas editor of Ubidots Dashboard.

To setup the widget which show the farm status on the Dashboard, Ubidots need to receive the first MQTT ping from the main controller which will register the device and the amount of sensor declared. If the MQTT ping works correctly, the devices tab will be showing the main controller and the value which is register as variables.



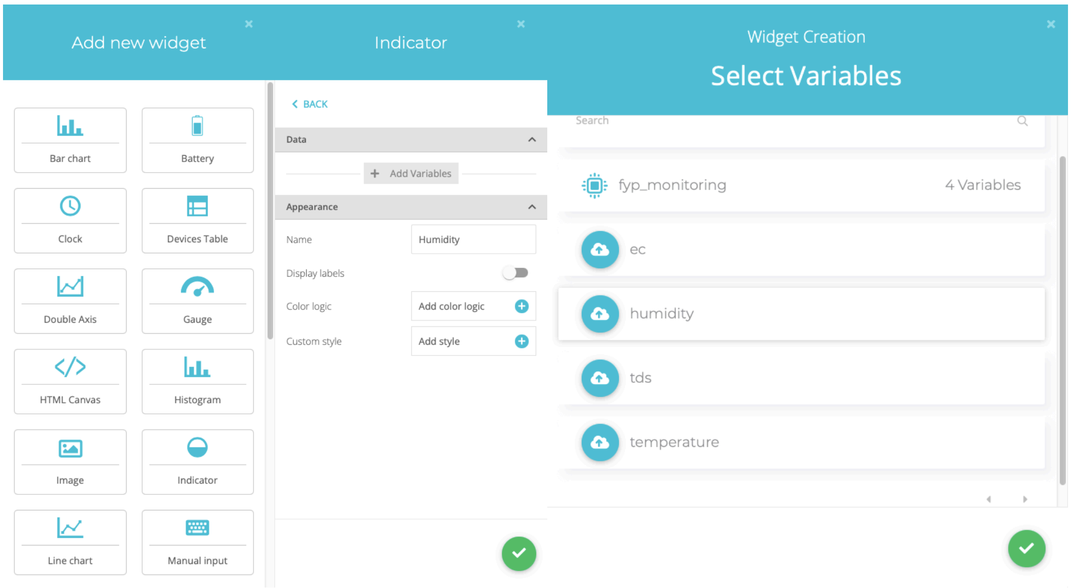
fyp_monitoring — a few seconds ago — 4 Variables

Description: Change description
Tags: Add new tag
API Label: fyp_monitoring
ID: 605aeed873efc3051f514e65
Location
Mode: Auto
Latitude: -
Longitude: -

Variable	Value	Last update
ec	1,839.74	a few seconds ago
humidity	48.60	a few seconds ago
tds	884.49	a few seconds ago
temperature	28.60	a few seconds ago

Figure 5.4.6 Shows the main controller is registered into the Ubidots services.

To place the widget which contain the value of data read from the sensors, select the add new widget and select type of widget to place. Then, in the data column select the device and which variable to display. Rename the widget into user-friendly name and select the green tick to confirm.



Name: Humidity

Display labels

Color logic: Add color logic

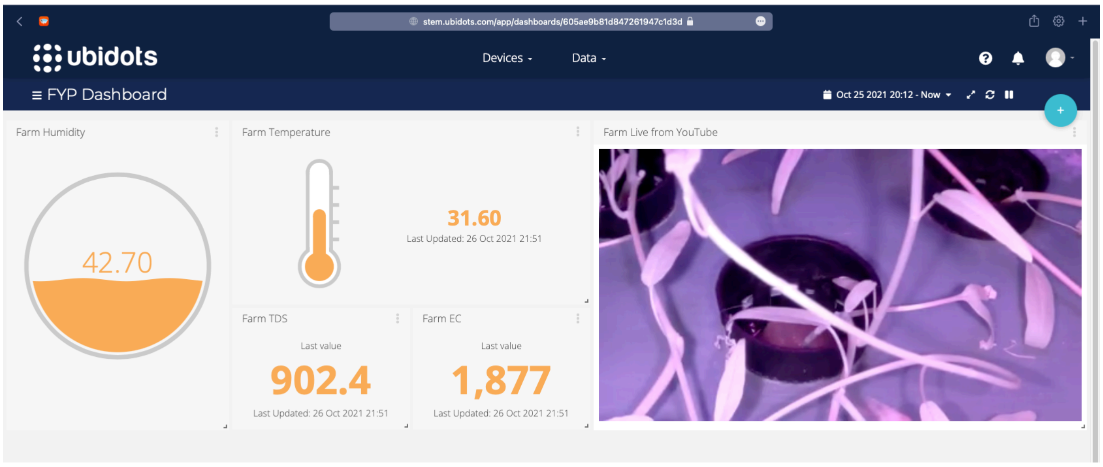
Custom style: Add style

fyp_monitoring 4 Variables

- ec
- humidity
- tds
- temperature

Figure 5.4.7 Shows the selection to define variable and place widget.

With just a login into Ubidots, the status of the farm can be known in just a glance.



Farm Humidity: 42.70

Farm Temperature: 31.60 Last Updated: 26 Oct 2021 21:51

Farm Live from YouTube

Farm TDS: Last value 902.4 Last Updated: 26 Oct 2021 21:51

Farm EC: Last value 1,877 Last Updated: 26 Oct 2021 21:51

Figure 5.4.8 Shows the interface of the Ubidots Dashboard.

5.5 Concluding Remark

This chapter briefly discuss the process to setup the system and how to connect the sensors & actuators to the main controller. Then, the software setup & configuration is discuss followed by the operation of system which include the expected output of the system.

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

To ensure the system works well & without any error, system testing is a crucial step which will show any imperfects in the operation of the system. To check for any imperfects that is hidden in the current system, three (3) indicator is selected as the performance metrics to evaluate if the system is usable, which is:

- Sensor's accuracy,
- Live stream feed continuity, and
- Growth of the crops.

Sensor's accuracy is the most crucial part in this project as it is the indicator which will tell if the farm is working as intended within the optimal environment. In this system, there are four (4) value which is needed to be take note of, which is the temperature, humidity, TDS, and EC. Temperature and humidity are much more consistent as there are python library which can be used immediately, meanwhile the TDS & EC require calibration through ADC MCP3008 as the main controller Raspberry Pi does not have any analog pin and the TDS & EC values are temperature sensitive.

Live stream feed continuity is the second crucial part in this project as it is the only method to observe the crops status from a remote location without going near the farming framework. A continuous live feed is important where the crops condition can be monitor at any time & a recording can be made at the same time to rewind whenever necessary.

Growth of the crops is the indicator of the effectiveness of the farming framework. To show that the farming framework is effective, three (3) traits of the crop's growth is selected as the indicator, which is:

- Height of crops,
- Number of leaves, and
- Colour of leaves.

6.2 Testing Setup and Result

The sensor will be mainly tested using a thermometer with humidity and TDS/EC meter to ensure the value retrieve is accurate and consistent at different temperature.

Sensor's accuracy

Temperature & Humidity

Thermometer		**DHT22**	
Temperature (°C)	**Humidity (%)**	**Temperature (°C)**	**Humidity (%)**
26.4	56	26.7	57.3
27.2	60	27.5	62.4
28.6	68	28.8	69.1
29.1	71	29.5	71.7
30.3	79	30.9	80.2
31.4	80	31.6	81.4

Table 6.2.1 Shows the comparison of thermometer with DHT22.

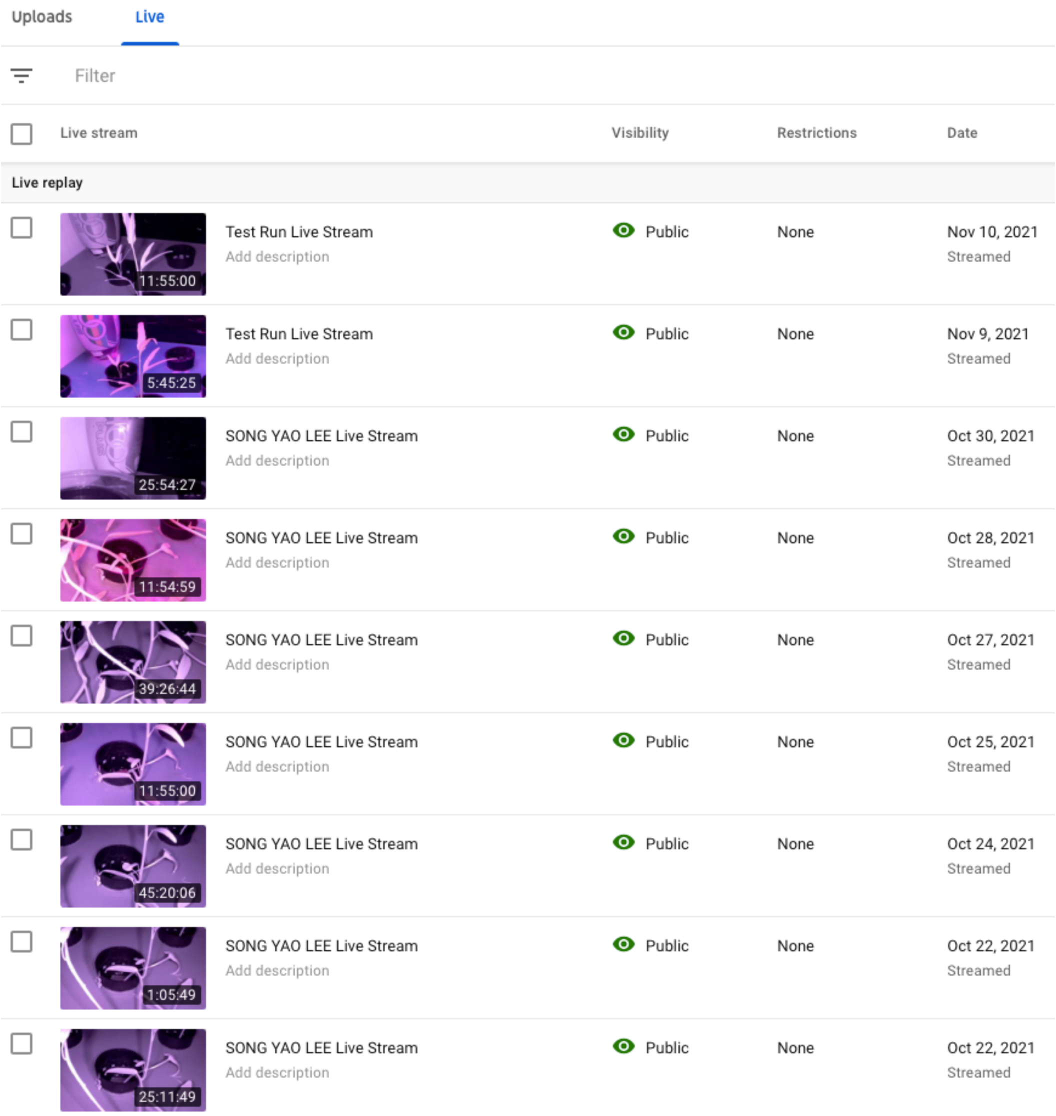
TDS & EC

Temperature (°C)	**Meter**		**Sensor**	
	TDS (ppm)	**EC**	**TDS (ppm)**	**EC**
26.3	825	1651	875	1820
28.5	873	1746	885	1840
29.5	798	1597	885	1842
30.9	770	1541	886	1847
31.4	843	1636	884	1839

Table 6.2.2 Shows the comparison of TDS/EC Meter with TDS sensor.

Live stream feed continuity

From the YouTube Live dashboard, the previous live streams shows that the live video is able to stream as long as the main controller is able to serve the live feed through the *raspivid* and *ffmpeg* software. In this project development period, the longest stream had a duration of more than 45 hours which is proven that with stable internet connection, the stream can last as long as the user desire. Due to the farming framework begin located further from the consumer routing devices, the main controller can only connect to the network using WiFi connection which might be interfere by other devices which is connected onto the same network.

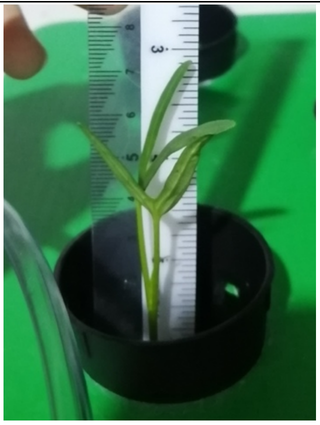
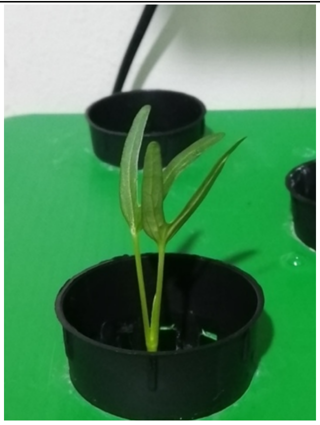
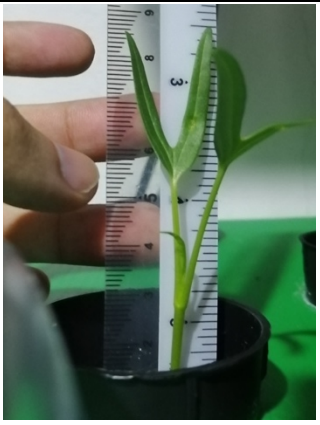
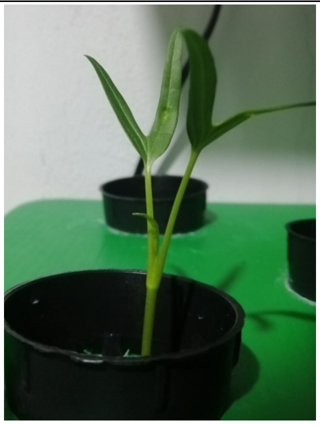
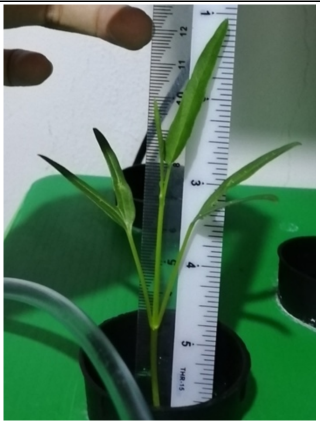
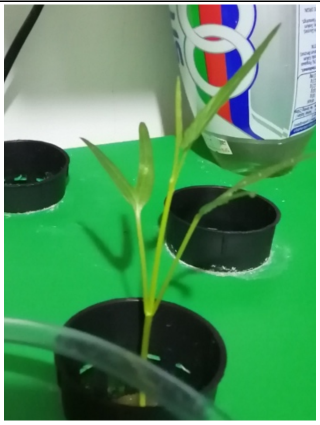


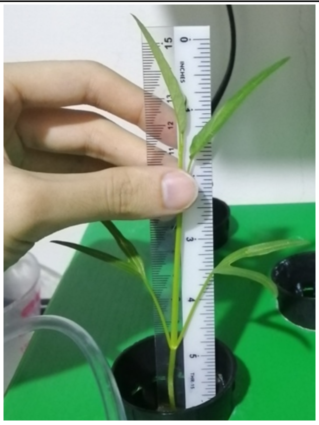
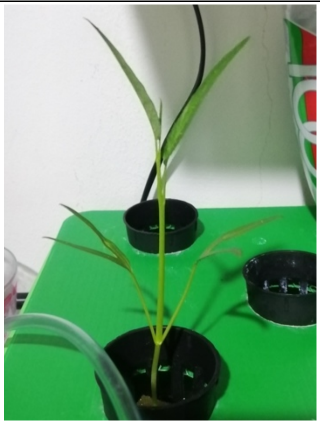
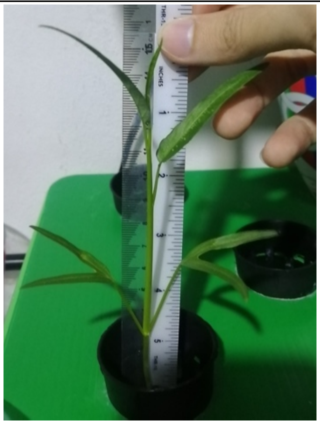
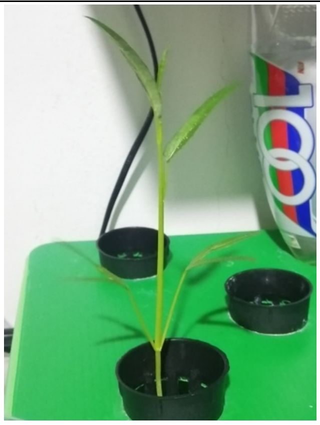
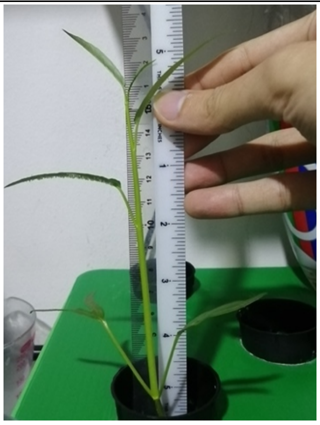
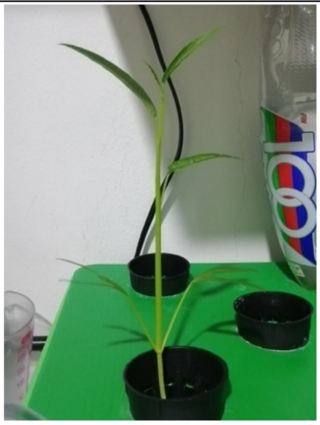
□	Live stream	Visibility	Restrictions	Date
	Live replay			
□	11:55:00 Test Run Live Stream Add description	Public	None	Nov 10, 2021 Streamed
□	5:45:25 Test Run Live Stream Add description	Public	None	Nov 9, 2021 Streamed
□	25:54:27 SONG YAO LEE Live Stream Add description	Public	None	Oct 30, 2021 Streamed
□	11:54:59 SONG YAO LEE Live Stream Add description	Public	None	Oct 28, 2021 Streamed
□	39:26:44 SONG YAO LEE Live Stream Add description	Public	None	Oct 27, 2021 Streamed
□	11:55:00 SONG YAO LEE Live Stream Add description	Public	None	Oct 25, 2021 Streamed
□	45:20:06 SONG YAO LEE Live Stream Add description	Public	None	Oct 24, 2021 Streamed
□	1:05:49 SONG YAO LEE Live Stream Add description	Public	None	Oct 22, 2021 Streamed
□	25:11:49 SONG YAO LEE Live Stream Add description	Public	None	Oct 22, 2021 Streamed

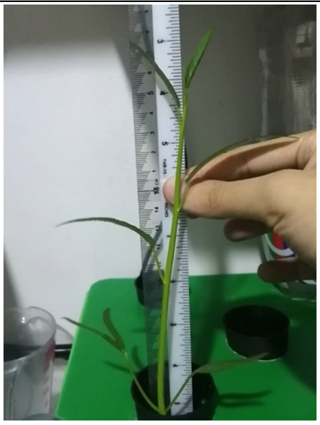
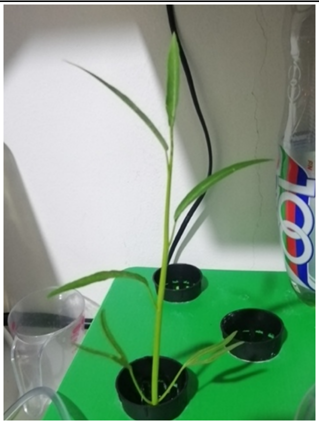
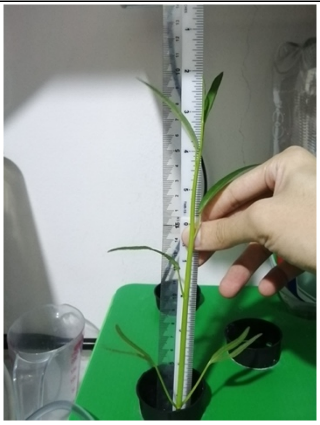
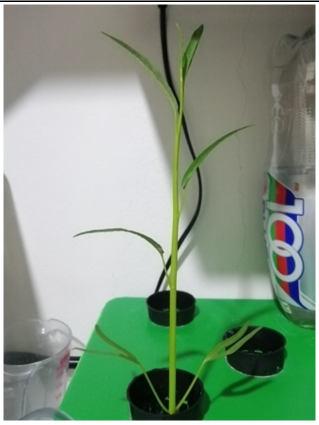
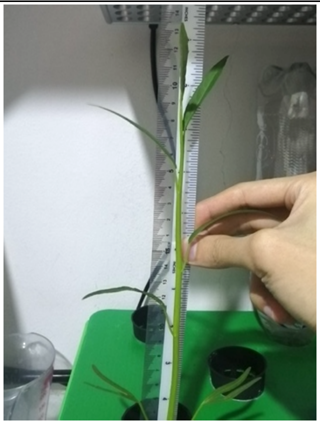
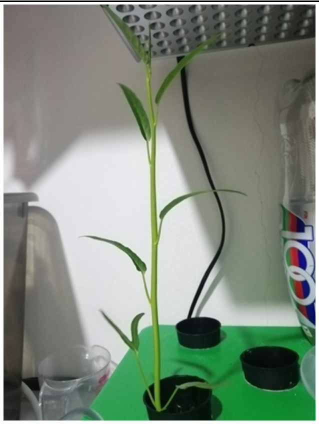
Figure 6.2.1 Shows the YouTube Live previous live stream.

Growth of crops

The growth of crops, *Water Spinach (Kangkung),* is observe & recorded after germination & is ready to place within the farming framework.

Day	Length (cm)	Number of leaves	Colour of leaves
5	 7.5 cm	 4	Green
6	 8.8 cm	 4	Green
7	 12.5 cm	 6	Green

8	 16 cm	 7	Green
9	 17 cm	 7	Green
10	 19 cm	 8	Green

11	 23 cm	 8	Green
12	 25 cm	 9	Green
13	 28.5	 9	Green

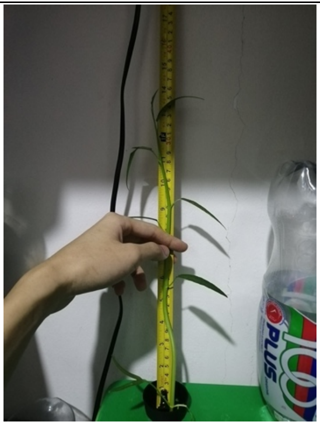
14	 30 cm	 10	Green
15	 32 cm	 11 - 1 (1 leaf is cut off in an accident when measuring the length)	Green
16	 36 cm	 10	Green

Table 6.2.3 Shows the growth of crops.

6.3 Project Challenges

There are a few challenges which need to be address in this project, which is:

- Environment temperature & humidity
- Live feed continuity
- TDS & EC temperature sensitivity

As the system is design to be a self-sustainable farming framework which should be able to source locally, there is no temperature and humidity controller which will completely isolate the farming framework and therefore, the temperature & humidity of the farming framework is completely dependent on the location the farming framework is located.

The live feed continuity is also a challenge which is directly impacted by the condition of the internet connection. The YouTube Live platform does not support reconnection after disconnection for a few seconds cause a problem in maintaining a stable live feed to the SaaS Ubidots Dashboard as after a disconnection, the live feed will be compiled into a video and the link which is used by the Dashboard to locate the live feed will be showing that specific live feed. To get the latest live feed showing on the Dashboard, the command to run *raspivid* & *ffmpeg* need to be run again, then dismiss the previous live feed in the YouTube Live studio and get the latest embedded link from the YouTube Live preview.

Due to the temperature-sensitive nature of TDS & EC, the value is not going to be accurate as the value read from a TDS/EC Meter. The system lacks a waterproof temperature sensor which can be place in the nutrient solution to monitor the temperature of the nutrient solution. Therefore, the calculation of TDS & EC read from the sensor is fixed using a specific temperature which is 28°C. When the temperature is getting colder or warmer, the value read from the sensor will have an incorrect reading ranging in ±200 to 300.

6.4 SWOT

	Helpful	**Harmful**
Internal Origin	**Strengths** - Simple & straight-forward setup. - Live feed can be embedded into SaaS Ubidots Dashboard.	**Weakness** - Lack of temperature & humidity controller. - Lack of live continuity if disconnection occur.
External Origin	**Opportunities** - Affordable components. - Vast Python library available.	**Threats** - Inaccurate TDS & EC reading due to temperature. - Unstable temperature & humidity when setup at locations with extreme climate.

Table 6.4.1 Shows the SWOT table.

6.5 Objectives Evaluation

To ensure the farming framework developed is usable and meet the requirement, the objectives is used as the guideline:

1. To create a fully automated indoor hydroponic farm framework.
   - Tends the crops within the farm automatically with the least human interaction possible.
2. To design the indoor hydroponic farming framework with the IoT technologies.
   - Using Single Board Computer (SBC) Raspberry Pi 3B+ as the main controller to collect data from sensors, publish data online, and control actuators to modify the status of the farm base on analysing data collected from the sensors.
3. To maintain the start-up of the farming framework to be easy in sourcing and setting up.
   - The framework of the farm is build using everyday material which should be found as locally as possible, and the setup should be straight forward for stress-free deployment in any other location.

The first objective is considered as achieve partially as the data from the sensor can be read and transmitted without any issues, but the live feed could not reconnect if disconnection occurs and require human interaction to restart the live feed.

The second objective is considered as achieve perfectly by having the main controller Raspberry Pi to connect to all sensors, actuators and is able to stream the live feed to the streaming platform at the same time without any bottlenecking in performance while other computing devices are only required for remote desktop into the main controller.

The third objective is considered as achieve perfectly as the setup of a running framework is straight forward with the help of vast python library to perform automation and most of the farming framework can be built up from everyday items which can be obtain from franchise hypermarket & hardware stores while the main controller, sensors, & the relay module which are required can be obtained from specific online retailer but are still accessible by the public such as Cytron & Autobotic.

6.6 Concluding Remark

This chapter listed the details of the system testing & which metrices is chosen to evaluate the performance of the developed farming framework. The result of different testing is then display. Then, the challenge of this project is briefly discussed with the SWOT following. The chapter is then concluded in the evaluation of the objectives.

CHAPTER 7 CONCLUSION & RECOMMENDATION

7.1 Conclusion

The report purpose is to develop an automated indoor farming framework which can be easily built using everyday items and is able to perform monitoring & automation. Currently, the major farming method are still the traditional soil-based method which relies on weather.

In this project, an indoor hydroponic farming framework which will allow the user to monitor the status of the farming environment through sensors, live feed through camera, and automation through relay module which can be deploy without purchasing much custom hardware will be develop & prototypes will be deployed for performance evaluation.

The indoor hydroponic farming framework main function is to read the data from the sensors, perform automated actions using actuators connected through the relay module, stream live feed to streaming platform, and publish data collected to SaaS Dashboard using MQTT API. The user can view the status of the farm from the Dashboard which will integrate all data onto one interactive webpage.

Lastly, the proposed system which is developed will be isolated which reduce dependencies to the weather making the farm to be more controllable in creating a consistent environment for growing crops.

7.2 Recommendation

In this project, the indoor hydroponic farming framework is designed for the people which are not planning for having large scale farm with small budget which will not break a hole in the wallet. So, the farming framework is only equipped with the basic monitoring sensors for the temperature, humidity, EC, & TDS with some automation actuators which is the grow light & fans for exhaust & air circulation.

There are a few recommendations which will be a great enhancement to the system in the future:

- Controllable temperature & humidity which can further isolate the farm to reduce dependencies to environment temperature & humidity.
- Automated dossing system which increases the nutrient solution when the TDS & EC value read from the sensor is getting too low.
- Automated water level replenishment system which maintains the water level from evaporations.
- User can adjust trigger point for the actuators to start working from the SaaS Ubidots dashboard.

BIBLIOGRAPHY

Aliac, C. J. G. & Maravillas, E., 2018. IOT Hydroponics Management System. Baguio City, Philippines, IEEE.

Bakhtar, N. et al., 2018. IoT based Hydroponic Farm. Tirunelveli, India, IEEE.

Crisnapati, P. N., Wardana, I. N. K., Aryanto, I. K. A. A. & Hermawan, A., 2017. Hommons: Hydroponic Management and Monitoring System for an IOT Based NFT Farm Using Web Technology. Denpasar, IEEE.

Gabriela, C., Monica, H. & Boris, B., 2018. Internet of Things (IoT) Applied to an Urban Garden. Barcelona, Spain, IEEE.

Garrett, C., 2021. MicroPython: An Intro to Programming Hardware in Python – Real Python. [online] Realpython.com. Available at: <https://realpython.com/micropython/> [Accessed 10 August 2021].

Génération Robots. 2021. MCP3008 – 8-Channel 10-Bit ADC with SPI Interface. [online] Available at: <https://www.generationrobots.com/en/402440-mcp3008-8-channel-10-bit-adc-with-spi-interface.html> [Accessed 3 November 2021].

MD SAAD, N., 2021. Covid-19 has disrupted supply chain | New Straits Times. [online] NST Online. Available at: <https://www.nst.com.my/opinion/letters/2021/08/715780/covid-19-has-disrupted-supply-chain> [Accessed 22 November 2021].

Monahan, T., 2021. complete list of ffmpeg flags / commands. [online] Github Gist. Available at: <https://gist.github.com/tayvano/6e2d456a9897f55025e25035478a3a50> [Accessed 9 November 2021].

Raspberrypi.com. 2021. Raspberry Pi Documentation - Camera. [online] Available at: <https://www.raspberrypi.com/documentation/accessories/camera.html#raspicam-commands> [Accessed 9 November 2021].

Raspberrypi.org. 2021. Raspberry Pi 3 Model B+. [online] Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> [Accessed 7 August 2021].

Saraswathi , D., Manibharathy , P., Gokulnath , R. & Sureshkumar , E., 2018. Automation of Hydroponics Green House Farming using IOT. ICSCAN, IEEE.

Shiklo, B., 2021. 8 Software Development Models Organized in Charts and Explained. [online] Scnsoft.com. Available at: <https://www.scnsoft.com/blog/software-development-models> [Accessed 3 November 2021].

Store.arduino.cc. 2021. Arduino Uno Rev3 | Arduino Official Store. [online] Available at: <https://store.arduino.cc/usa/arduino-uno-rev3> [Accessed 10 August 2021].

Store.arduino.cc. 2021. Arduino Uno Rev3 | Arduino Official Store. [online] Available at: <https://store.arduino.cc/usa/arduino-uno-rev3> [Accessed 10 August 2021].

Wiki.keyestudio.com. 2021. KS0429 keyestudio TDS Meter V1.0 - Keyestudio Wiki. [online] Available at: <https://wiki.keyestudio.com/KS0429_keyestudio_TDS_Meter_V1.0> [Accessed 3 November 2021].

Wiki.sunfounder.cc. 2021. 8 Channel 5V Relay Module - Wiki. [online] Available at: <http://wiki.sunfounder.cc/index.php?title=8_Channel_5V_Relay_Module> [Accessed 3 November 2021].

Wong, L., 2020. 4 local engineers believe vertical farming offers answer to food sustainability. [online] The Star. Available at: <https://www.thestar.com.my/lifestyle/living/2020/03/06/4-engineers-believe-vertical-farming-offers-answer-to-food-sustainability> [Accessed 22 November 2021].

APPENDIX A

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 1, Year 3	**Study week no.:** 1
Student Name & ID: Lee Song Yao, 19ACB05829	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: A Prototype of Indoor Hydroponics Farming Framework	

1. WORK DONE

- Stream live video feed from Pi Camera to internet and embedded within the Ubidots dashboard.
- Devise study plan for the trimester.

2. WORK TO BE DONE

- Increase content on Project II report.
- Stress test the system on data collection and streaming altogether.
- Monitoring & calibration whenever required.

3. PROBLEMS ENCOUNTERED

- No problem encountered so far.

4. SELF EVALUATION OF THE PROGRESS

- The current project development is progress as planned.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 1, Year 3	**Study week no.:** 2
Student Name & ID: Lee Song Yao, 19ACB05829	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: A Prototype of Indoor Hydroponics Farming Framework	

1. WORK DONE

- Chapter 5 (system implementation) of report completed.
- Starting a new batch of crops for monitoring.

2. WORK TO BE DONE

- Increase content on Project II report.
- Stress test the system on data collection and streaming altogether.
- Monitoring & calibration whenever required.

3. PROBLEMS ENCOUNTERED

- No problem encountered so far.

4. SELF EVALUATION OF THE PROGRESS

- The current project development is progress as planned.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 1, Year 3	**Study week no.:** 3
Student Name & ID: Lee Song Yao, 19ACB05829	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: A Prototype of Indoor Hydroponics Farming Framework	

1. WORK DONE

- Making amendment on project II report chapter 3 – 5.

2. WORK TO BE DONE

- Increase content on Project II report.
- Stress test the system on data collection and streaming altogether.
- Monitoring & calibration whenever required.

3. PROBLEMS ENCOUNTERED

- No problem encountered so far.

4. SELF EVALUATION OF THE PROGRESS

- The current project development is progress as planned.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 1, Year 3	**Study week no.:** 4
Student Name & ID: Lee Song Yao, 19ACB05829	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: A Prototype of Indoor Hydroponics Farming Framework	

1. WORK DONE

- Data collection for chapter 6
- Project II report chapter 6.

2. WORK TO BE DONE

- Proofread on Project II report.
- Stress test the system on data collection and streaming altogether.
- Monitoring & calibration whenever required.

3. PROBLEMS ENCOUNTERED

- No problem encountered so far.

4. SELF EVALUATION OF THE PROGRESS

- The current project development is progress as planned.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 1, Year 3	**Study week no.:** 5
Student Name & ID: Lee Song Yao, 19ACB05829	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: A Prototype of Indoor Hydroponics Farming Framework	

1. WORK DONE

- Daily data collection for Chapter 6
- Amend Project II report chapter 1, 6 and 7.

2. WORK TO BE DONE

- Proofread on Project II report content and formatting.
- Stress test the system on data collection and streaming altogether.
- Monitoring & calibration whenever required.

3. PROBLEMS ENCOUNTERED

- No problem encountered so far.

4. SELF EVALUATION OF THE PROGRESS

- The current project development is progress as planned.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(~~Project I~~ / Project II)

Trimester, Year: Trimester 1, Year 3	**Study week no.:** 6
Student Name & ID: Lee Song Yao, 19ACB05829	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: A Prototype of Indoor Hydroponics Farming Framework	

1. WORK DONE

- Major amendment on Project II report chapter 1, 6 and 7.
- Plagiarism checking through Turitin.
- Proofread & finalized Project II report content and formatting.

2. WORK TO BE DONE

- Prepare for presentation.

3. PROBLEMS ENCOUNTERED

- No problem encountered so far.

4. SELF EVALUATION OF THE PROGRESS

- The current project development is progress as planned.

Supervisor’s signature

Student’s signature

APPENDIX B

Script to connect main controller Raspberry Pi to Ubidots MQTT:

```
import time
import requests
import math
import random

#SenseHat for testing
#from sense_emu import SenseHat
#hat = SenseHat()
#hat.clear()
#i = 0

#DHT22 Library
import time
import Adafruit_DHT
dht = Adafruit_DHT.DHT22

#MCP3008 Library
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn

#GPIO Control
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM) # GPIO Numbers instead of board numbers

TOKEN = "BBFF-3qG6GWKOlQzRccJY4OvogkSBxkY4nn"
DEVICE_LABEL = "fyp_monitoring"
VARIABLE_LABEL_1 = "temperature"
VARIABLE_LABEL_2 = "humidity"
VARIABLE_LABEL_3 = "tds"
VARIABLE_LABEL_4 = "ec"


def build_payload(variable_1, variable_2, variable_3, variable_4):
    humidity, temperature = Adafruit_DHT.read_retry(dht, 14)
    #hat_humidity = round(hat.get_humidity(), 2)

    #DHT22
    value_1 = temperature
    value_2 = humidity

    #MCP3008
    # create the spi bus
    spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
    # create the cs (chip select)
    cs = digitalio.DigitalInOut(board.D5)
    # create the mcp object
    mcp = MCP.MCP3008(spi, cs)
    # create an analog input channel on pin 0
    raw0 = AnalogIn(mcp, MCP.P0)
    vol = raw0.value * 5 / 1024
    # !!Need calibrations!!
    # TDS (accurate at 29C)
    value_3 = vol * 3.83
    # EC
    value_4 = (vol * 3.83) * 2.08

    int_value_2 = int(value_2)
    # Automated Exhaust
    if (int_value_2 < 60):
        pin = 24
        GPIO.setup(pin, GPIO.OUT) # GPIO Assign mode
        GPIO.output(pin, GPIO.LOW) # out
    else:
        pin = 24
        GPIO.setup(pin, GPIO.OUT) # GPIO Assign mode
        GPIO.output(pin, GPIO.HIGH) # on

    payload = {variable_1: value_1, variable_2: value_2, variable_3: value_3,
variable_4: value_4}
    return payload


def post_request(payload):
    # Creates the headers for the HTTP requests
    url = "http://industrial.api.ubidots.com"
    url = "{}/api/v1.6/devices/{}".format(url, DEVICE_LABEL)
    headers = {"X-Auth-Token": TOKEN, "Content-Type": "application/json"}

    # Makes the HTTP requests
    status = 400
    attempts = 0
    while status >= 400 and attempts <= 5:
        req = requests.post(url=url, headers=headers, json=payload)
        status = req.status_code
        attempts += 1
        time.sleep(10)

    # Processes results
    print(req.status_code, req.json())
    if status >= 400:
        print("[ERROR] Could not send data after 5 attempts, please check \
            your token credentials and internet connection")
        return False

    print("[INFO] request made properly, your device is updated")
    return True
```

```
def main():
    payload = build_payload(VARIABLE_LABEL_1, VARIABLE_LABEL_2, VARIABLE_LABEL_3,
VARIABLE_LABEL_4)

    print("[INFO] Attemping to send data")
    post_request(payload)
    print("[INFO] finished")


if __name__ == '__main__':
    while (True):
        main()
        time.sleep(1)
```

Script to activate actuators (Grow Light):

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM) # GPIO Numbers instead of board numbers

pin = 23
GPIO.setup(pin, GPIO.OUT) # GPIO Assign mode
GPIO.output(pin, GPIO.HIGH) # on
```

Script to deactivate actuators (Grow Light):

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM) # GPIO Numbers instead of board numbers

pin = 23
GPIO.setup(pin, GPIO.OUT) # GPIO Assign mode
GPIO.output(pin, GPIO.LOW) # out
```

Command to start *raspivid* and *ffmpeg* for live feed transmission:

```
raspivid -o - -t 0 -vf -hf -fps 30 -b 6000000 | ffmpeg -re -ar 44100
-ac 2 -acodec pcm_s16le -f s16le -ac 2 -i /dev/zero -f h264 -i - -
vcodec copy -acodec aac -ab 128k -g 50 -strict experimental -f flv
rtmp://a.rtmp.youtube.com/live2/ggfk-kh0v-2mhe-2vq0-356d
```

APPENDIX C

A PROTOTYPE OF INDOOR HYDROPONICS FARMING FRAMEWORK

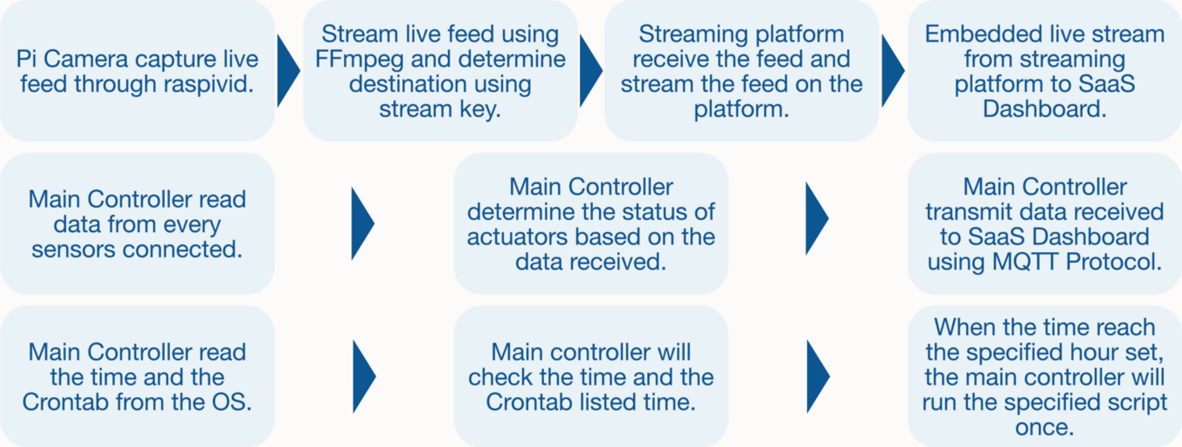
BY LEE SONG YAO

UNIVERSITI TUNKU ABDUL RAHMAN

INTRODUCTION

This project aims to develop an indoor hydroponic farming framework which is independent from outdoor weather which is able to monitor & automate task so that the user does not need to interact with the farming framework until the crops is ready to harvest. This farming framework provide basic monitoring for temperature, humidity, TDS & EC while also automated the task to control the exhaust fan based on the environment condition & automated grow light at specific times. Besides, the farming framework also provide live feed of the farming framework interior through streaming platform. The SaaS Dashboard receive the data collected from the sensors through MQTT and embedded the live feed which is being stream from the streaming platform to the dashboard. The items which are chosen to build the project are mostly everyday items which can be bought from franchise hypermarket & hardware store while the electronic can be purchase from the online retailer.

METHODOLOGY



Pi Camera capture live feed through raspivid.

Stream live feed using FFmpeg and determine destination using stream key.

Streaming platform receive the feed and stream the feed on the platform.

Embedded live stream from streaming platform to SaaS Dashboard.

Main Controller read data from every sensors connected.

Main Controller determine the status of actuators based on the data received.

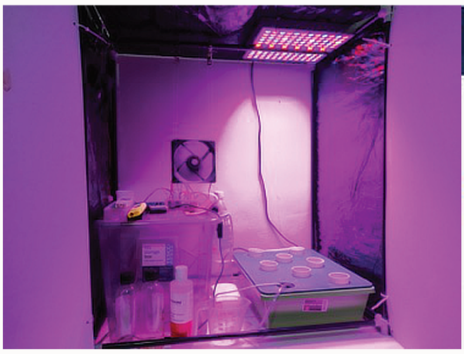
Main Controller transmit data received to SaaS Dashboard using MQTT Protocol.

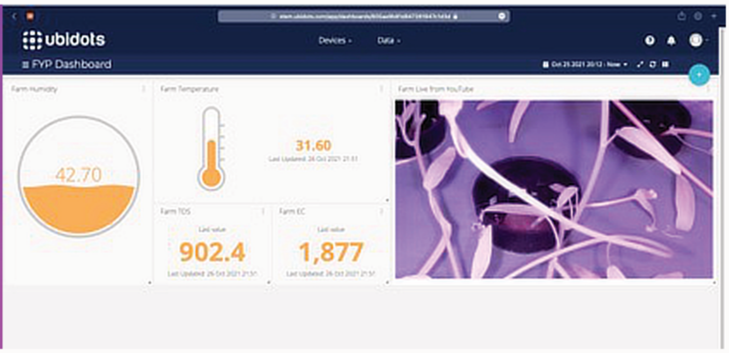
Main Controller read the time and the Crontab from the OS.

Main controller will check the time and the Crontab listed time.

When the time reach the specified hour set, the main controller will run the specified script once.

PROTOTYPE



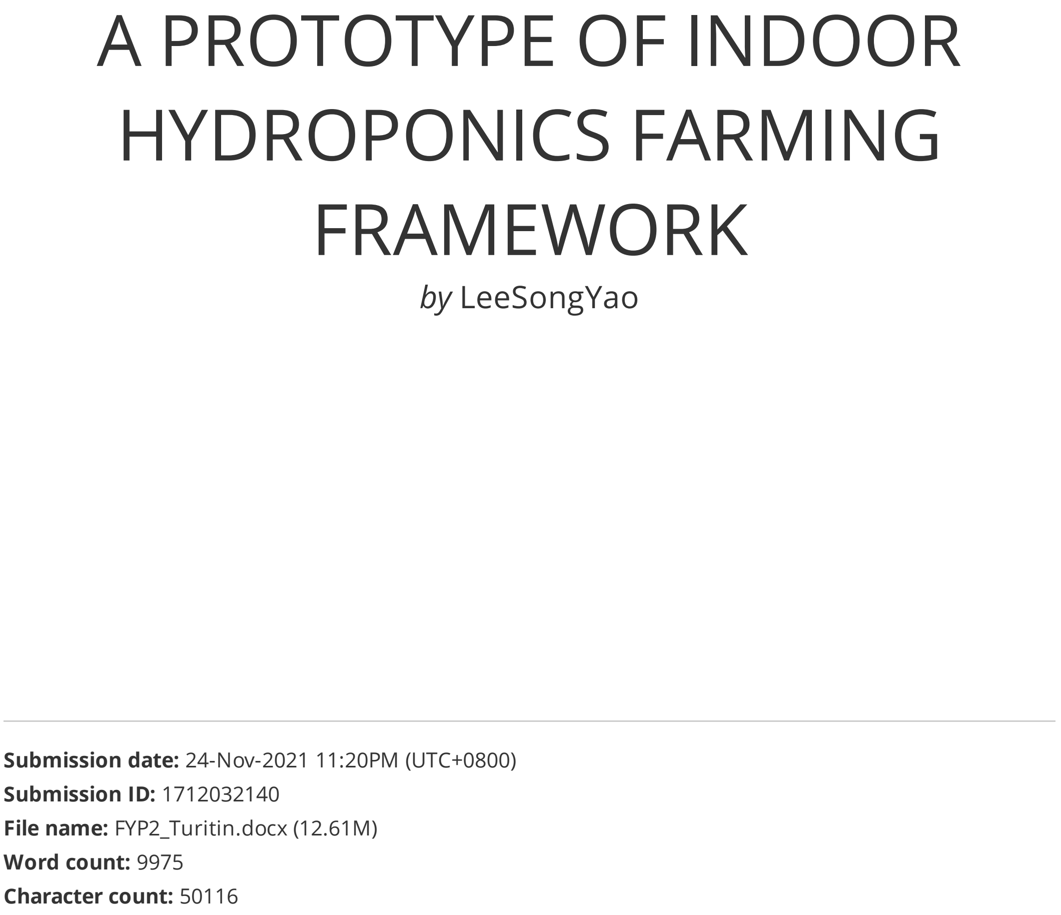


CONCLUSION

With the help of the internet and IoT technologies, farming can be easier and more controllable with the aid of sensors, actuators, and cloud platforms. Tasks can be automated be replace human labour in maintaining the necessities for crops growth while cameras can be placed to let the user peek on the status of the farm at any time while waiting for the crops to mature for harvesting.

SUPERVISE BY TS DR. GOH HOCK GUAN

PLAGIARISM CHECK SUMMARY

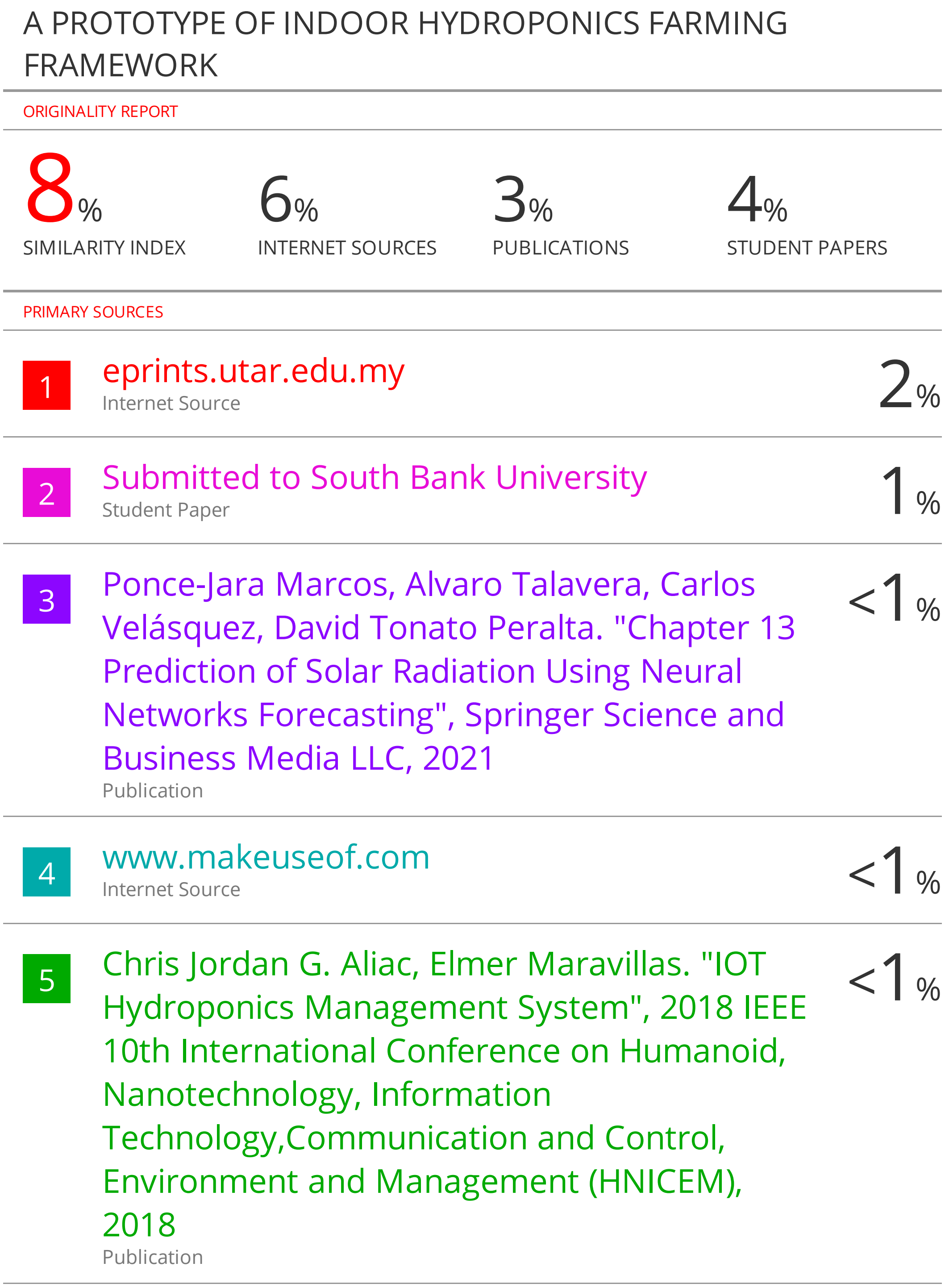


A PROTOTYPE OF INDOOR HYDROPONICS FARMING FRAMEWORK

by LeeSongYao

Submission date: 24-Nov-2021 11:20PM (UTC+0800)
Submission ID: 1712032140
File name: FYP2_Turitin.docx (12.61M)
Word count: 9975
Character count: 50116

PLAGIARISM CHECK SUMMARY



A PROTOTYPE OF INDOOR HYDROPONICS FARMING FRAMEWORK

ORIGINALITY REPORT

8%	6%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	Submitted to South Bank University Student Paper	1%
3	Ponce-Jara Marcos, Alvaro Talavera, Carlos Velásquez, David Tonato Peralta. "Chapter 13 Prediction of Solar Radiation Using Neural Networks Forecasting", Springer Science and Business Media LLC, 2021 Publication	<1%
4	www.makeuseof.com Internet Source	<1%
5	Chris Jordan G. Aliac, Elmer Maravillas. "IOT Hydroponics Management System", 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), 2018 Publication	<1%

6	Azimbek Khudoyberdiev, Shabir Ahmad, Israr Ullah, DoHyeun Kim. "An Optimization Scheme Based on Fuzzy Logic Control for Efficient Energy Consumption in Hydroponics Environment", Energies, 2020 Publication	<1 %
7	Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur Student Paper	<1 %
8	Submitted to University of Technology, Sydney Student Paper	<1 %
9	wiki.keyestudio.com Internet Source	<1 %
10	Submitted to Technological Institute of the Philippines Student Paper	<1 %
11	link.springer.com Internet Source	<1 %
12	Manuel J. Ibarra, Edgar W. Alcarraz, Olivia Tapia, Aydeé Kari, Yalmar Ponce, Rosmery S. Pozo. "Chapter 47 IoT Computing for Monitoring NFT-I Cultivation Technique in Vegetable Production", Springer Science and Business Media LLC, 2022 Publication	<1 %

13	Submitted to Oxford Brookes University Student Paper	<1 %
14	Guy Hart-Davis. "Deploying Raspberry Pi in the Classroom", Springer Science and Business Media LLC, 2017 Publication	<1 %
15	publications.theseus.fi Internet Source	<1 %
16	Submitted to University of Wollongong Student Paper	<1 %
17	sysadmin.psu.ac.th Internet Source	<1 %
18	www.bernama.com Internet Source	<1 %
19	Submitted to TAR University College Student Paper	<1 %
20	www.pbtech.co.nz Internet Source	<1 %
21	projects-raspberry.com Internet Source	<1 %
22	studentsrepo.um.edu.my Internet Source	<1 %
23	www.dilsonenterprises.net Internet Source	<1 %

24	www.elektormagazine.nl Internet Source	<1 %
25	docplayer.net Internet Source	<1 %
26	www.amazon.com Internet Source	<1 %
27	Jeff Cicolani. "Beginning Robotics with Raspberry Pi and Arduino", Springer Science and Business Media LLC, 2021 Publication	<1 %
28	Submitted to University of Portsmouth Student Paper	<1 %
29	dokumen.pub Internet Source	<1 %
30	electronicexperiments.blogspot.com Internet Source	<1 %
31	www.freepatentsonline.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		

PLAGIARISM CHECK SUMMARY

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LEE SONG YAO
ID Number(s)	19ACB05829
Programme / Course	BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMMUNICATIONS AND NETWORKING
Title of Final Year Project	A PROTOTYPE OF INDOOR HYDROPONICS FARMING FRAMEWORK

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 8 **%** **Similarity by source** Internet Sources: 6 % Publications: 3 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Goh Hock Guan

Date: 26/11/2021

Signature of Co-Supervisor

Name: __________

Date: __________

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB05829
Student Name	LEE SONG YAO
Supervisor Name	TS DR GOH HOCK GUAN

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
N/A	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Tables (if applicable)
√	List of Figures (if applicable)
N/A	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. ____________________ (Signature of Student) Date:26 November 2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ____________________ (Signature of Supervisor) Date: 26/11/2021