Faculty of Electrical and Electronic Engineering Technology

WEATHER MONITORING SYSTEM FOR DRONE DOCKING STATION

MUHAMAD LUTFIASHRAF BIN BAHARUDIN

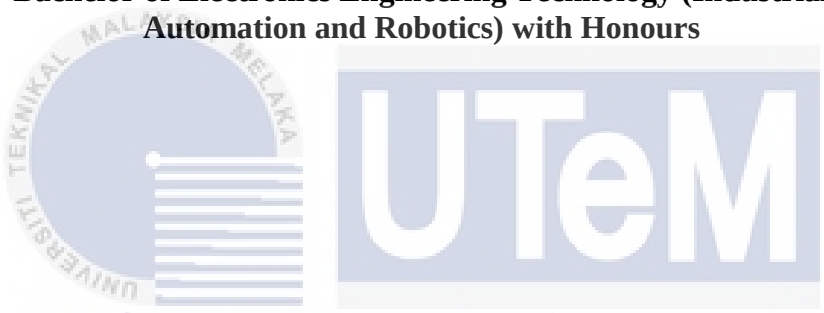
Bachelor of Electronics Engineering Technology (Industrial Automation and Robotics) with Honours

2021

WEATHER MONITORING SYSTEM FOR DRONE DOCKING STATION

MUHAMAD LUTFIASHRAF BIN BAHARUDIN

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology (Industrial Automation and Robotics) with Honours



Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I declare that this project report entitled “Weather Monitoring System For Drone Docking Station” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : 

Student Name : Muhamad Lutfiashraf Bin Baharudin

Date : 2 January 2022

APPROVAL

I hereby declare that I have checked this project report and, in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electronics Engineering Technology (Industrial Automation and Robotics) with Honours.

Signature : 

Supervisor Name : AMINURRASHID BIN NOORDIN

Date : 11/01/2022

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

This thesis is specially dedicated to:

My beloved father and late mother,

Baharudin bin Mohammad, Mislina binti Baharin,

My supervisor,

TS. Aminurrashid bin Noordin, And all my friends,

Thank you for their encouragement and unconditionally support.

ABSTRACT

The key goals of Weather Monitoring System for Drone Docking Station was to assist drone pilots in accessing and obtaining weather data. The weather monitoring technology allows the drone to dock anywhere in real time. Drones have both autopilot and manual flight modes. In comparison to manual drones, autoflight allows drones to fly and dock autonomously. However, in order to focus on weather monitoring, this project only used manual piloting. To study the weather and climate, atmospheric conditions like wind, rain, humidity and temperature are measured using weather station facility. The information provided can be used to predict weather forecast for the drone to dock safely. Without a weather station, users can't be alerted of the strong winds, heavy rain, heat waves, or any other weather-related emergency. As a result, the drone suffers damage, perhaps costing additional money to repair. Different sensors are used as a device to collects data related to the weather and environment for the weather station. To overcome the problems faced, there are techniques that can be implemented to assist this project. Internet of Things with sensors will be used to build a weather station for the project. The weather station can help provide data for forecasts in order to use drone. The user can view the history of information as well once a weather station is connected. Command prompt is used to connect the IP address with the system. The Ip will receive and react to the ping and command to enter the raspberry pi. Finally, the result will show at command prompt and at the IoT program.

ABSTRAK

Matlamat utama Sistem Pemantauan Cuaca untuk Stesen Pelabuhan Dron adalah untuk membantu pengendali dron mengumpul dan mendapatkan data cuaca. Teknologi pemantauan cuaca membolehkan dron mendarat di mana-mana. Dron mempunyai dua mod penerbangan autopilot dan manual. Autoflight Dron membolehkan dron terbang dan mendarat sendiri secara autonomi berbanding manual dron. Projek ini hanya menggunakan pandu arah secara manual supaya dapat mengkaji cuaca dan iklim, keadaan atmosfera seperti angin, hujan, kelembapan dan suhu diukur menggunakan kemudahan stesen cuaca. Maklumat yang diberikan boleh digunakan untuk meramalkan ramalan cuaca untuk dron berlabuh dengan selamat. Tanpa stesen cuaca, pengguna tidak boleh dimaklumkan tentang angin kencang, hujan lebat, gelombang panas atau sebarang kecemasan berkaitan cuaca yang lain. Akibatnya, dron mungkin mengalami kerosakan dan memerlukan wang tambahan untuk dibaiki.Penderia yang berbeza digunakan sebagai peranti untuk mengumpul data yang berkaitan dengan cuaca dan persekitaran untuk stesen cuaca. Bagi mengatasi masalah yang dihadapi, terdapat teknik yang boleh dilaksanakan untuk membantu projek ini. Internet Pelbagai Benda dengan penderia akan digunakan untuk membina stesen cuaca untuk projek itu. Stesen cuaca boleh membantu menyediakan data untuk ramalan untuk menggunakan dron. Pengguna boleh melihat sejarah maklumat sebaik sahaja stesen cuaca disambungkan. Command prompt digunakan untuk menyambungkan alamat IP dengan sistem. Ip akan menerima dan bertindak balas kepada ping dan arahan untuk memasuki raspberry pi. Akhirnya, hasilnya akan ditunjukkan pada command prompt dan pada program IoT.

ACKNOWLEDGEMENTS

Assalamualaikum Warahmatullah Wabarakatuh.

First and foremost, I thank to Allah the Almighty for blessing me to finish my Bachelor Degree Project 1. With my sincere gratitude I would like to say thank to TS Aminurrashid bin Noordin for giving me his consolation, direction, backing and inspiration all through this entire venture. Under his watch, I obtained a great deal of significant information and proposal just as certainty to finish this undertaking. Despite he is occupied with his activity and obligations, he despite everything figured out how to direct me along to accomplish this undertaking. In this way, here I am to demonstrate my gratefulness to him for showing me calmly and I am thankful to have him as my supervisor.

In addition, I might want to thank to my kindred companions who consistently prepared to help me when I required. What more, my gratefulness to them for has helped me and offer a great deal of smart thoughts that help to achieve my venture.

At long last yet significantly, my most profound appreciation goes to my father and late mother Baharudin bin Mohammad and Mislina binti Baharin for supporting me intellectually and monetarily all through the whole examinations in UTeM. Their unending help has reached out to me all through this certificate study and my life when all is said in done.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

°C	-	Celcius
%	-	Percent sign
V	-	Voltage
cm	-	Centimetre
in	-	Inches
g	-	Gram

LIST OF ABBREVIATIONS

IoT	-	Internet Of Things
GUI	-	Graphic User Interface
VCC	-	Voltage Common Collector
GND	-	Ground
LED	-	Light Emitting Diode
ADC	-	Analog-to-Digital Converter
NPN	-	Negative Positive Negative
DC	-	Direct Current
IR	-	Infrared
VDC	-	Volts of Direct Current
PWM	-	Pulse-Width Modulation
WI-FI	-	Wireless Fidelity
LCD	-	Liquid Crystal Display

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Weather is one of the important factors for human to decide activities they can do for the day. For example, in sunny weather most of people love to do outdoor activities such as sunbathe, playing in garden or even hang their clothes due hot temperature outside their house. Meanwhile if rainy days, people tend to stay inside and play indoor activities with their family and friends. Weather or climate changing are not easy or accurately can be determine by people themselves. Thanks to technological improvements, weather may now be projected and monitored on a regular basis using modern techniques. Data collected over time can be utilised to improve the accuracy of weather information.

Drones are divided into two types: manual flight, which is controlled by humans, and autonomous flight, which can fly autonomously without the intervention of humans. In today's society, using a drone has several advantages. Drones can collect data in regions that people find difficult or inaccessible. It also takes more exact measurements and collects more accurate data than humans. However, it has limitations, particularly in terms of weather. Drones can only fly and collect data in clear, sunny conditions. If the drone is flown in an area with bad weather, the electronic components will be damaged and connection between the drone and the controller will be disrupted.

Weather monitoring for drone docking station is a real-time weather monitoring system that can be accessed via IoT to determine the weather conditions for docking drones. A timetable high-way traffic monitoring can be implemented in the future by deploying autonomous pilot. It's crucial to have a pleasant weather while releasing the drone from the docking station. That is why, with the assistance of technology and IoT, weather station monitoring in docking systems may be accomplished.

Therefore, this project focuses on designing a weather monitoring system for a drone docking station. This task just shows the weather parameters such as wind, rain, humidity and temperature for drone docking system by utilizing an appropriate and moderate microcontroller and sensors. This project likewise will present or show the preparing of the weather monitoring system that may show ongoing activity of the process through. Figure 1.1 shows the idea of drone docking station with weather monitoring system. Subsequently, a source coding will be Python and make to understand the process from the sensors and display the parameters. A test will be lead for ongoing activity and do revision or improvement of the system both coding and equipment.

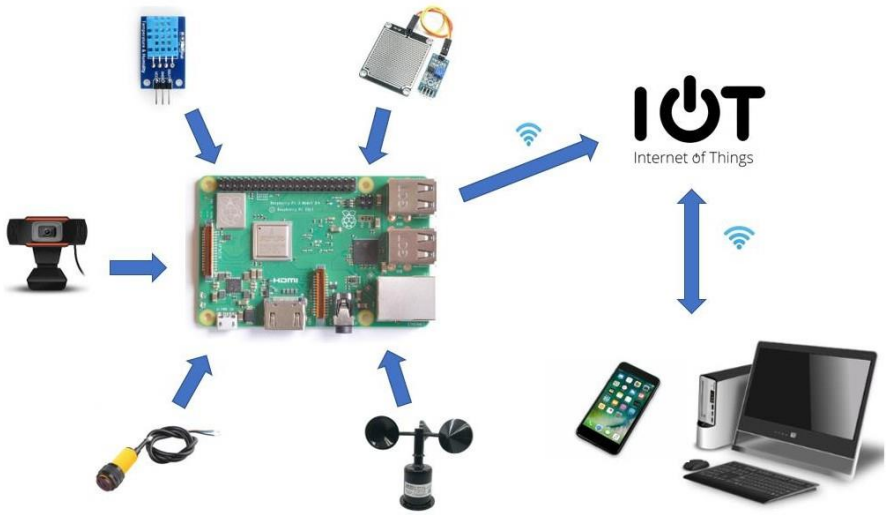


Figure 1.1: System design of weather monitoring system for drone docking station

1.2 Problem Statement

Drone nowadays been used for a autonomous mission such as highway traffic monitoring by a software such as autopilot, mission planner, drone harmony and others. However, for such mission the pilot needs to know the weather information at drone docking before launched or taking-off. For instances when there is strong wind, heavy rain, thunderstorm, hurricane or flash flood drone will not be released for flight. Else, the drone will be damaged, and the cost of maintenance to fix the damage will be significant. As a result, the project's goal is to develop a real-time weather monitoring system for drone docking station so that drone safety may be ensured before a mission starts. Hence weather information such as temperature, rain, wind speed and humidity are being monitored by sensors and IoT techology is utilized to gather all information including security using camera for the drone docking station. Figure 1.2 show general ideas for the electronics components used in this project.

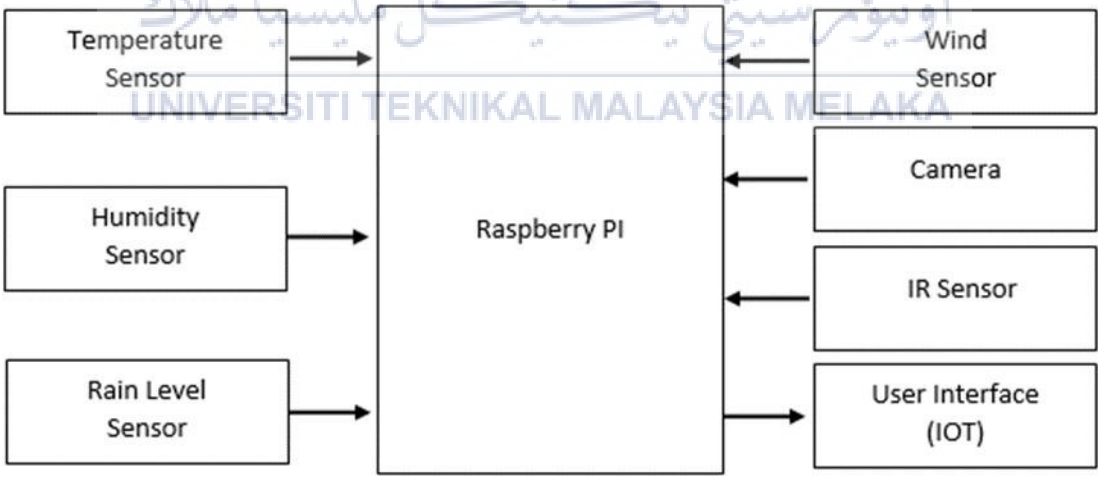


Figure 1.2: General ideas for the electronics components used in this project

Therefore, based on the above statement the IoT weather monitoring system needed to be developed with the suitable and affordable sensors and software that capable to execute weather monitoring from anywhere. All the consideration while develop this project will lead advantages to the weather monitoring capability. The enhancement and quality for the weather monitoring will ensure the system will gives the best accomplishment for weather station.

1.3 Project Objective

The following are the project's objectives:

a) To design weather monitoring system for drone docking station.

b) To develop a remotely monitored system using internet of thing (IoT) and display via web browser.

c) To analyze the performance of a weather monitoring system that display diverse weather informations such as wind, rain, humidity and temperature and security for the drone docking station.

1.4 Scope of Project

A few guidelines are proposed, by narrowing the needs for this project, to ensure that this project will achieve its objectives. The scopes covered are:

a) Raspberry PI is used as the main controller in this project.

b) The monitoring system consist of sensor such as wind, rain, humidity and temperature to monitoring the weather information at docking station.

c) A surveillance camera is installed to monitor drone docking station for security purposes.

d) This project used thinger.io as the IoT platform.

e) A graphic user interface (GUI) is designed for web-based application to monitor real time weather information system via smartphones and PC.

f) The entire test of the project will be made based on temperature and humidity of the environment, the presence of raindrops, wind speed, and system reliability performance.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The aim of weather monitoring system is to assist drone to dock better under weather changes. The controller system is required for it to control the sensor, python codding and etc. to develop the weather monitoring system. This project will use the interface by using the camera vision and Raspberry Pi controller for the measurement of variable sensor parameter thus the research will comprise these criteria so as to satisfy the objectives.

2.2 Weather Monitoring Station

Nowadays, weather monitoring system have advanced using recent technology to predict climate change. To do a weather monitoring system, a model of weather station must be built for gathering data from the sensor. Based on Mahmood et al., 2020, weather station is a tool to calculate weather variables to predict weather condition and analyze data. Alam et al.,2020 also describe in their journal that weather station gives information and forecast the weather conditions by analyzing the data using few methods.

This indicate that weather station compiles all the data from different sensor and do forecasting for weather based on the data. Weather station also store data that been compiled for the future prediction. It also used for monitoring and controlling of the weather conditions using the microcontroller (Mahmood *et al.*, 2020). From this, it shown that weather station important component that for monitoring, data gathering and forecasting weather for future use.

Weather station collect different types of weather information such temperature, wind, humidity, air and etc. to determine climate changes of the weather. This can be proving in Singh et al., 2020 journal where they said weather stations capable of collecting weather related information such as temperature, humidity, air pressure, wind speed, wind direction, visibility, particle levels and air etc.

In another studies (Math and Dharwadkar, 2019), weather station is developed to monitor three basic parameters such as rain, temperature and humidity to make it low cost-effective. From the statement we can see that parameter or data that been gather for weather station is climate related and depend on the cost project. If the cost project is high, more data can be collected, more parameter is need. Thus, this will make prediction much more accurate.

2.3 IoT (Internet of Things)

According to (Bhagat, Thakare and Choudhary, 2019), IoT connect various type of device to complete machine to machine interaction. In other words, IoT concept is connecting device to the Internet. This advances the intelligent level of devices when using Internet as the connector between two worlds without interaction of human. To connect the devices, sensor or actuator is needed. Tiwari et al., 2020 said, applications based on Internet of Things such as transducers and sensors attached to the microcontroller transform raw data to an important information that can be used anywhere. This sensor connected to IoT to get the raw data that needed in order to exchanges the information for future used. IoT connect and collect data from different devices (sensor) to distributed in many ways. This shows that different devices or sensor have different used, IoT get the data from sensor and upload in to Internet for the next process.

IoT also supports for more complex structures (Shewale and Gaikwad, 2017). With the used of IoT, more process can be done. Molnar et al., 2020 said that IoT collect, process, analyze and stored data real world and digital world. This conclude that IoT not only allowing communicate between devices and at the same times do various of process such as evaluates, interpretation, and analyze result.

IoT improves efficiency, accuracy, economic benefits along with reduced manpower (Vilayatkar *et al.*, 2019). More complicated process with highest chances of accuracy can be used using IoT. With technology around us it is not difficult to get further info through smart phones or websites. IoT makes certain application and devices level up to be smart devices for future. The various function of Internet of Things shown in Figure 2.1.

Figure 2.1: The various function of Internet of Things

2.4 Weather Monitoring System Using IOT

According to Muck and Homam, 2018, IoT can be more extensive in predicting and knowing weather condition for particular places just by connecting to Internet. This said that IoT can be used for predict and forecasting weather or climate changes for any places that can be connected to Internet and give a lot more convenience because no manpower needed to do this job. Moreover, using IoT in weather monitoring station gives a lot of advantages other than human interaction. In the Muck and Homam journal, they said that real time weather can be check at any places and anytime. This makes a lot easier to nowadays technology with 4G and 5G where people can check the weather using their phone and no need to go home just to check weather prediction using television.

Huang et al., also doing their research for weather monitoring system for city bus where they install the weather station at the bus to get prediction weather next day for their client can plan an activity a day ahead. This makes their client easier to check the weather and plan activities after they get the weather prediction. The client also can bring umbrella or make new plans if the next day weather is bad.

There is also a prototype project for agriculture field in India. A low-cost weather station that had been build allows farmers to carry out agricultural task at right time based on precision weather station that using IoT (Math and Dharwadkar, 2019). The research prototype that been done help farmer to forecast weather condition and situation for their agriculture. This project help in term of sustainable agriculture field and more profitable for farmers in India.

Microcontroller such as Arduino, Raspberry Pi or Node MCU is used for the low-cost weather monitoring system IoT. There are many related research and prototypes project that used this microcontroller for cost effective and combining the weather station to IoT. In table below shows that related work that used for weather monitoring system.

2.5 Comparison between each development

Table 2.1 shows comparison is based on IoT weather monitoring system development that have been done by previous researcher. Each developer uses different parameter of sensor, application of monitoring station and types of microcontrollers.

Table 2.1: Comparison is based on IoT weather monitoring system

Article	Parameter	Monitoring Station	Microcontroller
IoT Based Weather Monitoring and Reporting System Project (Bhagat, Thakare and Choudhary, 2019)	Temperature, humidity, moisture and rain level. Live reporting of weather conditions.	Web Server	Arduino UNO
Raspberry PI Based Weather Reporting Over IoT (Akilan *et al.*, 2020)	Rain level, pressure level, light intensity. Detect the weather patterns.	Sparkfun	Raspberry PI 3B

ESP 8266 Node MCU Based Weather Monitoring System (Mahmood *et al.*, 2020)	Temperature, humidity, and wind speed. Monitoring and controlling of the web server-based climate conditions	NET PI	ESP8266 NodeMCU
Weather Station IoT Educational Model Using Cloud Services (Molnár *et al.*, 2020)	Temperature, humidity, and pressure	Thingspeak	Arduino MEGA 2560
IoT Based Weather Station Using Raspberry Pi 3 (Muck and Homam, 2018)	Temperature, humidity, atmospheric pressure, light intensity, wind speed and direction, and rain precipitation.	Google Cloud SQL	Raspberry PI 3 & Arduino UNO R3
Low Cost IoT Based Weather Station for Real – Time Monitoring (Alam *et al.*, 2020)	Wind speed, temperature, humidity and rainfall	ThingSpeak	Arduino UNO NodeMCU
Low Cost IoT enabled Weather Station (Singh, Jerath and Raja, 2020)	Wind speed, wind direction, temperature, humidity	ThingSpeak	NodeMCU
An IoT Based Real-time weather monitoring system using Raspberry Pi (Shewale and Gaikwad, 2017)	Temperature, humidity, pressure, altitude, light intensity and rain water level.	Raspbian	Raspberry PI 1 Model B

IoT Based Low-cost Weather Station and Monitoring System for Precision Agriculture in India (Math and Dharwadkar, 2019)	Temperature, humidity, dew point, absolute pressure, relative pressure, light intensity and rain fall amount in real-time.	ThingSpeak	ESP32
IoT Based Weather Monitoring System using Raspberry Pi (Vilayatkar *et al.*, 2019)	Air temperature, air humidity, light, soil moisture, and rain detection.	Raspbian	Raspberry Pi Model 2 B
Weather Monitoring System Using IoT and Cloud Computing (Tiwari *et al.*, 2020)	Temperature and humidity. Use for detecting and forecasting weather	Cloud Server	Arduino UNO

2.6 Summary

Based on the previous research this project purposed a weather monitoring system using IoT for drone docking station using Raspberry PI as the main controller. This system comprises of sensors such as anemometer to read wind speed, rain sensor to monitor clear or rainy days, humidity sensor to identify if there is a fog around the drone docking station, and temperature sensor to monitor temperature surrounded drone docking station. In addition, a camera is installed for security and for surveillance monitor and infrared sensor is installed to detect drone that docking.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The performance system is split into a few stages to ensure that this project is completed successfully. The flowchart of this weather monitoring system is given in Figure 3.1.

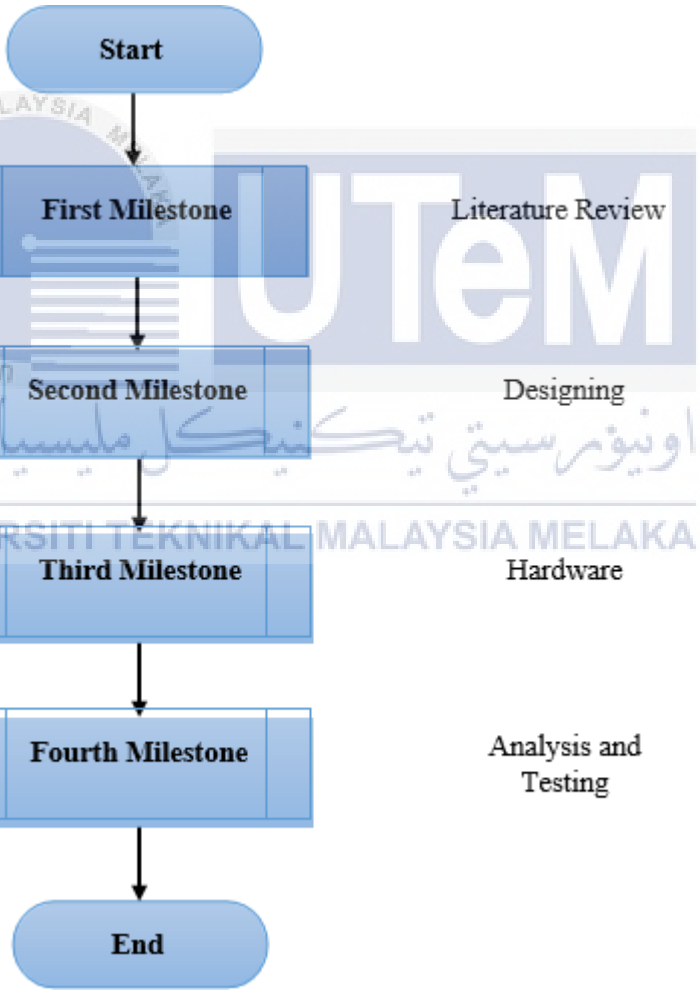


Figure 3.1: Flowchart of methodology

3.2 First Milestone

Firstly, this project's objectives were determined with the supervisor. The literature review for this project is completed when all of the objectives have been determined. The goal of a literature review is to have a general understanding of past research conducted by scholars or institutions. Ideas and information regarding the component used, solutions to manage the problem, and the analytic approach were gathered from previous studies. Then, the supervisor approved the scopes that will cover the objectives. The flow chart of literature review as shown in Figure 3.2.

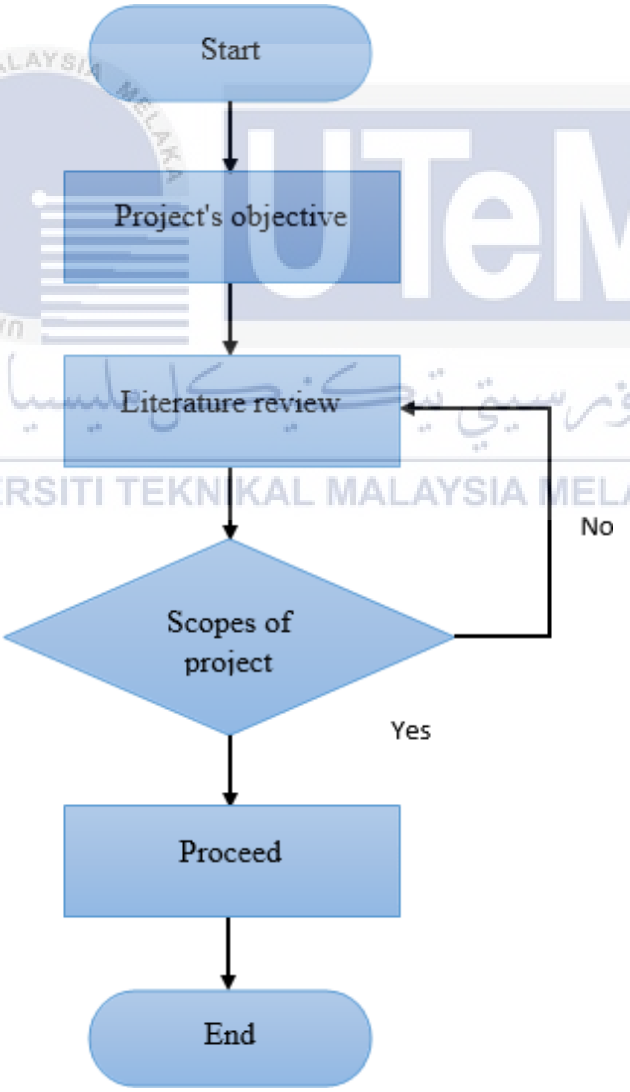


Figure 3.2: Flowchart of literature review

3.3 Second Milestone

The design and building circuit of an IoT weather monitoring system were demonstrated in this section. The design for wiring is using Fritzing and hardware is Auto CAD.

3.3.1 Design and Construction of IoT weather monitoring system

Microcontroller, power supply, input devices, output devices, and internet method are among the basic components of an IoT device. Weather monitoring and reporting solution for the IoT (Internet of Things) that allows you to gather, process, analyze, and publish your recorded values on a web server. The microcontroller communicates through unicast communication, sending commands to a single address across a single network that is requested, and responding or acting in compliance with the command. Sensors are in charge of gathering data and transmitting it to the IoT server. It will extract data from the sensor network after analyzing and packaging it into format data before sending it to the server. Servers are devices that run server software and are used to manage network resources. The system with the ability to deliver data acquisition on wind, rain, humidity and temperature sensors attached is the design and operation of a weather monitoring and controlling system. To make the system more intelligent, flexible, and efficient, the services or information given via the Internet are connected through Wi-Fi and made available to users via computer, smart phones, or other web browser devices. Almost everything in this system is automated, so it can take advantage of real-time direct parameter measurement. Fritzing Raspberry PI Weather Station shown in Figure 3.3.

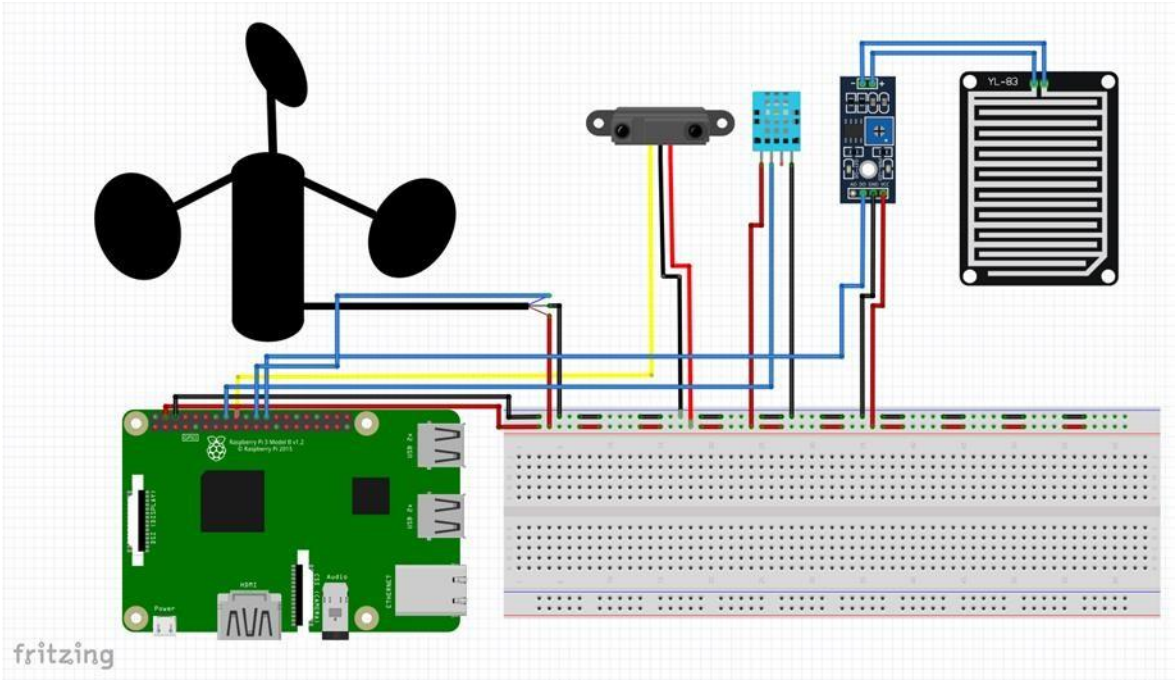


Figure 3.3: Fritzing Raspberry PI Weather Station

3.3.2 Design of base for weather monitoring drone docking station

Some specification need to consider to design base for weather monitoring drone docking station. Material type and the size of weather monitoring drone docking station need to be calculated and need to do thorough checking in order for the base to be stable, weather resistant and can accomodate drone weight. The material used is the most important part to design the weather monitoring drone docking station. The aluminium is choose based on the good durability material, light, portable and easy to take anywhere. The weathermonitoring drone docking station easy to carry around if we face bad weather such as heavy rain or flood. Size of the weather monitoring drone docking station also one of the main part in designing.

The lenght for the weather monitoring sensor is only is 150cm, 120cm in lenght for the drone docking, and 50cm x 50cm for the base of weather monitoring drone docking station. The size for weather monitoring drone docking station is suitable for Malaysian's height, not to tall and not to big so easier to maintenance and carried if facing with bad weather. Figure 3.4 shows AutoCAD of weather monitoring drone docking station.

Figure 3.4: AutoCAD of Weather Monitoring Drone Docking Station.

3.4 Third Milestone

This part discusses about the hardware of the circuit and the controller will be used in this project.

3.4.1 Raspberry Pi 3 Model B+

The Raspberry PI 3 is a PI series development board. It's a LINUX-based single-board computer. The board not only offers a lot of functionality, but it also has a fast processor, making it ideal for complex applications. The Raspberry PI board was created with engineers and hobbyists in mind who are interested in LINUX systems and IoT (Internet of Things). Even if the overall applications of PI are less, the Raspberry PI platform is more favored for developing complex applications. Furthermore, the Raspberry PI is an open-source platform that provides a variety of associated data, allowing you to customize the system to your own needs.

Because the system processing is so large, Raspberry PI 3 was chosen above other microcontrollers and development boards. The clock speed on most ARDUINO boards is less than 100MHz, they can only accomplish tasks that are within their capability. They are unable to handle high-end programmed for applications such as weather stations, cloud servers, gaming consoles, and etc. Raspberry PI can accomplish all of these complex functions because to its 1.2GHz clock speed and 1 GB RAM. When you need wireless connections, the Raspberry PI 3 features wireless LAN and Bluetooth capabilities, allowing you to create a WI-FI hotspot for internet access.

This functionality is ideal for the IoT (Internet of Things). The RASPBERRY PI has a dedicated connection for connecting a touch LCD display, which eliminates the requirement for a monitor and allows users to attach a camera to the PI board without any trouble. PWM outputs are also available on the Raspberry PI for application use. Raspberry PI 3 Model B+ with pinout shown in Figure 3.5.

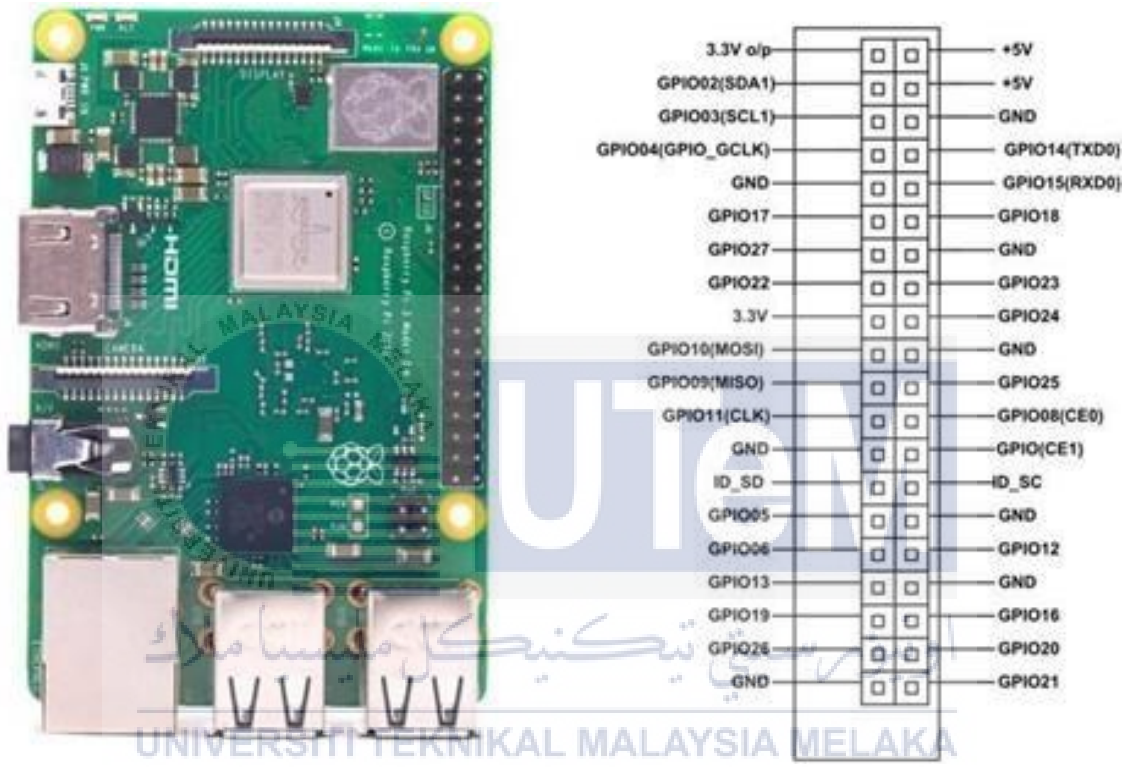


Figure 3.5: Raspberry Pi 3 Model B+ with pinout

Table 3.1 shows three microcontrollers have been chosen to be compared based on previous research and related work that have been reviewed. These microcontrollers have their advantages and disadvantages based on article from (Shah, 2018).

Table 3.1: Advantages and disadvantages of microcontrollers

Microcontroller	**Advantage**	**Disadvantage**
Arduino	- Affordable. - Cross-platform. - Programming environment that is easy to understand. - Software and hardware that are open source and adaptable.	- Memory limits - Less efficient - The microcontroller has less processing than the power processor. - Some tasks, such as scheduling and database storage, necessitate effort.
Raspberry PI	- Extremely powerful, with enough of memory and computing capability. - There are a lot of GPIOs accessible, and the more GPIOs you have, the more sensors you can connect. - Use many programming languages that can be used to program the Pi that any other microcontroller can. - Can be linked to a variety of applications.	- To get things moving, you'll need a basic understanding of Linux systems. - Because we will simply use it to transfer data across, the processing power will be an overkill for most of the applications. - This is a closed source. - Power hungry.

NodeMCU	- Compared to Arduino, ithas built-inWi-Fi capabilities. - Cost-effective. - WIFI network support is built-in. - Low usage of energy.	- Because it's a 3.3V device, it might not work with all peripherals. - Official documentation is lacking. - The Wi-Fi code consumes a significant amount of CPU time.

Advantages Raspberry PI over Arduino on Table 3.2 below. NodeMCU is not being compared due NodeMCU itself are similar to Arduino but consist Wi-Fi module.

Table 3.2: Advantages Raspberry PI over Arduino

Raspberry PI	**Arduino**
Multitasking and complex project suitability.	Runs only one application at a time, making it ideal for repetitive tasks.
Good for software projects.	Good for hardware projects
There are no restrictions on programming languages.	Arduino IDE and C/C++ are the only options.

Ethernet port built-in for networking.	External hardware must be connected, and coding must be implemented.
Assist other computers and connected devices by acting as a server.	Controls small devices such as sensors, motors, and lights exceptionally well.

3.4.2 DHT11 Sensor

A sensor or a module of the DHT11 sensor can be obtained. The sensor will be packaged in a four-pin package, while the module will have three, however only three pins will be used. The filter capacitor and pull-up resistor will be built into the module. If you need to use the sensor, you must do so externally.

The DHT11 is a temperature and humidity sensor that is widely used. A dedicated NTC for temperature measurement is included with the sensor. Temperature and humidity values are output as serial data by an 8-bit microcontroller. The sensor is factory calibrated, making it simple to connect to other microcontrollers. The sensor can detect humidity ranges from 20% to 90%, with a precision of ±1°C and ±1% with temperatures ranging from 0 °C to 50 °C. So, if you're trying to measure in this range, this sensor might be the best option. DHT11 temperature and humidity sensor with sensor pinout is shown in Figure 3.6.

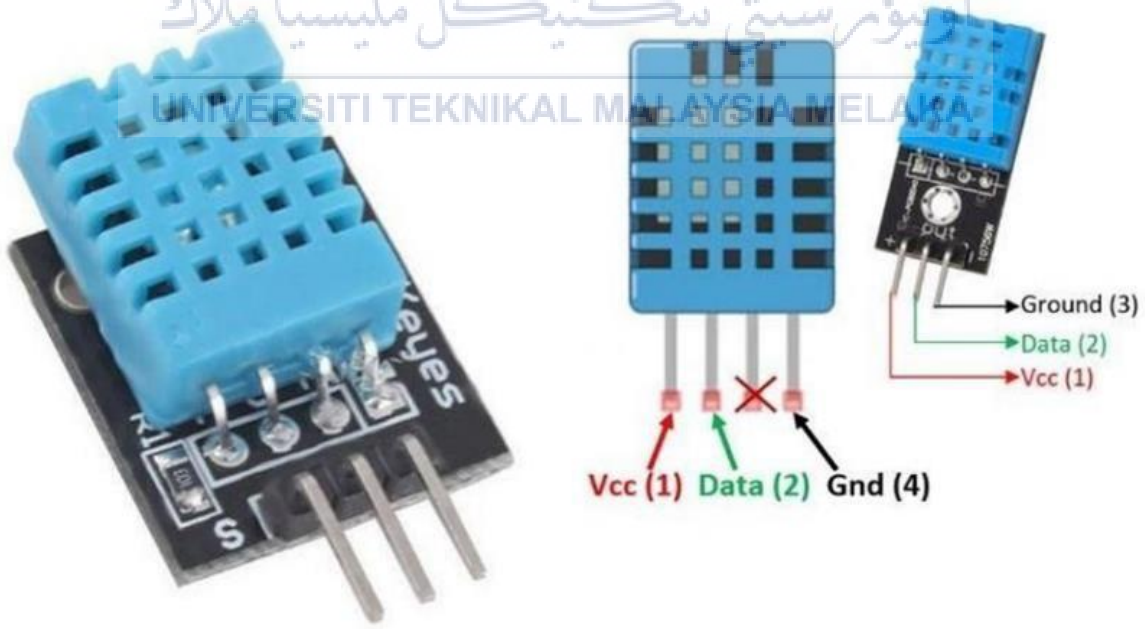


Figure 3.6: DHT11 Temperature and Humidity Sensor with sensor pinout

3.4.3 Raindrop Sensor FC-37 Module

A raindrop sensor is a device that detects raindrops. It is made up of two modules which consist a control module that controls the rain which compares and converts the analogue value to a digital value and a rain board that detects the presence of rain. Raindrop sensors can be used in the vehicle industry to automatically regulate the windshield wipers, it's used in agriculture to detect rain, and it's also found in home automation systems. It's easy to connect the raindrop sensor to a microcontroller like the Raspberry PI, Arduino, or PIC. The control module of the raindrop sensor is connected to the rain board module.

The raindrop sensor's control module has four outputs. The VCC pin of the module is connected to a 5V supply, while the GND pin is connected to the ground. For digital output, the D0 pin is connected to the microcontroller's digital pin, alternatively the analogue pin can be used. The A0 pin of a microcontroller can be linked to the ADC pin to utilize the analogue output. A potentiometer, LN393 comparator, LEDs, capacitors, and resistors are all included in the sensor module. The copper rails on the rain board module operate as a variable resistor. Its resistance changes depending on how wet the rain board is. Raindrop sensor FC-37 module with sensor pinout is shown in Figure 3.7.

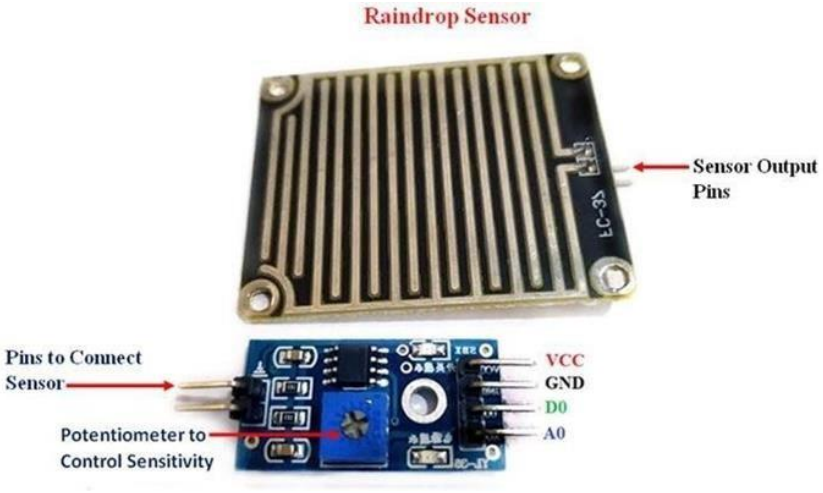


Figure 3.7: Raindrop Sensor FC-37 Module with sensor pinout

3.4.4 Wind Speed Anemometer NPN Output

Wind speed is one of the elements measured at weather stations, and it is one of the most essential determinants for human life and food security. Wind speed detector knowns as anemometer. It's lightweight, portable, and simple to assemble, and it's constructed of polycarbonate, which resists corrosion and rust while also combating the ageing issues associated with ABS plastic. The equipment's bottom outlet and outlet side are compatible with each other. In greenhouses, environmental protection, weather stations, ships, docks, and breeding habitats, the output signal pulse (NPN/Open-drain output) is frequently utilized in wind speed monitoring.

It weighs approximately (336g x 11.9oz) and measures approximately 19x17.5x17cm (7.5x6.9x6.7in) in size. It worked perfectly when Arduino, Raspberry PI or other microcontrolleris connected 5 volts with sensor and the power source by default is 10 to 30 volts DC. The operating humidity range of 0% RH to 80% RH, with a temperature range from -20°C to +60°C. The wind speed ranges from 0 to 70 meters per second, with a minimum measurable wind speed of 0.2 meters per second. The output is NPN or open drain, with a maximum current sink of 100mA. Wind Speed Anemometer NPN Output is shown in Figure 3.8.

Figure 3.8: Wind Speed Anemometer NPN Output

3.4.5 Digital Infrared Sensor

The electronic component such as IR sensor or infrared sensor is used to detect specific characteristics in its surroundings through emitting or detecting IR radiation. To detect the heat of a target and the motion, IR sensors can be used to measured it. This sensor similar to human's visionary senses when detect obstacles. The sensor which simply measures IR through radiation that are not visible to the eyes but can be sensed using IR sensors. By using the sensor, the photodiode is used as a detector meanwhile an IR LED is used as an emitter. The output voltage & resistance will be changed in proportion to the received IR light magnitude once an infrared light drops on the photodiode.

Generally, the purpose of digital infrared sensor is offered up to 30cm sensing distance. It is a non-contact detection by using a mobile robot for obstacle detecting and can be used for an automation machine. The sensing distance can be adjusted manually. It uses diffuse reflective type and input voltage from 6 to 36 VDC. NPN output with dimension of 1.6cm (D) x 6.7cm (L). The implementations of modulated IR signals immune the sensor to the interferences that caused by the normal light of a light bulb or the sunlight. Digital infrared sensor shown in Figure 3.9.

Figure 3.9: Digital Infrared Sensor

3.4.6 HD Webcam

HD Webcam that been used in this project are high quality with exquisite appearance. You can adjust the angle as you like, up & down 30 degrees rotatable with high definition with 12.0M pixels and true color images. Auto focus, don't need to adjust the lens. Imported optical lens, high precision and no distorted pictures. Compatible with USB2.0. Auto white balance, auto color correction. Designed for both laptop and desktop.

Superior quality glass lens with dynamic resolution of 640x 480. The specifications of HD webcam it is driverless DSP chip with CMOS image sensor. Item dimension: 8 x3 x 11cm / 3.15 x1.18 x4.33in. Package weight: 130g. Frame rate of 30fps and USB length 140cm. Support various video meeting software, like Meet meeting and works great with MSN, Yahoo and Skype etc. Support Windows 2000 / XP / win7 / win8 / win10/Vista 32bit/mac. HD Webcam shown in Figure 3.10.

Figure 3.10: HD Webcam

3.4.7 IoT Apps Thinger.io

Thinger.io provides an Internet of Things (IoT) platform that is customized to each project. The platform was created to allow data from sensors and actuators to be integrated into the internet. Data collection and storage from IoT devices has become a lot easier. Furthermore, developers can use the platform to create vertical applications without having to worry about hosting.

A cloud-based platform (SaaS) that can scale elastically and handle millions of devices and data streams. Easily manages data ownership for sharing and monetization, lowering the cost and risk of application development. As the platform improves, users can simply customize additional features for a variety of business applications and market segments. You have the option of using public or private cloud hosting to personalize your solution based on your requirements. Thinger.io application for IoT shown in Figure 3.11.

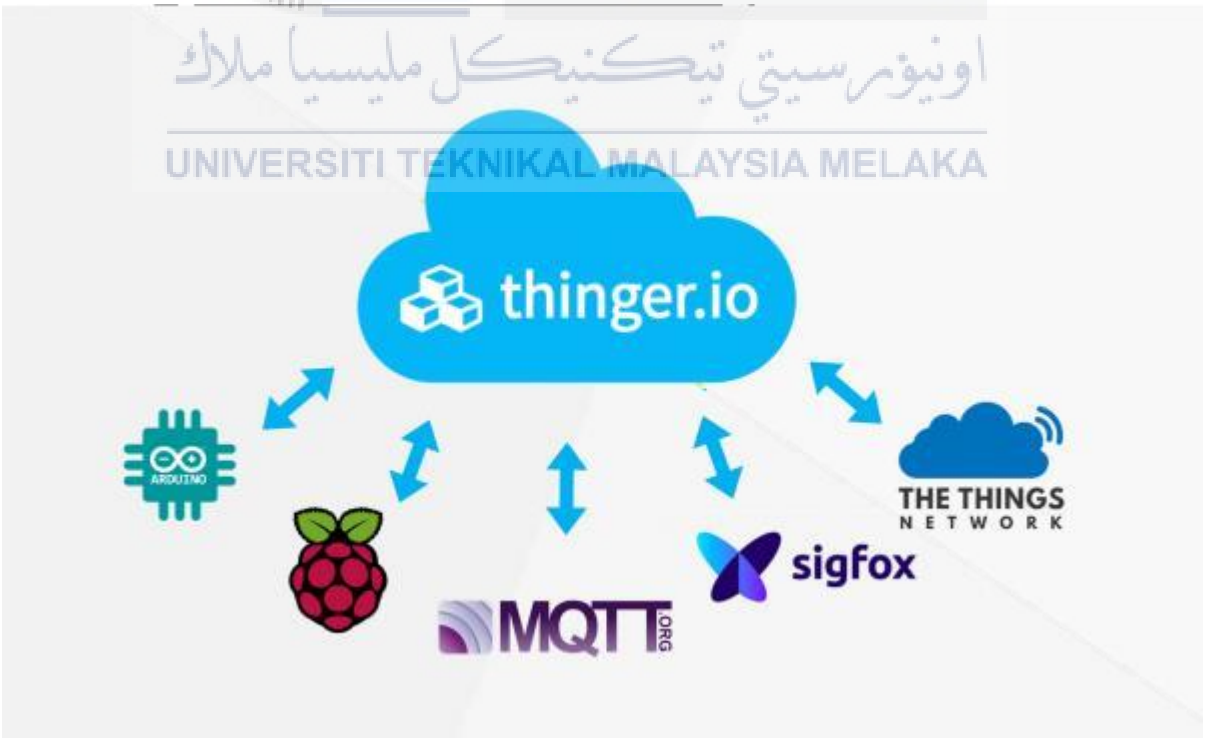


Figure 3.11: Thinger.io application for IoT

3.5 Fourth Milestone

In this section, the analysis and testing method is being explain. The process begins with the testing of the component functionality and analysis of the data.

3.5.1 Analysis and testing of the IoT monitoring weather station

Analysis and testing purpose is to system reliability performance of the IoT monitoring weather station. The different type of sensor and different type of environment which the weather station needs to perform will be test. The process of testing is clearly shown in figure 3.12. The performance of weather station in term of detecting the sensor and the weather conditions will be observed. Then, the data will be analyzed to improve the monitoring weather station performance.

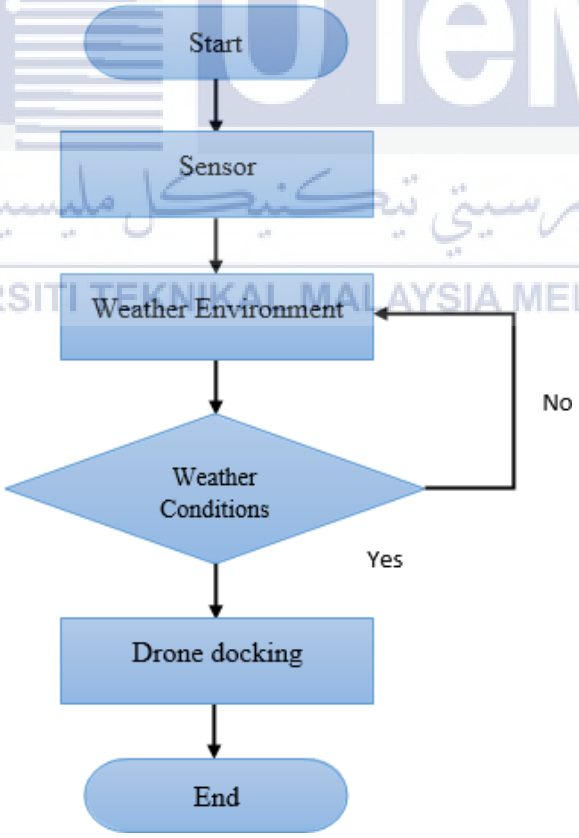


Figure 3.12: Algorithm of testing

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the results and analysis on the development of real time weather station monitoring system for drone docking station. This result to show the development of IoT weather station monitoring system result by using Raspberry PI 3B+ that connected to DHT11, raindrop sensor FC-37 module, digital infrared sensor to streaming the data simultaneously with Thinger.io and HD webcam to streaming video simultaneously with web browser. The weather station monitoring system located at location of urban area and estimated using the proposed approach over a 1-week period.

4.1.1 Streaming Raspberry PI 3B+ with all weather data to Command Prompt

To operate the project, need to connect to same internet network to receive the IP address. Open command prompt and ping 192.168.1.10 (Raspberry pi Ip address). The Ip address will receive and react to the ping and put “ssh pi@192.168.1.10” command to enter the raspberry pi. Enter password and it will show the raspberry pi is in online condition. After that to use the live video stream need to input “sudo motion” command and input “python3 weather_system_thinger.io.py” command to use the IoT application. The result will shown at command prompt. Thinger.io application for IoT shown in Figure 4.1.

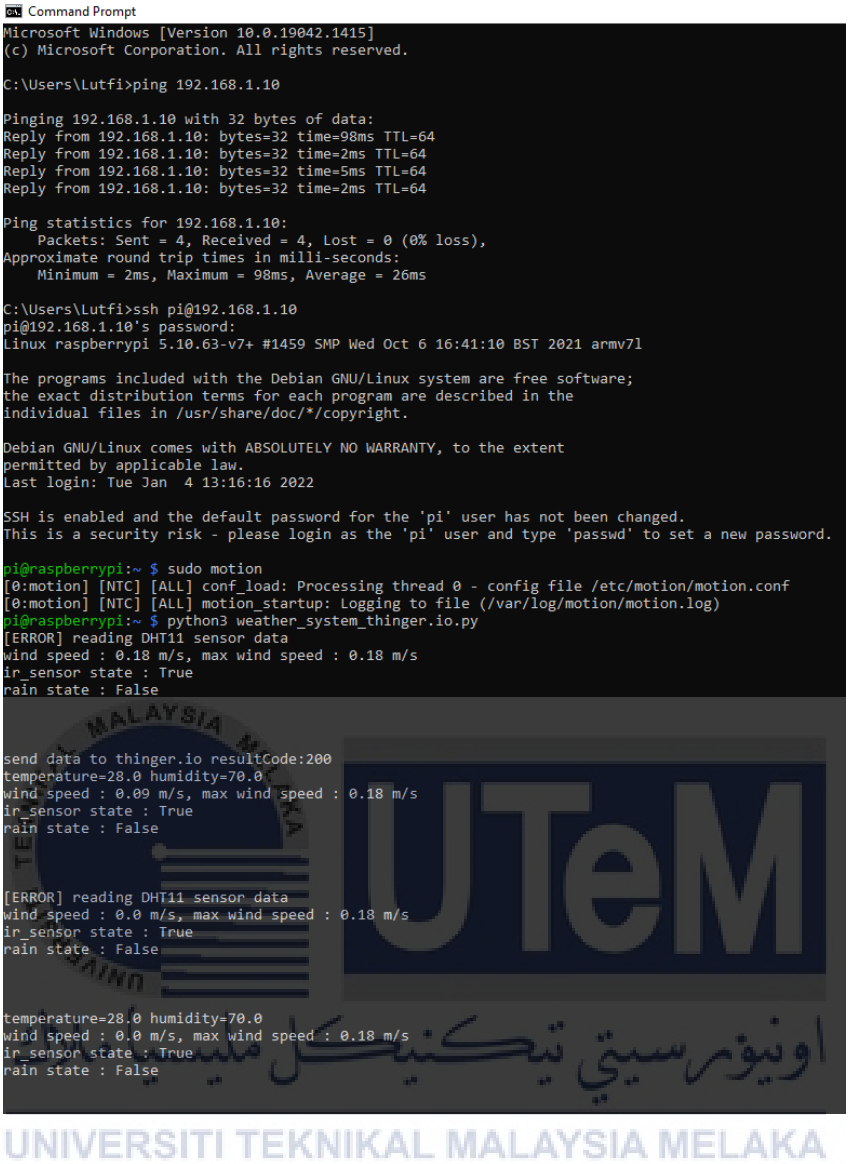


Figure 4.1: Weather data in Command Prompt

4.1.2 Streaming Raspberry PI 3B+ with DHT11 sensor to Thinger.io

The Raspberry PI 3B+ is sending the DHT11 data through Wi-Fi connection to the command prompt and it send to Thinger.io IoT. By using web browser to show the temperature and humidity that receive form DHT11 data that store to Thinger.io data bucket.

The various temperature and humidity data shown at Thinger.io in Figure 4.2 to Figure 4.5. The result shown that average temperature for sunny weather is around 34℃ while average humidity is around 44%. For rainy weather the average temperature is around 27℃ while average humidity for rainy weather is 85%. For cloudy weather, the average temperature is around 30℃ while the average humidity for cloudy weather is 68%. Lastly for windy weather, the average temperature 29℃ and the average humidity of windy weather is 78%.

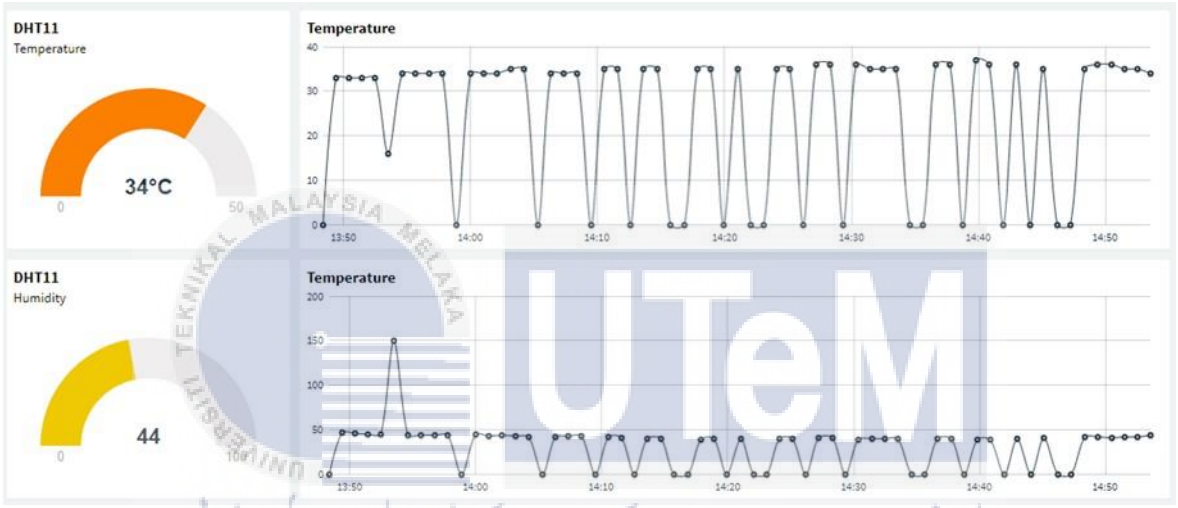


Figure 4.2: Sunny temperature and humidity data in Thinger.io

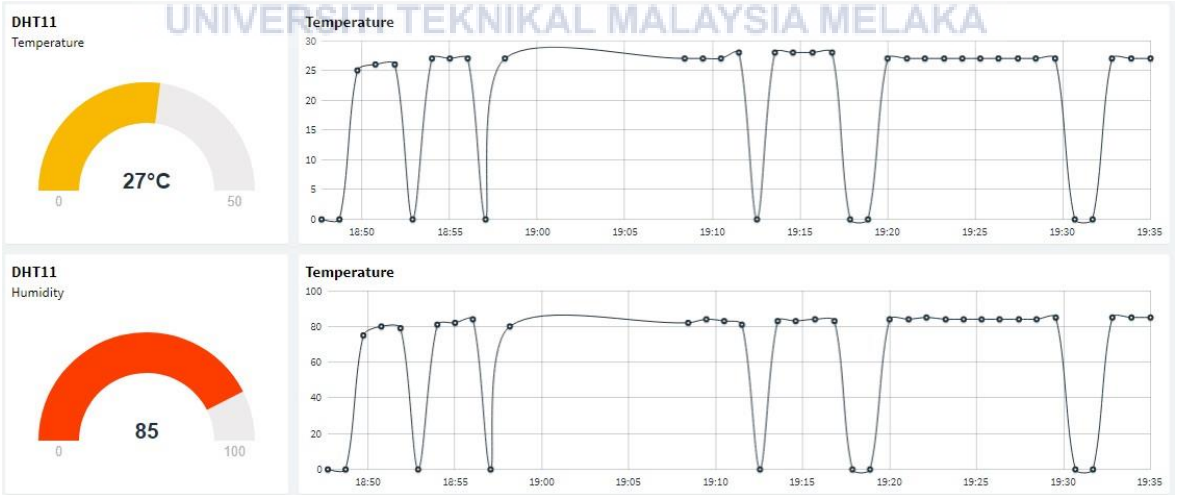


Figure 4.3: Rainy temperature and humidity data in Thinger.io

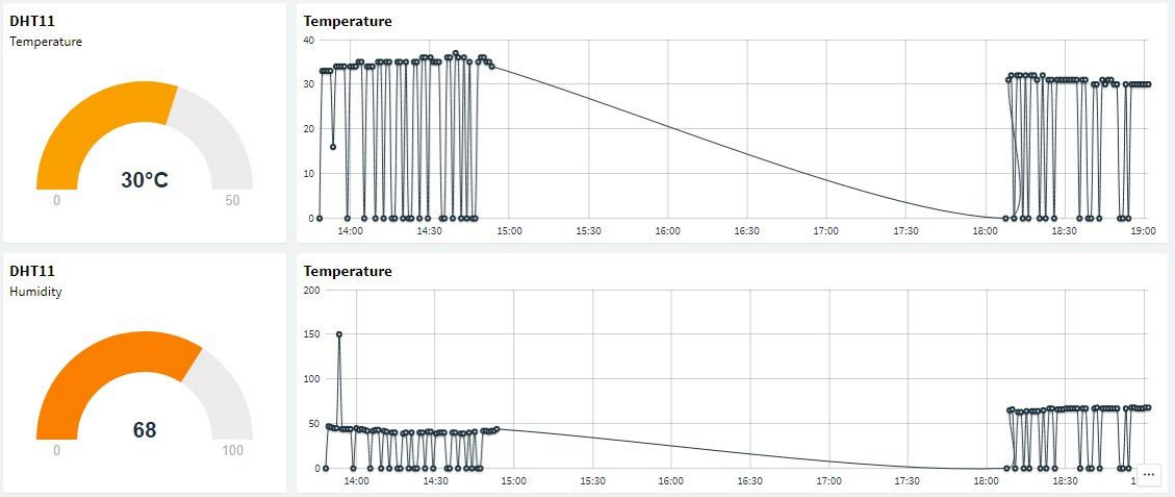


Figure 4.4: Cloudy temperature and humidity data in Thinger.io

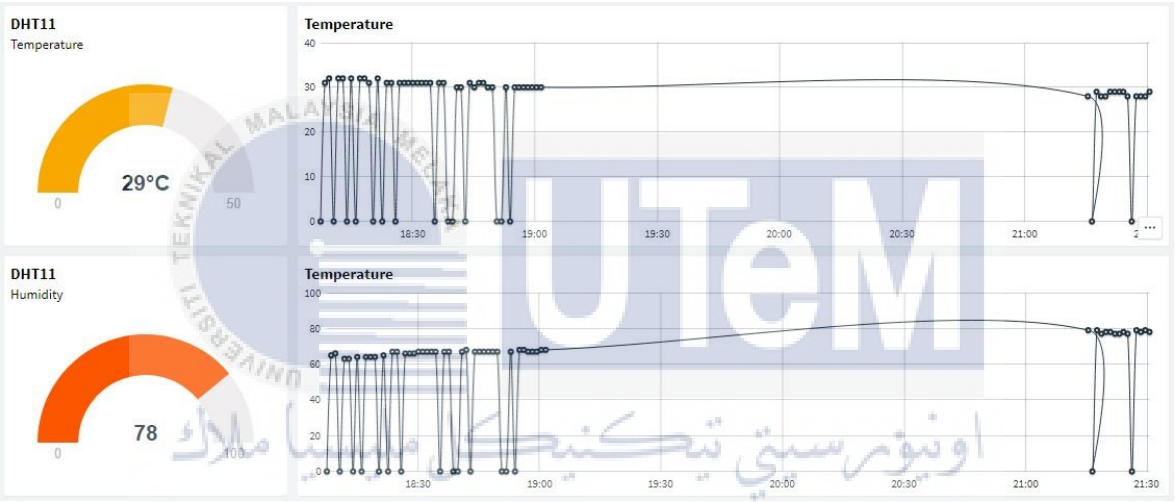


Figure 4.5: Windy temperature and humidity data in Thinger.io

4.1.3 Streaming Raspberry PI 3B+ with Wind Speed sensor to Thinger.io

The Raspberry PI 3B+ is sending the Wind Speed data through Wi-Fi connection to the command prompt and it send to Thinger.io IoT. By using web browser to show the speed of wind that receive from Wind Speed Anemometer data that store to Thinger.io data bucket. The various wind speed data shown at Thinger.io in Figure 4.6 to Figure 4.9.

The result shows that wind speed data is different based on the weather. For example, sunny weather shows that wind speed data is around 1.5m/s to 3.5m/s. For rainy weather, the wind speed data is around 2.0m/s to 3.5m/s while for cloudy weather the wind speed data is around 1.0m/s to 2.5m/s. For windy weather the wind speed data is also different which is around 3.0m/s to 8.0m/s.

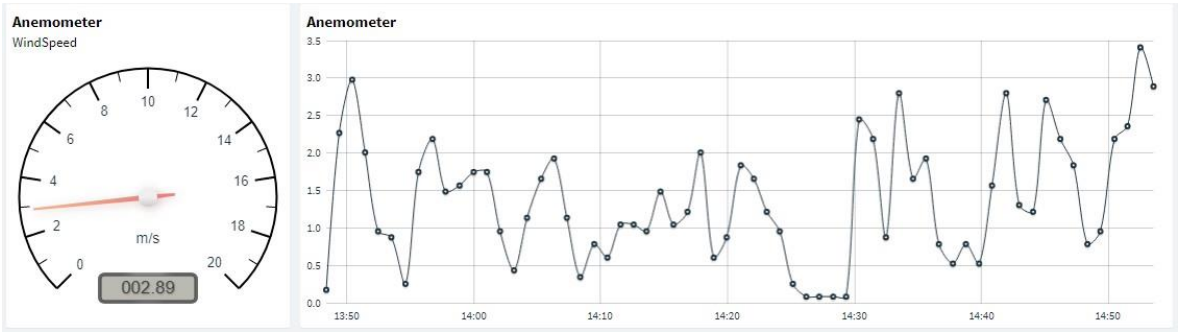


Figure 4.6: Wind speed data from sunny weather in Thinger.io

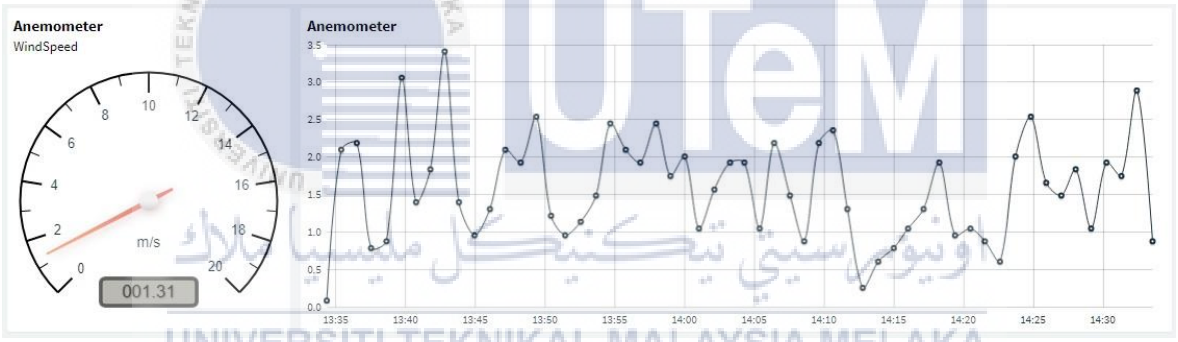


Figure 4.7: Wind speed data from rainy weather in Thinger.io

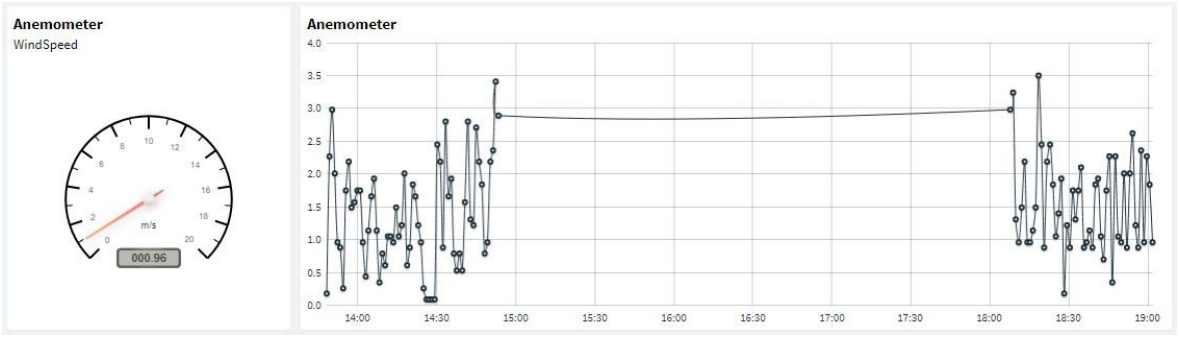


Figure 4.8: Wind speed data from couldy weather in Thinger.io

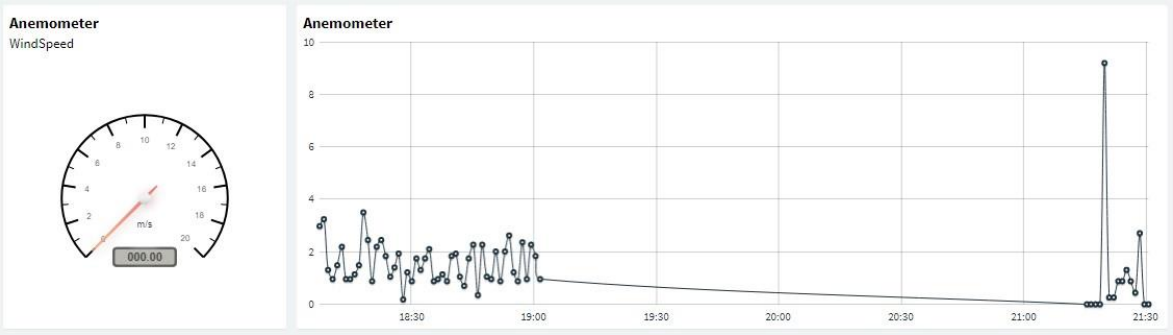


Figure 4.9: Wind speed data from windy weather in Thinger.io

4.1.4 Streaming Raspberry PI 3B+ with Raindrop and Digital IR sensor to Thinger.io

The Raspberry PI 3B+ is sending the Raindrop and Infrared data through Wi-Fi connection to the command prompt and it send to Thinger.io IoT. By using web browser to show the raindrop and drone detection that receive from Raindrop and Digital IR data that store to Thinger.io data bucket. The raindrop and drone detection data shown at Thinger.io in Figure 4.10 and Figure 4.11.

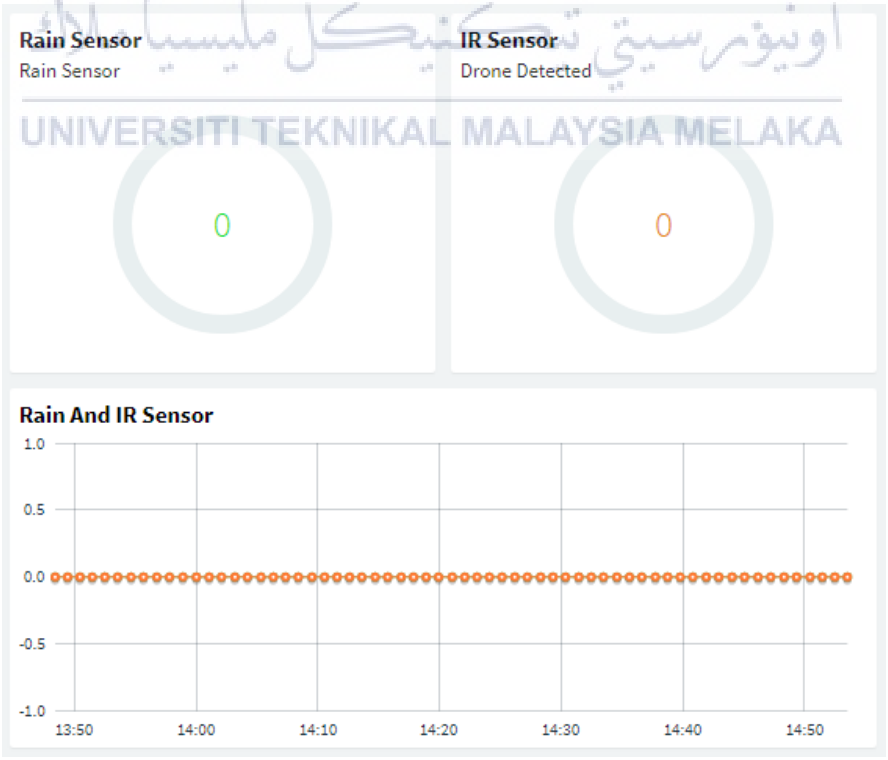


Figure 4.10: Raindrop and drone not detected data in Thinger.io

Figure 4.11: Raindrop and drone detected data in Thinger.io

4.1.5 Streaming Raspberry PI 3B+ with HD Webcam to Web Browser

The HD Webcam that connected to Raspberry PI 3B+ is act as surveillance camera for the weather monitoring drone docking station. The webcam uses to show the live streaming of the area of the drone docking. Web browser as platform to stream the live streaming video. To access the video streaming need to input IP address of the Raspberry PI and add ":8081" port. The HD Webcam act as surveillance camera shown in Figure 4.12. HD Webcam display live streaming video that shown toy car that acts as subtitute of drone.

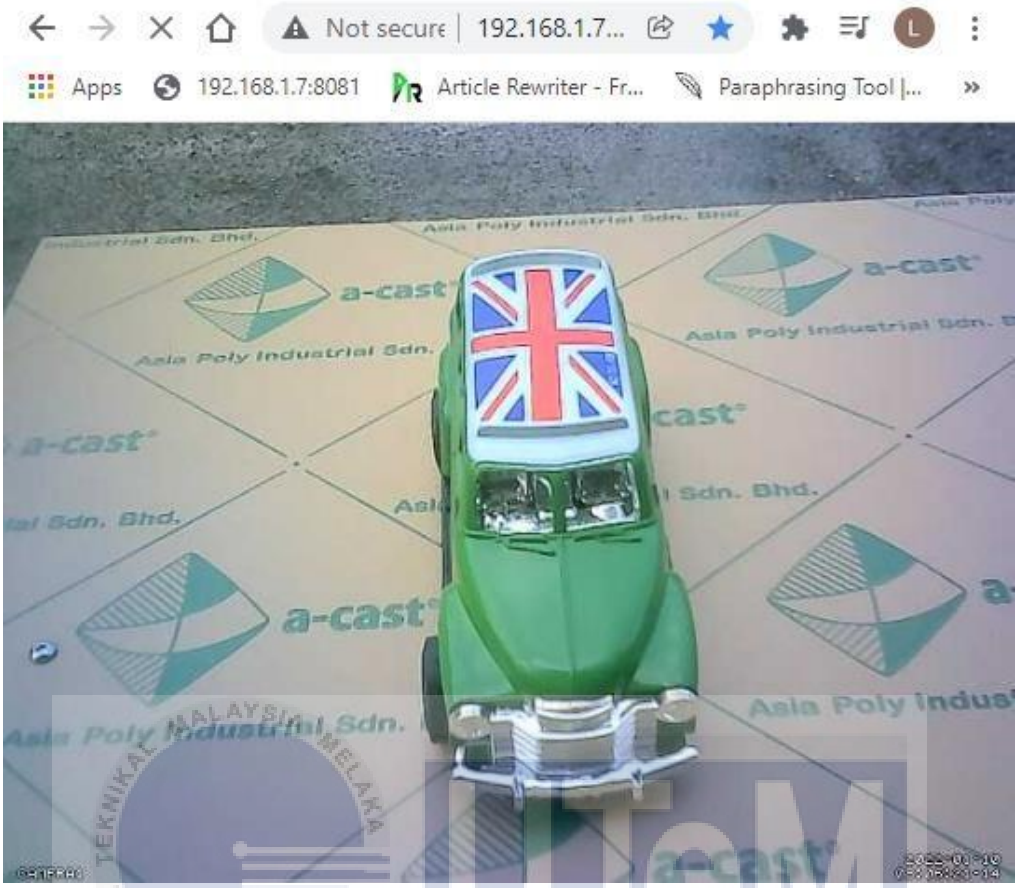


Figure 4.12: HD Webcam show toy car that display at live streaming video

4.2 Summary

This chapter presented result from the IoT weather monitoring system for drone docking station. The analysis result shows from connection of Command Prompt with Raspberry PI 3B+ that represent IoT that connected to DHT11 to collect average temperature and average humidity, wind speed to collect wind speed data, raindrop sensor FC-37 module and digital infrared sensor to streaming the data simultaneously with Thinger.io in order to collect rain drop and HD webcam to streaming video simultaneously with web browser and act as surveillance camera. Full data of weather station in Thinger.io IoT shown in Figure 4.13.

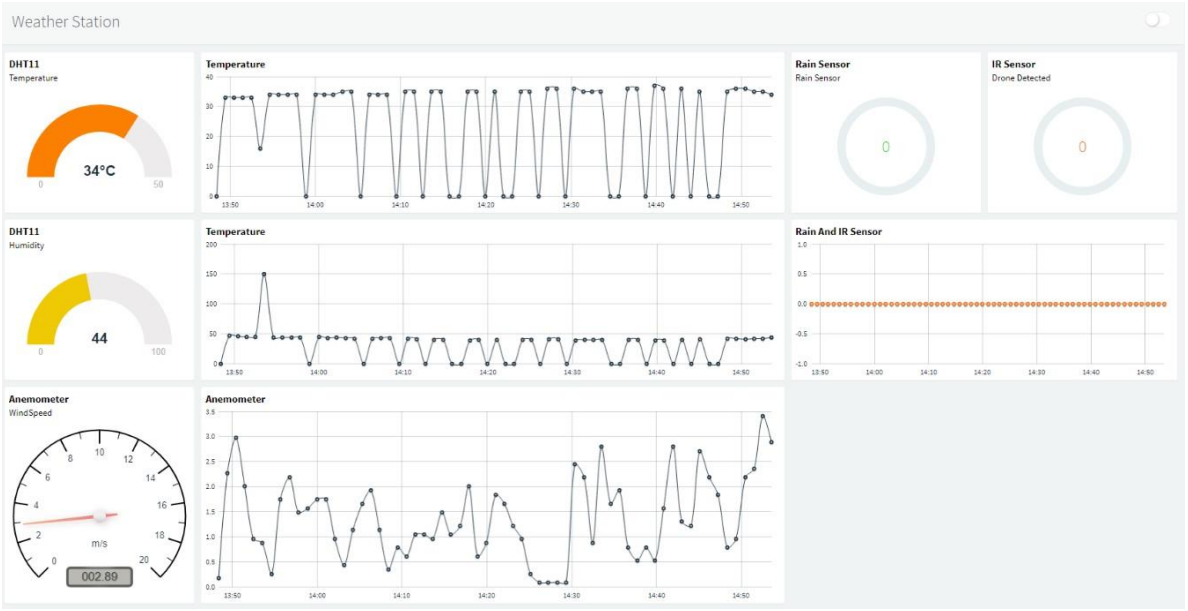


Figure 4.13: Full data of weather station in Thinger.io IoT

All the data is collected using Thinger.io and data store and using web browser to show the result of weather forecasting and monitoring system for drone docking. The data is stored and collected for weather station are wind, rain, humidity and temperature that detected using sensor. From the result and data collected, we can further forecast or predict the weather. This is important in order to keep drone from damage before or after docking. The weather station monitoring system located at urban area and estimated using the proposed approach over a 1-week period. The result shows that the weather monitoring system for drone docking is functioning well and have achieve an expected results.

Figure 4.14: Weather Monitoring Drone Docking Station

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This report presents a weather monitoring system for drone docking station using IoT. In this project, user can access data anywhere in real-time by using Internet of Things (IoT).

If the weather shows good, the drone can dock but if the weather is bad, the damage that might be happen to the drone can be avoid. User can get the information of current weather if it safe to send the drone to dock once a weather station is connected. User can view the history of information, and can figure out the trends in the measurements as well. The design of the weather station and the component needed is explained detailed in this project.

Result and analysis have also been done in order to test the weather station monitoring system for drone docking is function correctly. It have shows the expected result based on the data that have been collected. This project hopefully can help to overcome the problem and reduce the number of damages that might happen in order to make a new production more efficiently.

5.2 Future Works

For future improvements, a weather monitoring system for drone docking station could be enhanced to give more estimation results as follows:

i) The body frame of the weather station can be upgrade to steel to make it more sturdy and solid.

ii) Add lighting sensor to collect thunder storm weather data.

iii) Cover the sensor part with waterproof casing.

iv) Used PLC microcontroller with IoT based and industrial sensor for commercialization.

v) Automatic weather station with charging station.

5.3 Project Potential

This project was planned and developed to achieve its purpose of serving as a weather monitoring station in order to reduce the risk of drone damage. It is designed to assist users in obtaining real-time weather information based on the sensors that have been placed, such as temperature, rain, humidity, and wind speed sensors. These sensors collect data and provide real-time weather to the user via the dashboard online. IoT will transform data for current weather and send it to the website for users to view. Users may utilise the weather monitoring system to current weather conditions and determine whether or not to fly their drone.

This prototype are portable, light, inexpensive, and simple to use. Based on the results, the prototype is usable, functioning well with the sensor, and capable of sending data to the online dashboard to provide real-time weather information. However, this idea may undoubtedly be improved by adding a lighting sensor to capture thunder storm meteorological data. This allows for the coverage of thunder storm weather data. In the future, the sensor section can be covered with a waterproof shell, and an autonomous charging station can be added to the weather station. By doing so, the sensor is shielded from rain and can charge the drone automatically while docked. As a result, the pilot is not required to come and change the battery because it has been handled automatically.

Furthermore, for commercialization, the prototype may be upgraded by integrating a PLC microcontroller with IoT-based and industrial sensors. The prototype can be used to monitor the weather and roadways in highway plazas. It may also be used for apartment security purposes, such as weather monitoring and surveillance. Finally, technology may aid agriculture by optimising agricultural operations, increasing crop yield, and monitoring crop growth. Sensors and digital imaging capabilities may provide farmers with a sharper image of their fields, allowing them to implement resource-efficient and precision farming strategies in real time.

REFERENCES

Akilan, T. et al. (2020) 'Raspberry Pi Based Weather Reporting over IoT', in Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020. Institute of Electrical and Electronics Engineers Inc., pp. 540–544. doi: 10.1109/ICACCCN51052.2020.9362971.

Alam, M. J. et al. (2020) 'Low Cost IoT based weather station for real-time monitoring', 2020 IEEE 2nd International Conference on Circuits and Systems, ICCS 2020, pp. 127–133. doi: 10.1109/ICCS51219.2020.9336596.

Bhagat, A. M., Thakare, A. G. and Choudhary, K. A. M. | N. S. M. | P. V. (2019) 'IoT Based Weather Monitoring and Reporting System Project', International Journal of Trend in Scientific Research and Development, Volume-3(Issue-3), pp. 365–367. doi: 10.31142/ijtsrd21677.

Mahmood, S. et al. (2020) 'ESP 8266 Node MCU Based Weather Monitoring System', Department of Computer Technical Engineering, Alkitab University, Iraq. doi: 10.4108/eai.28-6-2020.2298609.

Math, R. K. M. and Dharwadkar, N. V. (2019) 'IoT Based low-cost weather station and monitoring system for precision agriculture in India', in Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018. doi: 10.1109/I-SMAC.2018.8653749.

Molnár, J. et al. (2020) 'Weather station IoT educational model using cloud services', Journal of Universal Computer Science, 26(11).

Muck, P. Y. and Homam, M. J. (2018) 'IoT Based Weather Station Using Raspberry Pi 3', International Journal of Engineering & Technology, 7(4.30). doi: 10.14419/ijet.v7i4.30.22085.

Shah, H. M. (2018) Internet of Things – Three Popular Development Boards. Available at: https://blog.trigent.com/internet-of-things-three-popular-development-boards-trigent/.

Shewale, S. D. and Gaikwad, S. N. (2017) 'An IoT Based Real-Time Weather Monitoring System Using Raspberry Pi', International Research Journal of Engineering and Technology(IRJET), 4(4), pp. 3313–3316. doi: 10.15662/IJAREEIE.2017.0606009.

Singh, D. K., Jerath, H. and Raja, P. (2020) 'Low cost IoT enabled weather station', in Proceedings of International Conference on Computation, Automation and Knowledge Management, ICCAKM 2020. doi: 10.1109/ICCAKM46823.2020.9051454.

Tiwari, M. M. et al. (2020) 'Weather Monitoring System Using IoT and Cloud Computing', Bharti Vidyapeeth's College of Engineering, New Delhi, 29(12s), pp. 2473–2479.

Vilayatkar, S. R. et al. (2019) 'IoT Based Weather Monitoring System using Raspberry Pi', International Research Journal of Engineering and Technology (IRJET), 6(1), pp. 1187–1190.

APPENDICES

Appendix A

Codding of Weather Monitoring System Drone Docking Station

```
# run command > pip install request

import datetime, threading
import signal
import sys
import RPi.GPIO as GPIO
import time

import requests
import json
import time
import Adafruit_DHT as dht
import RPi.GPIO as GPIO

humidity = 0
temperature = 0

dht_pin = 23

ir_pin = 24
ir_State = 0
ir_lastState = 0

rain_pin = 8
rain_State = 0
rain_lastState = 0

wind_Pin =  25
wind_speed =  0
wind_max_speed_in_minute =  0
wind_pulseCount = 0

def signal_handler(sig, frame):
    GPIO.cleanup()
    sys.exit(0)

def pinInterrupt(channel):
    global wind_pulseCount
    wind_pulseCount = wind_pulseCount + 1
```

```
def wind_count():
    global wind_pulseCount
    global wind_speed
    global wind_max_speed_in_minute
    wind_speed = round((wind_pulseCount/20.0) * 1.75 , 2 )
    if wind_max_speed_in_minute < wind_speed:
        wind_max_speed_in_minute = wind_speed
    wind_pulseCount = 0
    threading.Timer(1, wind_count).start()

def update_thinger():
    global wind_max_speed_in_minute
    global humidity
    global temperature
    global rain_State
    global ir_State

    url =
"https://backend.thinger.io/v3/users/Lutfi25/devices/raspi_weather/callback/
data"
    headers = {
        "Authorization": "Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJEZXZpY2VDYWxsYmFja19yYXNwaV9
3ZWF0aGVyIiwic3ZyIjoiYXAtc291dGhlYXN0LmF3cy50aGluZ2VyLmlvIiwidXNyIjoiTHV0Zmk
yNSJ9.cuaOn1d0otlEiAgLtM6_L_MjpvcUBYfZ3wlDkzDsprw",
        "Content-Type": "application/json"
    }

    payload = json.dumps(
        {
            "temperature": temperature,
            "humidity": humidity,
            "ananometer": wind_max_speed_in_minute,
            "rain": rain_State,
            "ir_sensor": ir_State
        }
    )

    wind_max_speed_in_minute=0
    response = requests.request("POST", url, headers=headers, data=payload)
    print ("send data to thinger.io
resultCode:{}".format(response.status_code))

    threading.Timer(60, update_thinger).start()


if __name__== "__main__":
```

```
    GPIO.setmode(GPIO.BCM) # Broadcom pin-numbering scheme

    GPIO.setup(wind_Pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.add_event_detect(wind_Pin, GPIO.FALLING, callback=pinInterrupt,
bouncetime=1)

    GPIO.setup(ir_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup(rain_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

    threading.Timer(1, wind_count).start()
    threading.Timer(5, update_thinger).start()

    while True:


        humidity, temperature = dht.read_retry(dht.DHT11, dht_pin,
retries=5, delay_seconds=1)
        if all([humidity, temperature]):
            print('temperature={} humidity={}'.format(temperature,
humidity))
        else:
            humidity = 0
            temperature = 0
            print('[ERROR] reading DHT11 sensor data')

        ir_State = not GPIO.input(ir_pin)
        rain_State = not GPIO.input(rain_pin)

        print("wind speed : {} m/s, max wind speed : {}
m/s".format(wind_speed, wind_max_speed_in_minute))
        print("ir_sensor state : {}".format(ir_State))
        print("rain state : {}".format(rain_State))
        print("\n\n")
        time.sleep(1)
```

Appendix B

Codding of HD Webcam

```
import io
import picamera
import logging
import SocketServer
from threading import Condition
from PIL import ImageFont, ImageDraw, Image
import cv2
import traceback
import StringIO

import numpy as np

import datetime as dt

#import SimpleHTTPServer

from BaseHTTPServer import BaseHTTPRequestHandler,HTTPServer

PAGE="""\
<html>
<head>
<title>SkyWeather MJPEG streaming demo</title>
</head>
<body>
<h1>SkyWeather MJPEG Streaming Demo</h1>
<img src="stream.mjpg" width="1296" height="730" />
</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
```

```
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace;
boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                        # now add timestamp to jpeg #
                        Convert to PIL Image
                        cv2.CV_LOAD_IMAGE_COLOR = 1 # set flag to 1 to give
colour image
                        npframe = np.fromstring(frame, dtype=np.uint8)
                        pil_frame =
cv2.imdecode(npframe,cv2.CV_LOAD_IMAGE_COLOR)
                        #pil_frame = cv2.imdecode(frame,-1)
                        cv2_im_rgb = cv2.cvtColor(pil_frame,
cv2.COLOR_BGR2RGB)
                        pil_im = Image.fromarray(cv2_im_rgb)

                         draw = ImageDraw.Draw(pil_im)

                        # Choose a font
```

```
                        font =
ImageFont.truetype("/usr/share/fonts/truetype/freefont/FreeSans.ttf", 25)
                        myText = "SkyWeather
"+dt.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

                        # Draw the text
                        color = "rgb(255,255,255)"
                        #draw.text((0, 0), myText,fill = color, font=font)

                        # get text size
                        text_size  =  font.getsize(myText)

                        # set button size + 10px margins
                        button_size = (text_size[0]+20, text_size[1]+10)

                        # create image with correct size and black
background
                        button_img = Image.new('RGBA', button_size, "black")

                        #button_img.putalpha(128)
                        # put text on button with 10px margins button_draw
                        = ImageDraw.Draw(button_img)
                        button_draw.text((10, 5), myText, fill = color,
font=font)

                        # put button on source image in position (0, 0)

                        pil_im.paste(button_img, (0, 0))
                        bg_w, bg_h = pil_im.size
                        # WeatherSTEM logo in lower left
                        size = 64
                        WSLimg =
Image.open("WeatherSTEMLogoSkyBackground.png")
                        WSLimg.thumbnail((size,size),Image.ANTIALIAS)
                        pil_im.paste(WSLimg, (0, bg_h-size))

                        # SkyWeather log in lower right
                        SWLimg = Image.open("SkyWeatherLogoSymbol.png")
                        SWLimg.thumbnail((size,size),Image.ANTIALIAS)
                        pil_im.paste(SWLimg, (bg_w-size, bg_h-size))

                        # Save the image
                        buf= StringIO.StringIO()
                        pil_im.save(buf, format= 'JPEG')
                        frame = buf.getvalue()
                    self.wfile.write(b"--FRAME\r\n")
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
```

```
                    self.end_headers()
                    self.wfile.write(frame)
                    self.wfile.write(b'\r\n')
                except Exception as e:
                    traceback.print_exc()
                    logging.warning(
                        'Removed streaming client %s: %s',
                        self.client_address, str(e))
            else:
                self.send_error(404)
                self.end_headers()

class StreamingServer(SocketServer.ThreadingMixIn, HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

#with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
#with picamera.PiCamera(resolution='1920x1080', framerate=24) as camera:
with picamera.PiCamera(resolution='1296x730', framerate=24) as camera:
    output = StreamingOutput()
    camera.start_recording(output, format='mjpeg')
    camera.annotate_foreground = picamera.Color(y=0.2,u=0, v=0)
    camera.annotate_background = picamera.Color(y=0.8, u=0, v=0)
    try:
        address = ('', 443)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()
```

Appendix C