Development of IoT Based Weather Reporting System

Development of IoT Based Weather Reporting System

A F Pauzi[1] and M Z Hasan[1]

[1]Faculty of Engineering Technology, Universiti Malaysia Perlis (UniMAP), Malaysia

zamrihasan@unimap.edu.my

Abstract. Weather forecast these days is unpredictable too be exact because of the climate changes drastically over weather. In cause of that, Weather Reporting System is mostly used to monitor the continuously changing climatic and weather conditions over controlled areas likes house, industry, agriculture and etc. in real time monitoring. Internet of Things (IoT) platform use is ThingSpeak it's should be able displaying the weather parameters and the information will visible wherever in the world and it's also displaying on the OLED with two-way microcontroller communication via Wi-Fi hotspots. The condition of some particular place that be reported by satellite weather report system does not give the exact condition. However, the problem occurs when needed the accurate weather report for current time. With weather reporting system all weather parameters sensor will be controlled by ESP32 microcontroller as the server that will send all the data collected by sensors to the database by ThingSpeak and will visible anywhere in the world and also display on OLED that use Wemos D1 mini as its microcontroller and a client. This data then will be compared with the weather forecast data and statistics made by forecast station. All data collected will be also saved in google sheet format by IFTT tool for easier to analyse the data. This system will monitor the changes of weather condition happening over the environment and then provides the users fastest way to access the information from anywhere.

1. Introduction

Climate plays an important role in human life the unprecedented growth of industries and vehicular traffic have seriously affected the purity of clean air and environment [1]. Satellite weather report system gives condition of present which does not give the exact condition of the particular place. The building sector offers a great potential for the energy savings, where it is necessary to have accurate weather data in the exact location where the building is being built in order to improve the calibration of energy simulation programs [2]. By develop a controlling local weather reporting system with ESP32 and Wemos D1 mini microcontroller can minimize the error in weather forecast system at exact location. A precision agriculture and farming can be defined as the art and science of using technology to improve crop production [3]. Even though water is a scarce resource, overall 50% of water is wasted in agriculture due to the improper scheduling of irrigation [4]. In this context, the real-time monitoring of water usage in the fields can prevent misuse of water [4].

Use of technology in the field of agriculture plays important role in increasing the production as well as in reducing the extra man power efforts, some of the researches tried for betterment of farmers and provides the systems that use technologies which are helpful for increasing the agriculture yield [5]. Difficulty to monitor weather parameters through offline system such as agriculture zone during certain hazardous envy and critical situations where the people need to check manually the weather condition at the places and it will take time unless it is online system. In the evolving generation of wireless technology, the concept of smart cities and IOT has given a new remark in the world. One such remark leads towards the online smart weather station system [6]. The weather parameters should be able displaying, analysing and monitoring system using ThingSpeak that connect user with internet that visible anywhere in the world. To analyse and monitoring system using ThingSpeak that connect user with internet that visible anywhere in the world. Internet of Things (IoT) is playing a leading role in providing solutions to many applications with the support of software, internet and embedded systems [7].

There is a necessity in security or alarming system that give warning and alerting when there is a bad condition at the place. The existing technologies are developed using microcontrollers like Arduino, Node MCU etc. and ARM processors like Raspberry Pi [7]. So, machine learning techniques achieved better performance than traditional statistical methods in learning without being expressly customized [8]. Data measured by the stations could be used for various purposes, such as: air quality management to reduce pollutant gases in the local atmosphere and climate monitoring for a better yields of the region crops [9]. To making alarming system this project has uses the Blynk apps to send a notification when the weather at bad condition to the user and also the buzzer sound at the weather station to put the user at home alert about the situation outside.

2. Methodology

This project will focus on development of the ThingSpeak an IoT platform that to show the data of the sensor. The method divided into two parts which are hardware and software development part. The hardware development involves the circuit construction and develops the prototype. Meanwhile, the software part involves the IoT coding, circuit schematic diagram, circuit simulation and data acquisition.

By using three (3) types of sensor to monitoring the weather parameter that are temperature, humidity, rain, and air quality the system will be able to display the weather condition by an analysis is about the current weather with the sensor value data. All the data will be control by a microcontroller ESP32 and Wemos as the client that will receive the sensor data from ESP32 and display it on OLED. Furthermore, this system will also be seen on ThingSpeak channel that has been created to simplify user to check online and also an android application that is Blynk to display the sensor data. The data collected will be analyse and compare it with Jabatan Meteorologi Malaysia to ensure the precise of data and weather condition on current condition. The Internet of Things (IoT) will connect the system with the user wireless and online without the need of checking manually.

2.1. Project Block Diagram

The overall project block diagram is illustrated in **Error! Reference source not found.**. The block diagram consists of the components that are utilized in this project. There are two modes available in this project working operation. Firstly, controlling mode will involve ESP32 and monitoring mode will involve Wemos D1 mini. This two-microcontroller board will communicate each other in order the monitoring mode get sensor data from controlling mode via wireless communication and hotspot Wi-Fi. Controlling mode will collect all the sensor data then send to the ThingSpeak website and monitoring mode to display on OLED. The client will display the sensor data on OLED and also on Blynk app. The data collected will be analyse to configure the actual condition and the current condition by using simple formula in Equation 1. The result of this data analysis then will be made the weather state for this system to tell the user about the rain and air quality condition is it good or bad in actual condition.

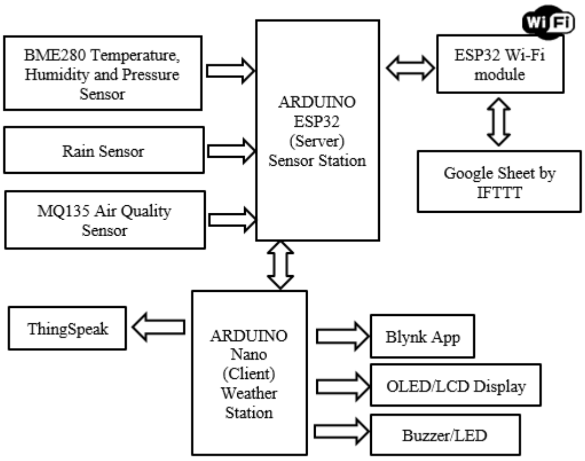


Figure 1. Project block diagram.

2.2. Control Unit (Sensor Unit)

Generally, this project is proposed to implement the IoT technology as a communication medium in this project. As stated in the previous section. The process of the system is start after the microcontroller ESP32 configured all the sensor and start to read the data from the sensor. Then, the data also be sends to the IoT platform that is ThingSpeak via wireless communication by ESP32 Wi-Fi network. The sensors that connect to ESP32 is act as the control unit of the system where all the data is collected here. This system automatically displays the temperature, humidity, pressure, rain, air quality, and weather condition on specific webpage of IoT in ThingSpeak as well it will also display this data on the weather station display. The flowchart of the con-trolling mode process is shown in Figure 1.

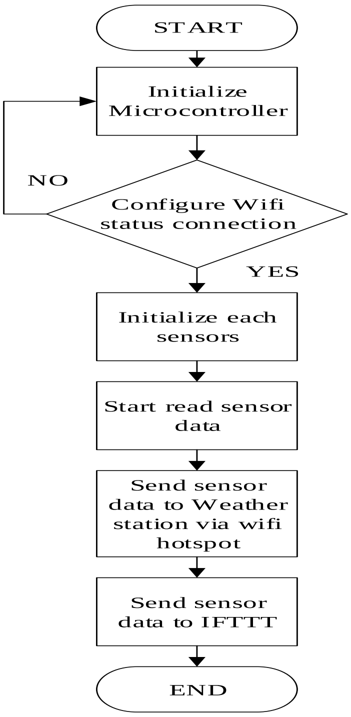

START
Initialize Microcontroller
NO
Configure Wifi status connection
YES
Initialize each sensors
Start read sensor data
Send sensor data to Weather station via wifi hotspot
Send sensor data to IFTTT
END

Figure 1. Control unit flowchart.

2.3. Monitoring Unit (Weather Station)

Meanwhile, wireless weather station using Wi-Fi system is employed monitoring unit. This will be controlled by Wemos D1 mini microcontroller to connect to the Wi-Fi and received live sensor data and display on OLED. This communication will simulate the master-slave network protocol or client-server protocol to develop a perfect data input/output from a control unit to a monitoring unit. If the sensor data come with faulty the buzzer/led will be on as the security of the system. Then a push notification will be sent to the user by Blynk app on android smartphone. All of these process flows are simplified in a form of flowchart in
Figure **2**.

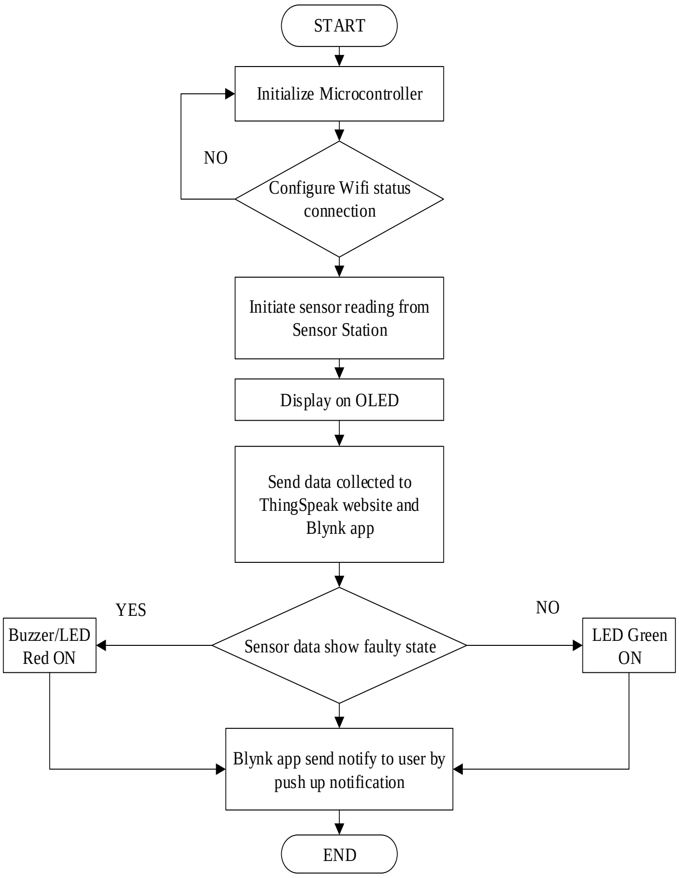

START
Initialize Microcontroller
NO
Configure Wifi status connection
Initiate sensor reading from Sensor Station
Display on OLED
Send data collected to ThingSpeak website and Blynk app
YES
NO
Buzzer/LED Red ON
Sensor data show faulty state
LED Green ON
Blynk app send notify to user by push up notification
END

Figure 2. Monitoring unit flowchart.

2.4. Hardware Development

The hardware selection is vital in this project hardware development process. Every hardware components are necessary to be considered first before selected to be utilized in the project. The components selection is according to the advantages and characteristic of the component to fulfil the functionality of every part used. For this project, ESP32, Wemos D1 mini, BME280, MQ135, Rain Sensor, OLED, LCD, Buzzer, and LED are used.

2.4.1. Circuit Construction. The circuit of the system can be divided to two where first circuit as weather station that displaying all the value of weather parameters. Other circuit is control unit circuit for controlling all the sensor data and send it to the ThingSpeak and websites. Weather station will be communicating with control unit via client-server communication where all the data catches by the sensor control unit will send it to weather station to display it. This station also equips by emergency alert where there is bad weather such as heavy rain and poor air quality. Figure 3 shows the electronic circuit of the sensor. The weather station completed circuit is shown in Figure 4. The bme280 sensor connect on I2C pin at ESP32 which is pin 21 and pin 22. There rain sensor and air quality sensor are connected on pin 33 and pin 32. In the monitoring mode the Wemos D1 mini is uses to connect I2C LCD and I2C OLED. The pin that use to connect LCD are D2 and D1 also OLED on D3 and D5.

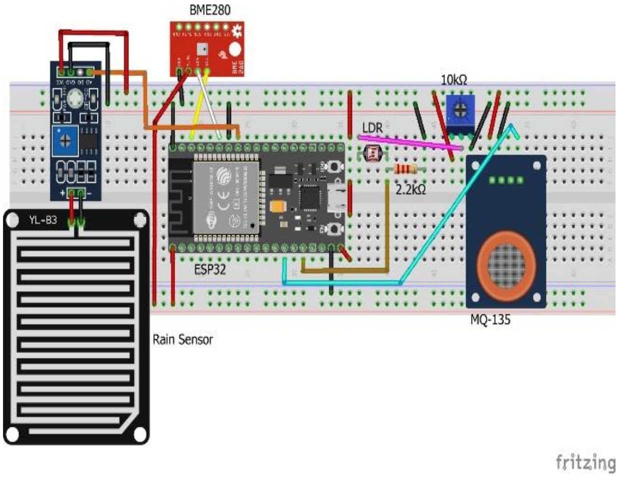


Figure 3. Sensor Unit.

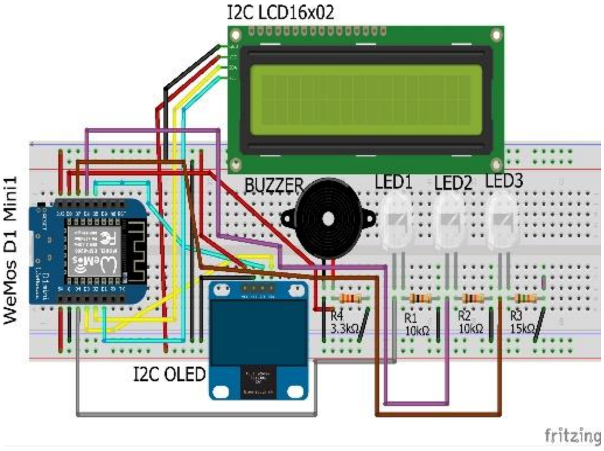
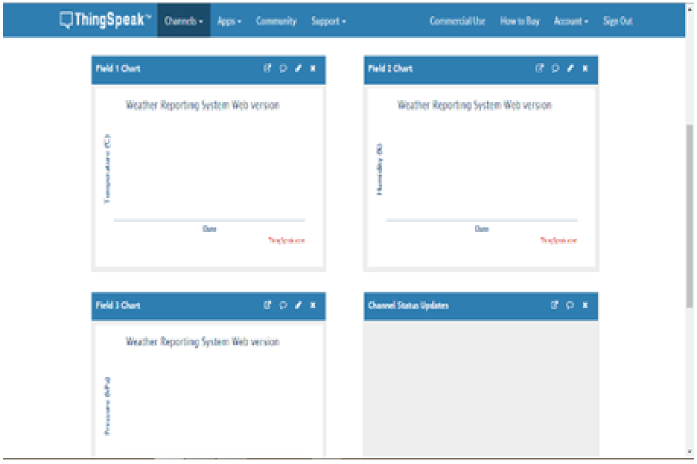


Figure 4. Weather station.

2.5. *Software Development*

Generally, most of project involves various type of software to be utilized in order to simulate and analyse the hardware configuration. This process can at least help the project member to troubleshoot and analyse the project configuration and result. Therefore, in this project few types of software will be used in order to develop the project. Most help is Arduino IDE with the software for compile and upload the coding, SolidWorks for 3D design, and Proteus for simulation are using in this project.

2.5.1. Internet of Things (IoT) Setup. The IoT platform that used in this project is ThingSpeak, new user needs to create ThingSpeak account at https://thingspeak.com. From the website create a new field as weather parameter output results as



2.5.2.

2.5.3. ***Figure 5***.

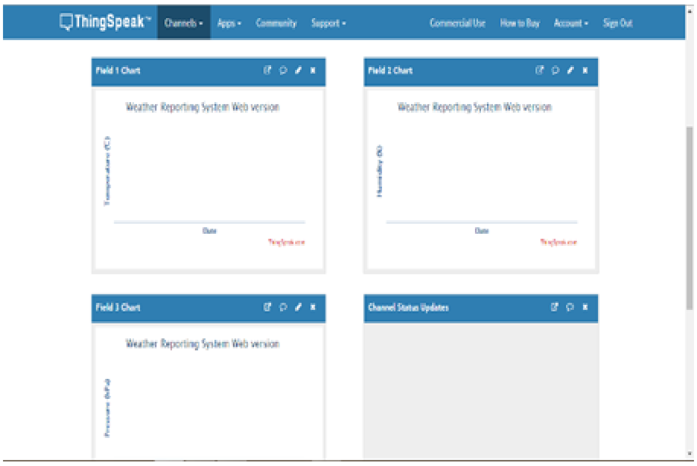
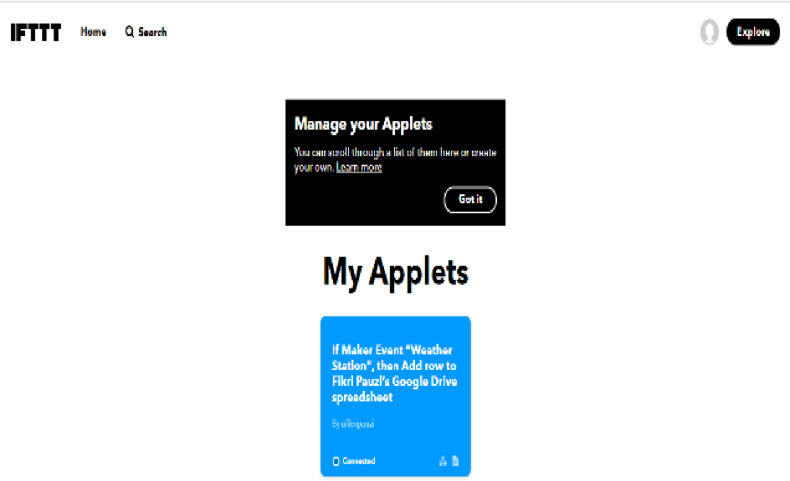


Figure 5. ThingSpeak create account and design.

2.5.4. Google Sheet with IFTTT platform. By using IFTTT as one of the IoT platform that can save data from any IoT devices and ensure the data saved to the google sheet. **Error! Reference source not found.** below shows the IFTTT platform that has been created to this project and Figure 6 show the data has successfully save in google sheet. With the assist of this platform the data can be recorded with ease and table can be perform.



Manage your Applets

You can scroll through a list of them here or create your own. Learn more

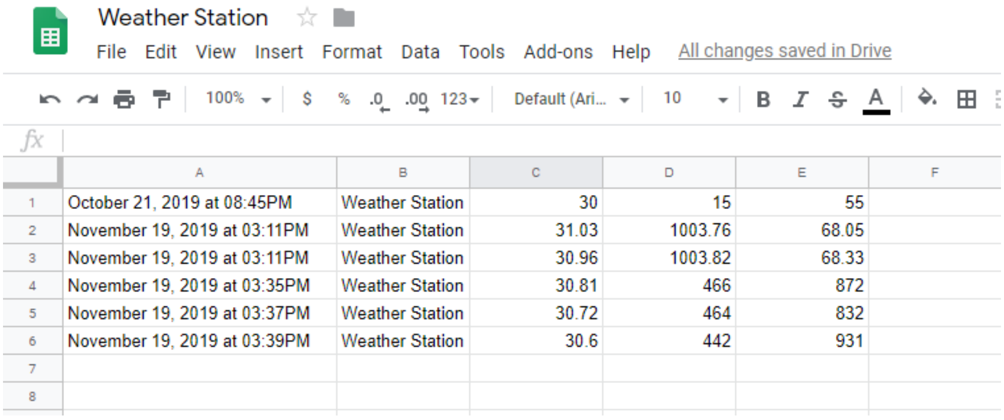
My Applets

If Maker Event "Weather Station", then Add row to Fikri Pauzi's Google Drive spreadsheet

By afikripauzi

Figure 7. My applets for project Weather System.

With the assist of this platform the data can be recorded with ease and Table 1 can be perform without to check manually to sensor station. Just go to the IFTTT website and sign up for free then create a new project that will link the sensor station with the server then save it in google drive by google sheet form.



	A	B	C	D	E	F
1	October 21, 2019 at 08:45PM	Weather Station	30	15	55	
2	November 19, 2019 at 03:11PM	Weather Station	31.03	1003.76	68.05	
3	November 19, 2019 at 03:11PM	Weather Station	30.96	1003.82	68.33	
4	November 19, 2019 at 03:35PM	Weather Station	30.81	466	872	
5	November 19, 2019 at 03:37PM	Weather Station	30.72	464	832	
6	November 19, 2019 at 03:39PM	Weather Station	30.6	442	931	
7						
8						

Figure 6. Data save in Google Sheet.

This data is received from sensor station, with ESP32 will send the weather parameters data to google sheet but unfortunately this method use by IFTTT can receives only three set of data so for this temperature, air quality, and rain data has been chooses.

IFTTT use to collect the data, so that the analysis can be done without any miss of data collecting. This system IoT platform converted the reading of three weather parameters to a table and IFTTT limited only three weather parameters can be saved.

2.6. Weather station alarm/security

The uses of Buzzer and LED are to alert the user about the hazardous weather condition outside by triggering the Buzzer and LED Red ON. The LED Green determine the current condition is good and LED Red determine the current condition is bad. The switch act will freeze the Buzzer alarm but the

LED still ON. Figure 9 shown the trigger weather parameters are temperature, rain, and air quality to avoid such hazardous condition to the user by alerting them.

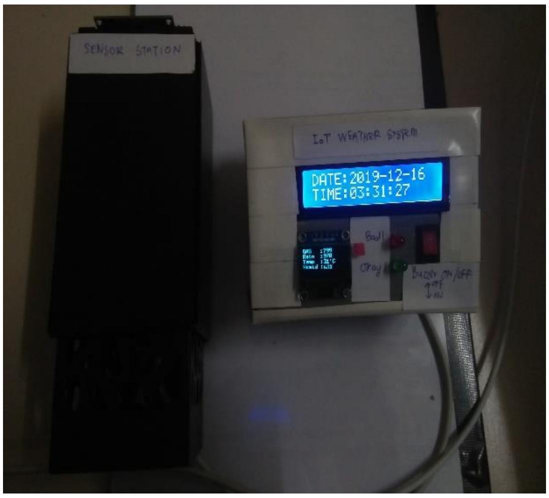


Figure 7. Sensor station and Weather station.

3. Results and Discussion

First the circuit of control unit system have been made that ESP32 microcontroller control all weather parameters sensor, that are BME280 (Temperature, Humidity, and Pressure) sensor, MQ135 (Air quality sensor), Rain sensor, and LDR sensor. Then it powered by USB cable also to upload the sketch of coding in ESP32 microcontroller. The sensor data can be display on serial monitor in Arduino IDE software. ESP32 will connect with the Wi-Fi hotspot that have applied to this system so that the web server can be create to display all the sensor data.

Data that received by weather station will be displaying on OLED as Figure 8 demonstrate the communication of both sensor station and weather station by using Wi-Fi hotspot. The communication is successfully established.

The web server contains html that can display the sensor data by simple coding and connection where the IP address of the ESP32 are needed to complete this action are shown in
Figure **9**. After that it will read all the sensor value and then send to the cloud data where ThingSpeak has been uses for this. ThingSpeak will stored the sensor value and display that data to the channel create there. The user can check the weather parameter via ThingSpeak websites. The data has collected from the reading of ESP32 for all sensor and send that data to the ThingSpeak as the results of this project objective. The analysis of the data has been made and there are table comparing the data with the help of the graph on this ThingSpeak channel.

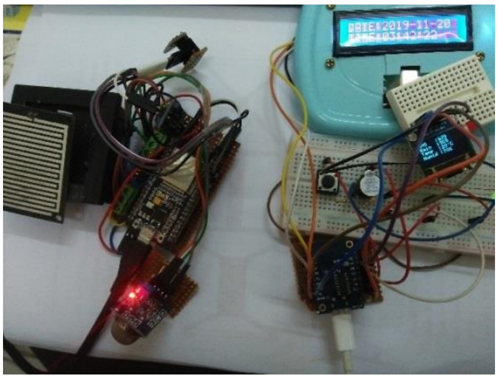
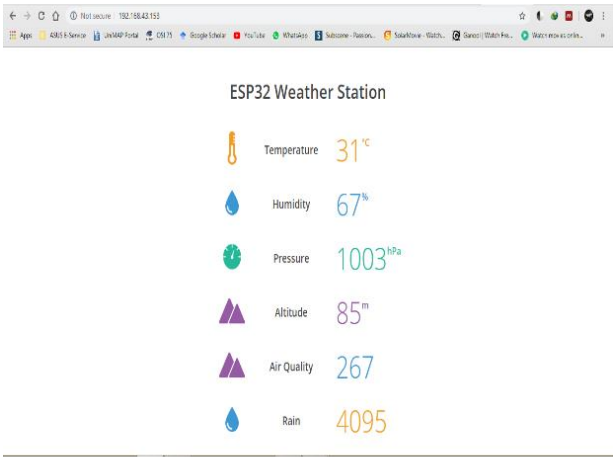


Figure 8. Full circuit for Sensor station and Weather station displaying sensor data.



ESP32 Weather Station

Temperature 31 °C

Humidity 67 %

Pressure 1003 hPa

Altitude 85 m

Air Quality 267

Rain 4095

Figure 9. Web Server construction.

As shown on the web server all the weather parameter clearly displays sensor data of the ESP32 sensor unit. After analyse the result from the web server and Thing speak graph there are these statistics the sensor data can been conclude as unpredictable. The data are taken from morning, noon, and night to seen the accuracy of the sensor reading. Based on ideally result of the practical results of the circuit operation are concluded in Table 1, 2, and 3. All data of bme280 sensor, air quality sensor, and rain sensor were recorded from google sheet form to Table 1, 2, and 3. The data temperature, air quality, and rain sensor are got from google sheet and the others data are got from ThingSpeak platform.

Table 1. BME280 Sensors data taken from different time.

Weather parameters	Time	Sensor value/Condition
Temperature	10am	24~28 (Normal)
Temperature	1pm	29~31 (Normal)
Temperature	10pm	27~29 (Normal)
Temperature	~	24~31
Humidity, (%)	10am	78

Humidity, (%)	1 pm	80
Humidity, (%)	10pm	77
Humidity, (%)	~	81
Pressure	10am	1001~1800
Pressure	1 pm	1005~1829
Pressure	10pm	997~1560

Based on the Table 1 above the calculation to determine the accuracy of the proposed sensor has been made. As the data get from met.malaysia.com for Jabatan Meteorologi Malaysia, stated the weather at my place in Table 4 then compared with data from Table 1, Table 2 and Table 3.

Table 2. Data read by MQ135 Air Quality Sensor.

Weather parameter	Time	Sensor value/Condition
MQ135 Air Quality	10am	148~210
MQ135 Air Quality	1 pm	166~230
MQ135 Air Quality	10pm	145~190
MQ135 Air Quality	~	< 200
Mq135 Air Quality	~	<600

From Table 2, the Mq135 sensor collect data for air quality which determine whether the air condition is safe without any haze or air pollution.

Table 3. Rain sensor to collect the actual data for comparison with real time data.

Weather parameter	Time	Sensor value/Condition
Rain	10am	1025 (No Rain)
Rain	1 pm	977 (No Rain)
Rain	10pm	841 (No Rain)
Rain	~	300<Rainy>600
Rain	~	< 300 (Heavy Rain)

Table 3 above shown the result of rain sensor value collecting data to make the comparison with the real time condition whether it rain or not. By the time the sensor value show large value there is no rain but when the sensor value show lower value there are maybe rainy or heavy rains.

Table 4. Data collect from met.malaysia.com for Perlis region.

Weather Parameters	value
Temperature, °C (Min)	25
Temperature, °C (Max)	32
Air Quality	~
Humidity, %	72
Rain, Morning	No Rain
Rain, Afternoon	No Rain
Rain, Night	Rainy

The data from Table 4 is get from Equation 1. To determine if a value is accurate compare it to the accepted value. As these values can be anything a concept called percent error has been developed;

$$\%\text{error} = \frac{|\text{Forecast}-\text{Actual}|}{\text{Actual}} \text{ x } 100\% \tag{1}$$

%Error = ±4.35 and also ±7.14 on data that has big difference.

To determine if a value is precise find the average of your data, then subtract each measurement from it. This gives you a table of deviations. Then average the deviations. This will give you a value called uncertainty. A plus or minus value that says how precise a measurement is;

$$\text{Deviation} = \text{Average} - \text{Actual} \tag{2}$$

$$\text{Standard deviation} = \frac{\text{All deviation added together}}{\text{Number of measurements}} \tag{3}$$

Based on the Equation 2 and Equation 3 the calculation has been;

Deviation = ±0.4314
Standard deviation = ±1.64

By calculations done from Equation 1, Equation 2 and Equation 3 above the accuracy of the temperature taken from sensor and forecast are not 100% accurate with less than ±4.35 and ±7.14 percentages error, % Error.

3.1. ThingSpeak display and data analysis

After testing with the web server, the ThingSpeak channel also received the same sensor data as web server displaying but with better display of graph plot. This graph makes the analysis of the sensor data be easier. All weather parameter come with analogue data with ThingSpeak this data will be plot on the graph and this will show the characteristic of the parameter.

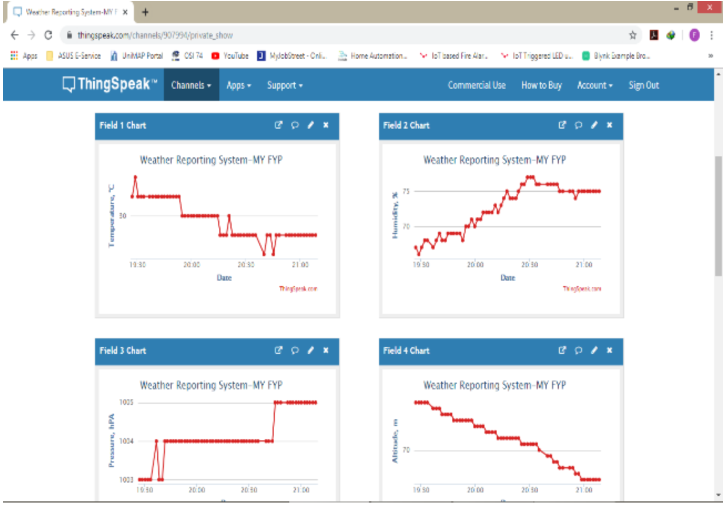


Figure 10 and Figure 11 show the graph of collected result from sensor.

The graph will keep getting the sensor data every one minute for the update sensor data. All the sensor data have been displaying in graph and there are six graph that are visible to the people there are, temperature, humidity, pressure, altitude, rain, and air quality. All data from the sensor can be converted to a table of table by ThingSpeak website.

With the ThingSpeak as the IoT platform for everyone that need short or easy way to monitor the online system. The website ease user and people to see the condition at certain place with online system by computer or smartphone with internet connection.

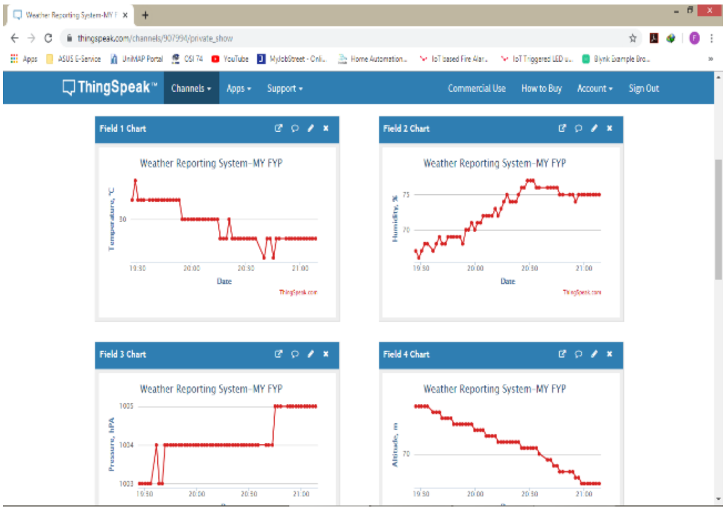


Figure 10. Sensor data plot on graph in ThingSpeak.

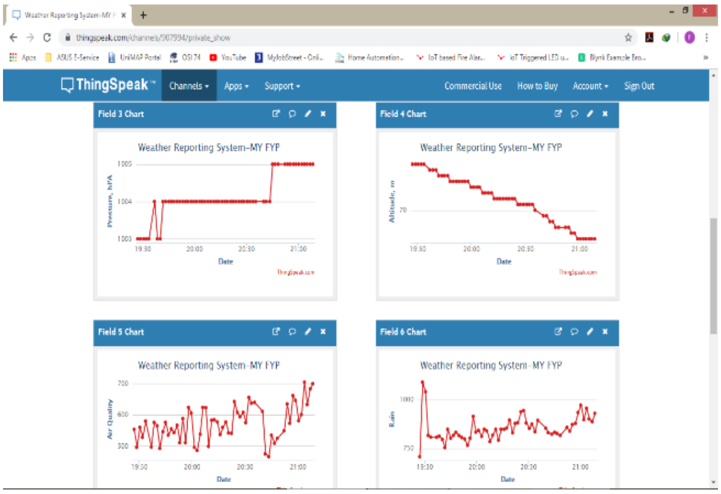


Figure 11. Sensor data plot on graph.

With these graphs, Table 1, 2 and 3 able to be constructed with the help of web server test also. There are BME280 (Temperature, Humidity, and Pressure) sensor, MQ135 Air Quality Sensor, and Rain Sensor. All the collected sensors data have been to make this analysis from the sensor data.

Based on the graph also the comparison between actual condition to forecast has been made and the results of this data are recorded in Table 5 with tell the exact condition with the sensor data comparison.

In Malaysia highest temperature record near 40°C and it so dangerous that people can get heat stroke and died. This analysis can make people to prepare or planning for their daily activity. From the analysis of the sensor data of collected chooses sensor and tested for few days this have been conclude that.

Table 5. Temperature data comparison with forecast data.

Sensor	Data	Forecast	Data
Temp, °C	24	Temp, °C	24
Temp, °C	23	Temp, °C	24
Temp, °C	31	Temp, °C	30
Temp, °C	30	Temp, °C	31
Temp, °C	32	Temp, °C	31
Temp, °C	28	Temp, °C	30
Temp, °C	29	Temp, °C	28

As the sensor for MQ135 and Rain sensor display on analog value, so analysis is needed to assume what value did the sensor read at actual condition. Table 6 recorded the analog value for both sensor and the actual condition for each reading take.

Table 6. Analysis for MQ135 sensor and Rain sensor.

Sensor	Sensor data/value	Actual condition
MQ135	405	Not good
MQ135	587	Not good
MQ135	322	Fair
MQ135	255	Fair
MQ135	122	Good
MQ135	100	Good
MQ135	421	Not good
MQ135	366	Fair
MQ135	303	Fair
MQ135	289	Fair
Rain Sensor	3888	No rain
Rain Sensor	3752	No rain
Rain Sensor	3774	No rain
Rain Sensor	3445	No rain
Rain Sensor	2551	No rain
Rain Sensor	1552	Rainy
Rain Sensor	1007	Rainy
Rain Sensor	1248	Rainy
Rain Sensor	831	Heavy rain
Rain Sensor	2897	No rain

As for conclusion the MQ135 sensor and Rain sensor is divided to three part for the results. The results will show the characteristic of sensor data for actual condition state which is to configure the sensor data value.

For MQ135 sensor;

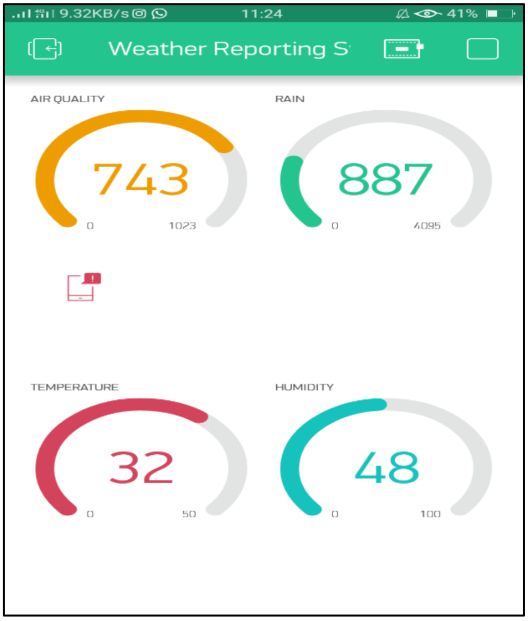
- For analog value more than > 400 it states that the air quality is not good.
- For analog value in between 250<a<400 the air quality is fair.
- And for analog value below or less than < 250 the air quality is good.

For Raindrop sensor;

- When the analog value goes up to >800 the actual condition is No Rain.
- If the analog value down to <800 the actual condition is Rainy.
- And if analog value below <300 it was Heavy Rain in actual condition.

3.2. Blynk display sensor data

Another IoT tools that can display sensor data via wireless communication is Blynk app. This application can do anything that relate to IoT development. In order to make this weather reporting



system can display sensor data and visible anywhere around the world this app has been built. The sensor data completely shown on Blynk app see **Figure 12**. As this android app is free to build by download it on Google Playstore and register with email to use the app. By some setting with the app to configure the hotspot connection of internet then specify each weather parameter into the app widget setting.

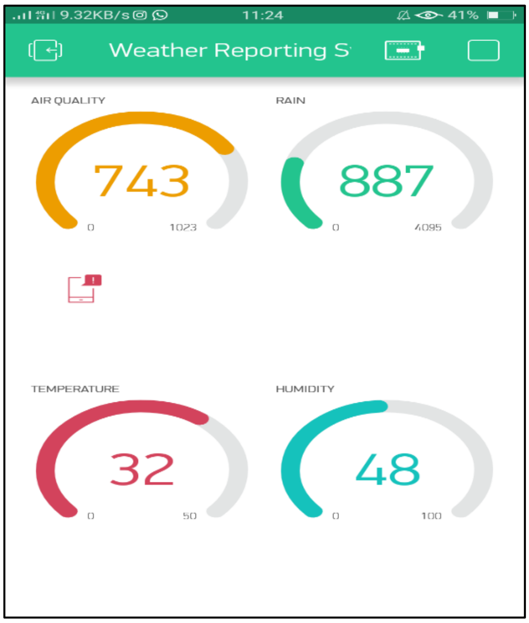


Figure 12. Blynk display data.

3.3. OLED display sensor data

The monitoring mode Wemos D1 mini is act as a client to receive the sensor data from ESP32 as the server. The data completed shown on OLED as shown on Figure 13 via wireless communication. The setting up of the board was to determine the I2C pin and power supply for both LCD and OLED where

OLED use 3.3V and LCD use 5V. The weather station will receive a packet of data from sensor station to display on OLED, the communication use between both boards is Wi-Fi hotspot communication. The LCD will display real time and data that use NTP client to update the time and date by internet communication. As the main point in this project is Internet of Things all the outcome comes in IoT result.

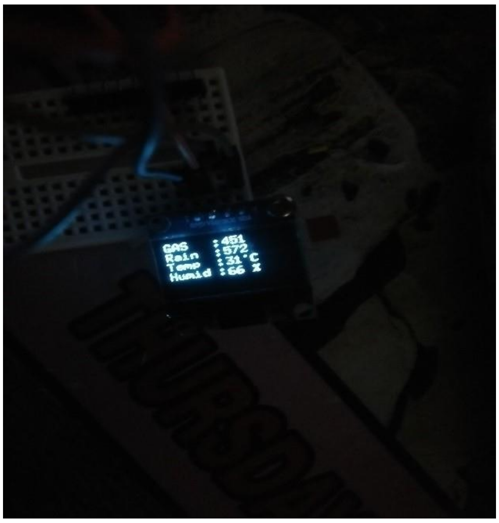


Figure 13. OLED display sensor data send by ESP32 via hotspot.

3.4. Discussion

Based on the results that are obtained by the sensors send and display to ThingSpeak for user viewing. This will make monitoring weather parameter more easily with the Wi-Fi connection this system will start and ThingSpeak start displaying sensor data by graph and also all this data can be analysing in ThingSpeak. From the data Table 1, 2 and 3 and Table **4**, the results have been compared and shown that the weather parameter from forecast station is not too accurate so with weather reporting system people can get the actual condition at their place with ease.

The results of MQ135 and rain sensor from Table 6 uses to make comparison in actual condition of particular location in determine the sensor data value corresponding with actual condition and conclusion of the results data recorded can be made.

With wireless monitoring network devices, the people can check online on the web page the weather condition to take certain steps and issues even in worst case for monitoring the weather parameters. With all the data also weather reporting system is to monitoring certainly weather to overcome the most important factor determining agricultural enterprise success or failure and have ease the user from getting inaccurate forecast from Forecast Company for their place. The IoT component that has established the communication between the microcontroller boards were ESP32 and Wemos D1 mini with communication via Wi-Fi hotspot.

4. Conclusion

As the conclusion this project have cleared the objective that to build a system that can monitored weather parameter by wireless system and IoT. The Sensor station and Weather station will be communicated by hotspot Wi-Fi and it is limited in areas covered but still better in communication via wireless. The value that been recorded from google sheet and Table 1, 2 and 3 it seen that the weather at particular place has different condition from the exact condition with the accuracy of weather reporting system and forecast system data has been compared. It says that weather reporting system is more accurate than forecast system. This weather reporting system will display the sensor data to

ThingSpeak and IFTTT to save the data into google sheet. It also can be checked in Blynk app that can be installed in google play store or Appstore.

Acknowledgements
All of the success in this thesis depends largely on the encouragement and guidelines of many others. I extended my sincere and heartily thankful to my Supervisor Mr. Mohd Zamri Bin Hasan, Faculty of Electrical Technology Engineering, University Malaysia Perlis (UniMAP), Perlis for encouragement, guidance and support to completing this project successfully.

References

[1] Kulkarni, V. A, Satpute G. M (2017). “Weather Reporting System Using FPGA : A Review,” vol. 4, no. 11, pp. 319–320.
[2] Carlos, M, Jorge, P.B, Daniel F, Pablo S (2018). “Design, Development and Implementation of a Weather Station Prototype for Renewable Energy System,” Journal Energies, 11(9), 2234, pp. 1-13.
[3] Karim F, Karim F and Frihida A (2017). “Monitoring system using web of things in precision agriculture,” *Procedia Computer Science.*, vol. 110, pp. 402–409.
[4] Kodali R K, Yerroju S and Sahu S (2018). “Smart Farm Monitoring Using LoRa Enabled IoT,” *Proceedings 2nd International Conference Green Computing Internet Things, ICGCIoT 2018*, pp. 391–394.
[5] Gahlot N, Gundkal V, Kothimbire S and Thite A (2015). “Zigbee based weather monitoring system,” *Internnational Journal Engineering Science*, vol. 4, no. 4, pp. 2319–1813.
[6] Kodali R K and Sahu A (2016). “An IoT based weather information prototype using WeMos,” *Proceedings 2016 2nd International Conference Contemporary Computing Informatics, IC3I 2016*, no. 1, pp. 612–616.
[7] Joe F, and Joseph J (2019). “IoT Based Weather Monitoring System for Effective Analytics,” International Journal of Engineering and Advanced Technology (IJEAT), no. 4, pp. 311–315.
[8] Nallakaruppan, M.K, and Kumaran U.S (2019). “IoT based Machine Learning Techniques for Climate Predictive Analysis,” International Journal of Recent Technology and Engineering (IJRTE), no. 5, pp. 171–175.
[9] Monteiro M S, De Caldas Filho F L, Barbosa L A, Martins L M C E, De Menezes J T M and Da Silva Filho D A(2019). “University campus microclimate monitoring using IoT,” *WCNPS 2019 - Workshop Communication Networks Power Systems*, no. Wcnps, pp. 3–7.