Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF IOT-BASED WEATHER MONITORING AND REPORTING SYSTEM USING NODEMCU

AINUL BALQIS BINTI HAIRUDDIN

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

2022

DEVELOPMENT OF IOT-BASED WEATHER MONITORING AND REPORTING SYSTEM USING NODEMCU

AINUL BALQIS BINTI HAIRUDDIN

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

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2022

DECLARATION

I declare that this project report entitled "Project Title" 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 : AINUL BALQIS BINTI HAIRUDDIN

Date : 10th 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 Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature :

Supervisor Name : TS. MOHD RAZALI BIN MOHAMAD SAPIEE

Date : 26/1/2023

Signature :

Co-Supervisor Name (if any) :

Date :

DEDICATION

I want to express my heartfelt thanks to my parents, Hairuddin Bin Mustafa and Rodziah Binti Mat Nayan, for their constant support and encouragement until I finished my final project. My father gives me a lot of support as I try to make my project a reality. He gave an idea of how to do it on time and stress-free. They also set up a welcoming space where I could get inspiration and ideas for completing my projects. Aside from that, I also want to mention my colleague Amni Najihah Binti Abd Aziz, who gave me a lot of advice on how to do my tasks effectively. In addition, I want to thank my supervisor, Ts. Mohd Razali Bin Mohamad Sapiee, for all the advice and support he gave me whenever I had questions concerning the project, regardless of the time.

ABSTRACT

This project aims to make it simple for individuals to obtain current weather data from any location. Weather predictions and real-time weather stations are different. The true purpose of current weather is to predict the weather for a particular location at a specific moment. The true purpose of a weather forecast is to predict the weather for a specific area at a particular time. On the other hand, a current-time weather station is a system that collects meteorological and environmental data using many sensors. On the other hand, a weather station is a device that monitors atmospheric characteristics such as temperature, humidity, and rain to offer data for weather study and discovery. Some believe that without weather stations, people won't be able to foresee and be notified about severe calamities like high winds, heat waves, heavy rain, tornadoes, lightning, and other weather-related occurrences that often occur in our country. As a result, these weather stations are necessary to gather current weather data and update predictions. These approaches might be employed to address the project's issues. This project, which will use sensors to construct weather stations, will use the Internet of Things. These weather stations may also supply data for forecasting purposes. Once this weather station is ready to be linked, users may monitor and review the history of information on previous and present weather.

ABSTRAK

Projek ini bertujuan untuk memudahkan individu mendapatkan data cuaca semasa dari mana-mana lokasi. Ramalan cuaca dan stesen cuaca masa nyata bukanlah perkara yang sama. Tujuan sebenar cuaca semasa adalah untuk meramal cuaca untuk lokasi tertentu pada masa tertentu. Tujuan sebenar ramalan cuaca adalah untuk meramal cuaca untuk kawasan tertentu pada masa tertentu. Stesen cuaca masa semasa, sebaliknya, ialah sistem yang mengumpul data meteorologi dan persekitaran menggunakan banyak penderia. Stesen cuaca, sebaliknya, ialah peranti yang memantau ciri-ciri atmosfera seperti suhu, kelembapan dan hujan untuk menawarkan data untuk kajian dan penemuan cuaca. Ada yang percaya bahawa tanpa stesen cuaca, orang ramai tidak akan dapat menjangka dan dimaklumkan tentang bencana yang teruk seperti angin kencang, gelombang panas, hujan lebat, puting beliung, kilat dan kejadian berkaitan cuaca lain yang sering berlaku di negara kita. Akibatnya, stesen cuaca ini diperlukan untuk mengumpulkan data cuaca semasa dan mengemas kini ramalan. Ini adalah beberapa pendekatan yang mungkin digunakan untuk menangani isu projek. Projek ini, yang akan menggunakan penderia untuk membina stesen cuaca, akan menggunakan *Internet of Things*. Stesen cuaca ini juga mungkin membekalkan data untuk tujuan ramalan. Setelah stesen cuaca ini sedia untuk dipautkan, pengguna boleh memantau dan menyemak sejarah maklumat tentang cuaca sebelumnya dan semasa.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts. Mohd Razali Bin Mohamad Sapiee for their precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and my parent for the financial support which enabled me to accomplish the project—not forgetting my colleague, Amni Najihah Binti Abd Aziz for the willingness to share his thoughts and ideas regarding the project.

My highest appreciation goes to my parents and family members for their love and prayer during my study. An honourable mention also goes to my father for all his motivation and understanding.

Finally, I would like to thank all the staff at the lectures, colleagues and classmates, the Faculty members, and other individuals not listed here for being cooperative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Using a variety of sensors, a weather station collects data and makes it easy for the user to obtain information about the weather and the environment. The weather stations are also a facility that can be used to measure atmospheric conditions to provide data for weather forecasts, study weather and climate, and control environmental conditions dependent on forecast data. A weather station measures temperature, air pressure, humidity, wind speed, direction, and precipitation levels, among other things. Weather stations are also known as weather centres, personal weather stations, professional weather stations, home weather stations, and weather predictions [1].

1.2 Problem Statement

Weather conditions influence human activity, and weather monitoring may aid activity regulation. It's critical to monitor and assess the region's weather patterns. Users only have a few options for learning about the weather, such as temperature, humidity, and wind speed, all of which are essential factors to consider [1]. Without a weather station, the user will not be notified of the weather. Strong winds, heat waves, or any other type of weather disaster are all possibilities.

Furthermore, forecasting the weather requires data. When a user uses a weather station, they can see the data's history. The user may identify the trends in the measurements. As a result, the user can examine patterns more closely.

1.3 Project Objective

This project's primary purpose is to develop a systematic and practical mechanism for accurately monitoring and reporting weather across the whole distribution network system. The following are the specific objectives:

a) To create a real-time weather station that allows users to obtain data in real-time from anywhere.

b) To implement the Internet of Things on the weather station.

c) To evaluate the weather station's capabilities and efficacy, as well as to create data for the user.

1.4 Scope of Project

The scope is essential since it limits the extent of the project. As a consequence, the weather station collects weather data such as temperature, humidity, air pressure, rain drop and light via the Internet of Things.

a) It can view temperature, humidity, and air pressure, rain drop and light data in the user scope.

b) It will be based on the system scope. Temperature, humidity, and air pressure data are collected from the weather station.

c) The data was saved in real time and transferred to a local server.

d) Every 5 minutes, the data from the weather station will be transferred to the database.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

New technical applications have recently enabled us to analyze various air characteristics from a distance, allowing us to monitor air quality remotely [2]. As a result of the internet of things and the development of new gadgets, some applications have been developed. Monitoring systems need an application interface, which might be a web page, software, or a mobile app, to identify what data was received or operate the system.

For weather forecasting and monitoring, many weather stations still utilize analogue technology. They utilize several types of equipment to monitor changes in the weather, including thermometers for temperature, barometers for air pressure, wind vans, rain gauges for precipitation, and so on. The bulk of these devices is based on basic analogue technology. Their results are then carefully documented and stored. The information is then supplied to news stations, TV networks, and radio stations so they may report on the weather.

2.2 Previous Related Projects

Similar projects in the past focused on using Internet of Things technology, which helped me understand this latest project better. This information will help with carrying out the project and wrapping it up. Because of this, the next section will give some background on a similar project that tried to solve some of the problems with weather stations.

2.2.1 Internet of Things (IoT) Based Weather Monitoring System.

This project is proposed by Bulipe Srinivas Rao, Prof. Dr. K. Srinivasa Rao, and Mr. N. Ome [3]. A suitable implementation model has been identified; it comprises several sensor devices and other modules with their respective functionalities. For this particular implementation approach, we utilized as an embedded device, an Arduino UNO board with a Wi-Fi module is used to detect and save data in the cloud. An analogue-to-digital converter (ADC), a digital-to-analog converter (DAC), digital output pins (D0-D13), and a Wi-Fi module that links the embedded device to the internet are all included on the Arduino UNO board. Sensors are connected to an Arduino UNO board for monitoring purposes. The analogue-to-digital converter (ADC) will transform the sensor's analog reading into a digital value. The appropriate environmental parameters will be monitored based on that value.

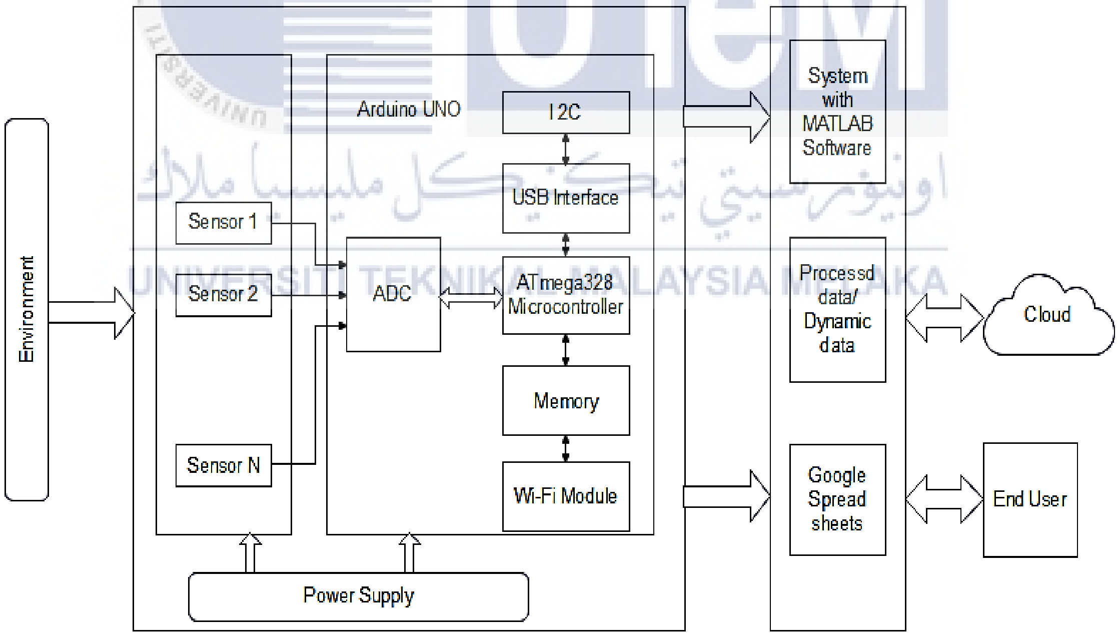


Figure 2.1 Schematic diagram of implementation model.

Figure 2.1 it depicts an embedded system designed with the aim of environmental monitoring, as well as the components that make it up. The embedded device is placed in a specific location for testing purposes. The sound sensor and the carbon monoxide (CO) sensor MQ-9 will record the air quality at that spot. If the threshold limit is exceeded, the proper steps will be taken. The sound sensor will record the air quality, while the CO sensor will measure the sound levels in the region. Wi-Fi modules are used to link each sensor device to the internet.

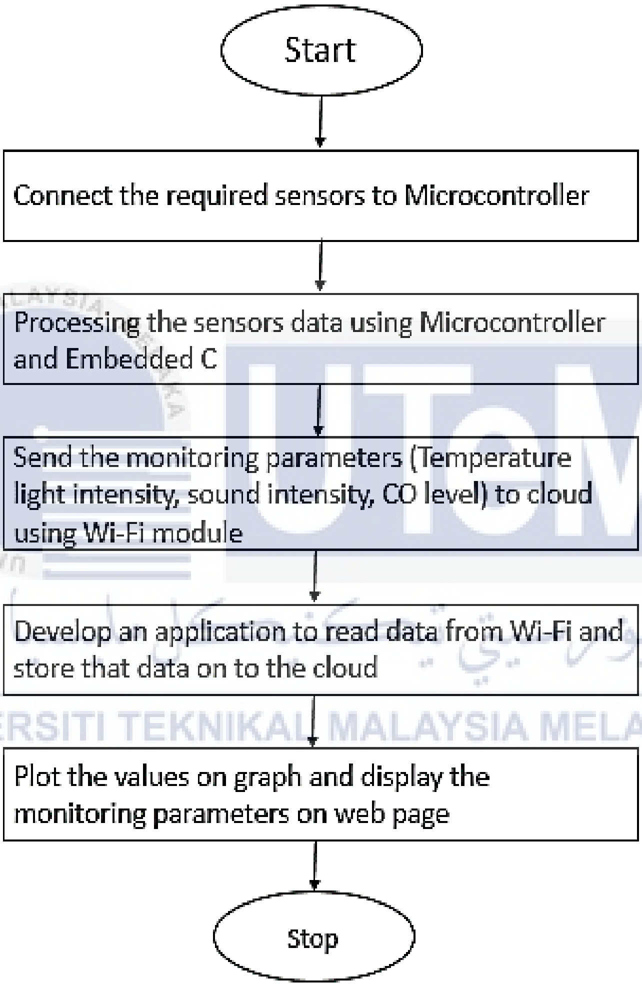


Figure 2.2 The flowchart.

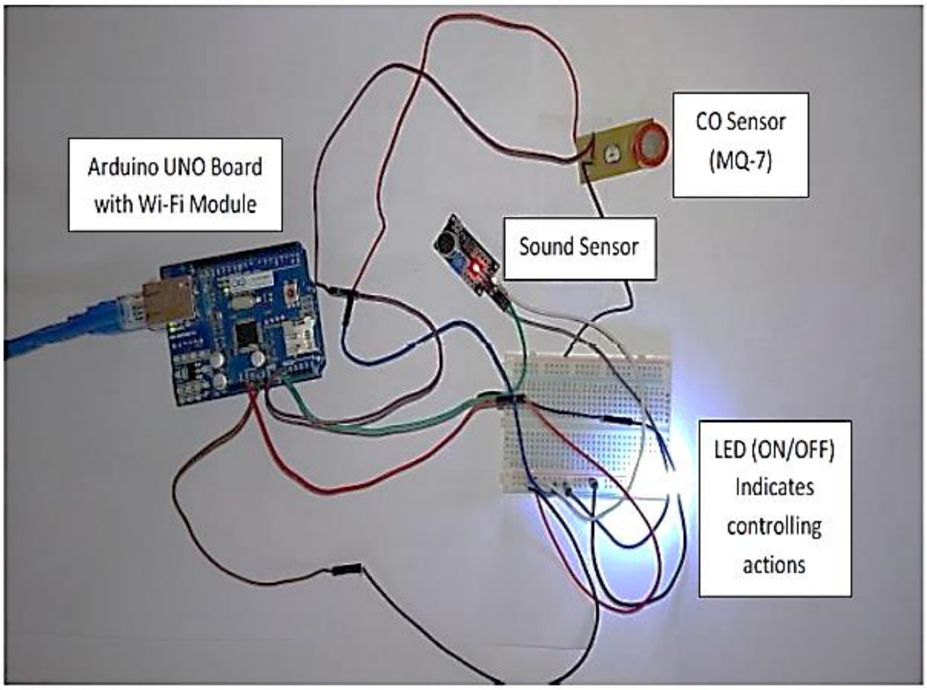


Figure 2.3 The noise and air pollution monitoring embedded system with its components.

See Figures 2.2 and 2.3 to see how the embedded system and its parts get environmental parameters from the cloud and store them. After the sensing is done without a hitch, the information will be analyzed and put in a database for future use. After the data analysis, the control threshold values will be set.

We will use the web server page to monitor and control the system. We will be able to access the related web page once we have entered the server's IP address that has been set up for monitoring. The website gives information on CO level and sound intensity changes in the area where the embedded monitoring equipment is placed.

2.2.2 Arduino-Based Weather Monitoring System.

This project was proposed by Ejodamen Pius Uagbae, Ekong, Victor Eshiet, Inyang, Udoinyang Godwin [4]. The functionality of the system includes the function of the whole system after the connectivity of all of its components, including software and peripheral devices. The operation of the system is broken down into three stages: the first stage involves reading the data from the sensors; the second stage involves reading the data from the EEPROM; and the third stage involves sending the data to the server (web page).

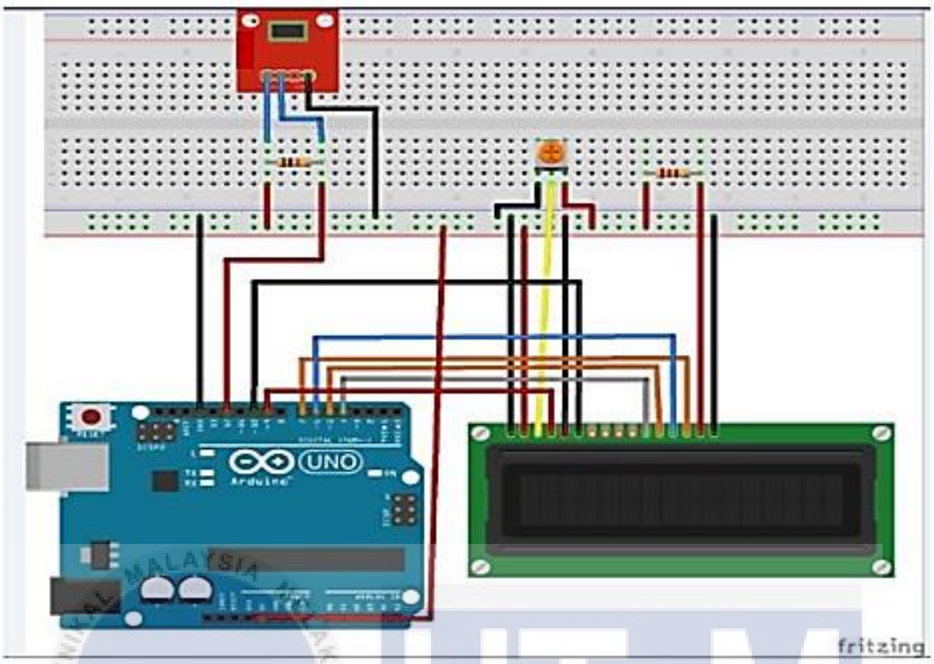


Figure 2.4 The connection for the Arduino-Based Weather Monitoring System.

The Arduino Uno Microcontroller Board was used as the primary piece of hardware. Instruction codes were written using the Arduino IDE and then uploaded to the microcontroller. Figure 2.4 shows how the chosen parts can be combined to make the circuit diagram work. The addition results in the connections that exist between the various components.

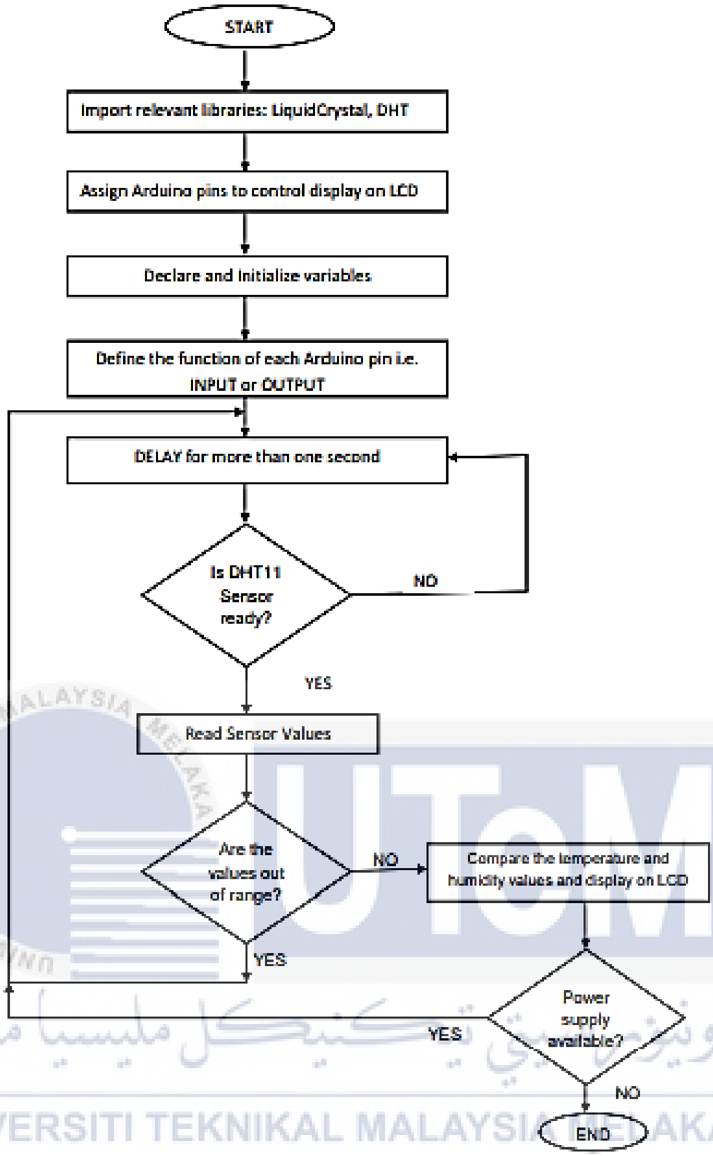


Figure 2.5 The connection for the Arduino-Based Weather Monitoring System.

The flowchart shown in Figure 2.5 is used during the design process for the instructions that drive the microcontroller. The Arduino IDE was used in the development of the firmware. This provided the tools to debug and upload the program to the microcontroller.

2.2.3 Weather Station Design Using IoT Platform Based On Arduino Mega

This project was proposed by Medilla Kusriyanto and Agusti Anggara Putra[5]. This system includes an Arduino Mega 2560 microcontroller, a DHT-22 temperature sensor, a rain detection sensor, a BMP-180 air pressure sensor, a DS3231 RTC, a 3.5" touch screen LCD, an ESP8266 as the information sending medium to the IoT platform, and an SD data storage card as the data processing center.

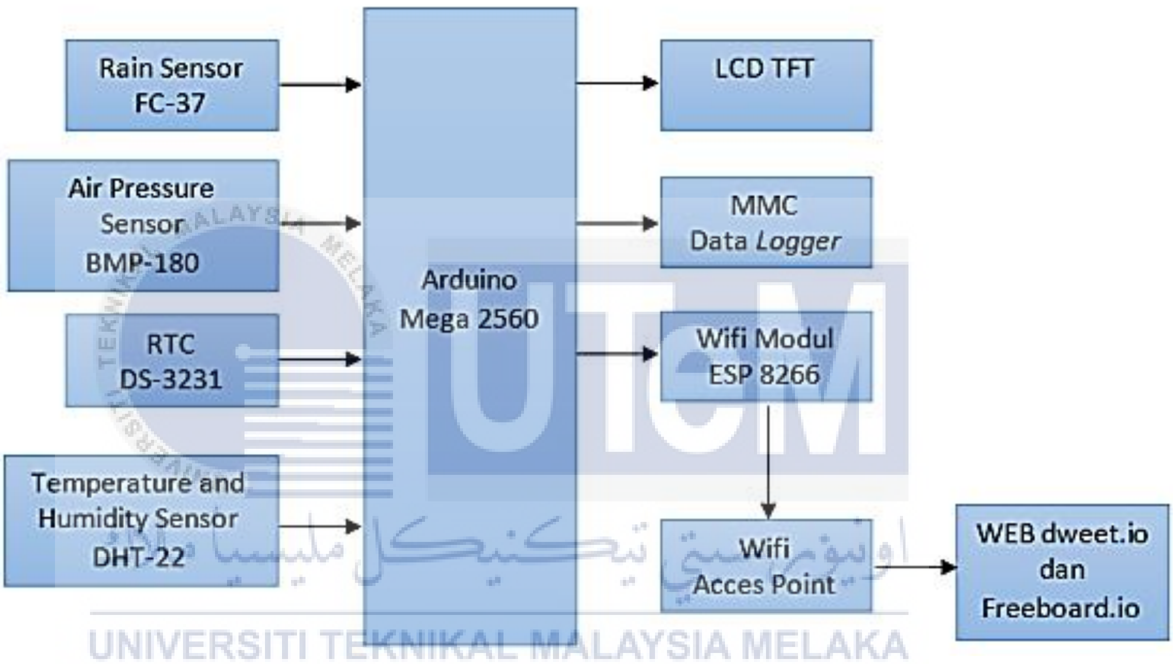
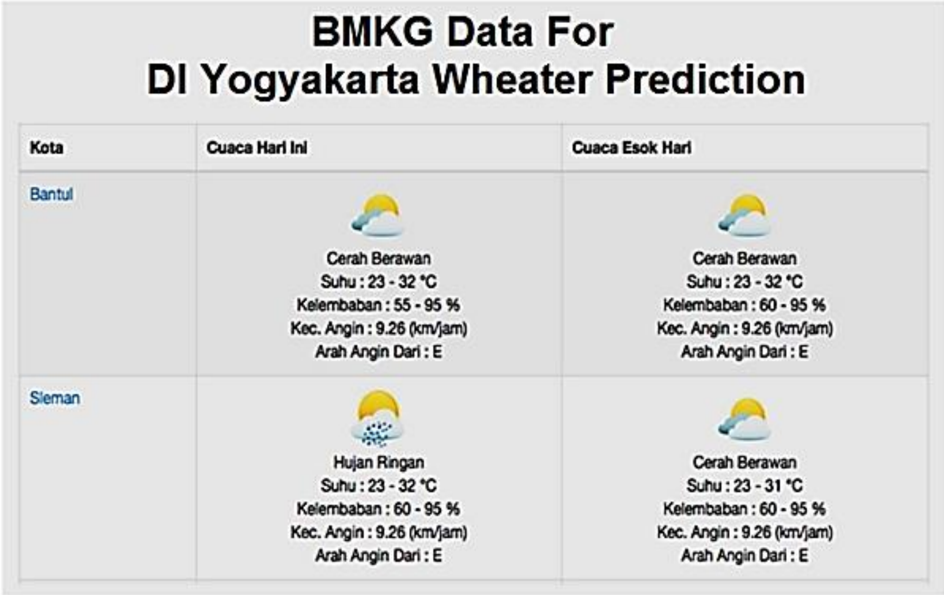


Figure 2.6 Weather System Block Diagram.

Based on Figure 2.6 on the block diagram, all sensors, namely the rain sensor, water pressure sensor, RTC DS-3231, temperature sensor, and humidity sensor, will be connected to the Arduino Mega 2560 on the input pin. All these sensors will record data over time. A TFT LCD, an MMC data logger, and a Wi-Fi module are located at the output pin. Three modules will also display and store the data collected from the sensors. When Wi-Fi is available, it will send data to the websites dweet.io and Freeboard.io.

Figure 2.7 The dashboard IOT platform.

On the data logger, testing is carried out by validating and determining the present state of the data transmission process to the website dweet.io. Based on the findings of the observations performed on the data logger's condition, when the Wifi ESP8266 module is correctly attached to the microcontroller and linked to the internet network, data transmission is possible. Figure 2.7 shows the results of the test.


BMKG Data For
DI Yogyakarta Wheater Prediction

Kota	Cuaca Hari Ini	Cuaca Esok Hari
Bantul	Cerah Berawan Suhu : 23 - 32 °C Kelembaban : 55 - 95 % Kec. Angin : 9.26 (km/jam) Arah Angin Dari : E	Cerah Berawan Suhu : 23 - 32 °C Kelembaban : 60 - 95 % Kec. Angin : 9.26 (km/jam) Arah Angin Dari : E
Sleman	Hujan Ringan Suhu : 23 - 32 °C Kelembaban : 60 - 95 % Kec. Angin : 9.26 (km/jam) Arah Angin Dari : E	Cerah Berawan Suhu : 23 - 31 °C Kelembaban : 60 - 95 % Kec. Angin : 9.26 (km/jam) Arah Angin Dari : E

Figure 2.8 The Wheather Status from BMKG data for Yogyakarta Wheater Prediction.

The test is carried out by comparing weather prediction algorithm data with data from the BMKG website, then compared to actual weather conditions. The air pressure drops, indicating that the weather is about to change. The findings of the weather forecast are stable rainy, and stable, according to the data. The term "stable rainy" means that there will be light rain in the area. The results are legitimate or by the weather forecast data acquired when the data is compared to the BMKG weather prediction results. Figure 2.7 shows the test result. Figure 2.8 shows the BMKG data for Yogyakarta wheat prediction.

2.2.4 IOT Based Wheather Reporting System.

This project was proposed by Vera SVET, Vasile Gurduza, and Eduard Medinschi [6]. Due to the architecture approach, the acquisition server implementation constitutes a unique system, particularly sensor connectivity. The sensor services separate the sensor components. The data flow from the sensors is routed via the communication module on the acquisition server, and data requests from the sensors are replaced with network requests.

Through the Wi-Fi network, the acquisition server may communicate with the regular Internet network.

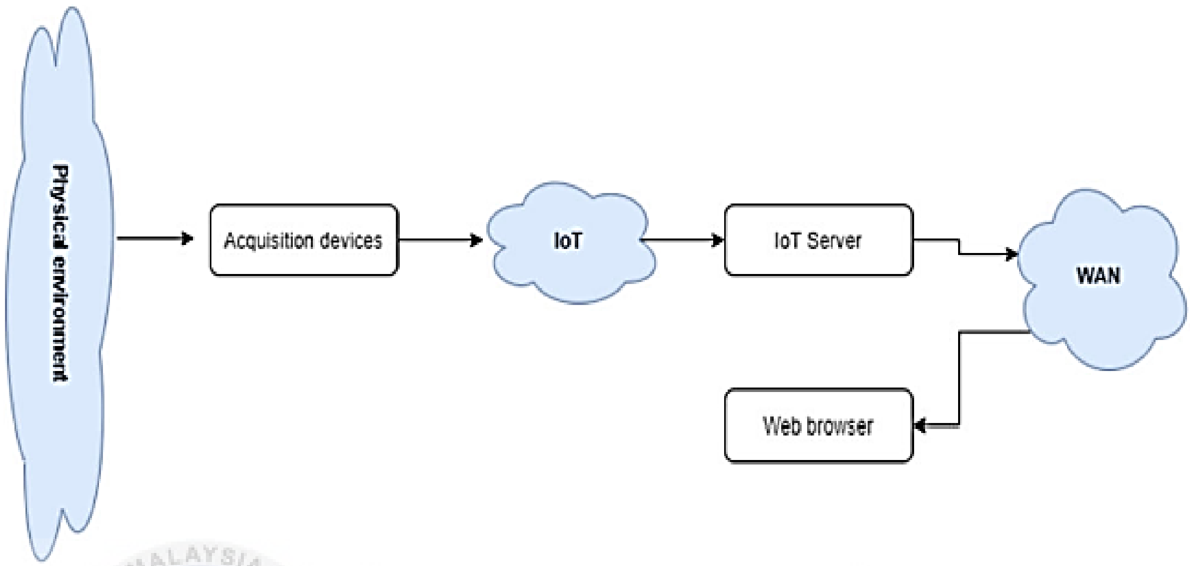


Figure 2.9 The general architecture of the system.

The goal of the system is to gather environmental data from weather stations all around the nation. The suggested system is a simplified weather monitoring system that uses the Internet of Things to make real-time data publicly accessible across various devices. Weather and climate change are monitored, and agricultural threats are monitored and forecasted using this equipment. The information will be saved on a system-specific server, with the option of accessing it through the Internet via a dedicated server-hosted web page. The system architecture is shown in the diagram.

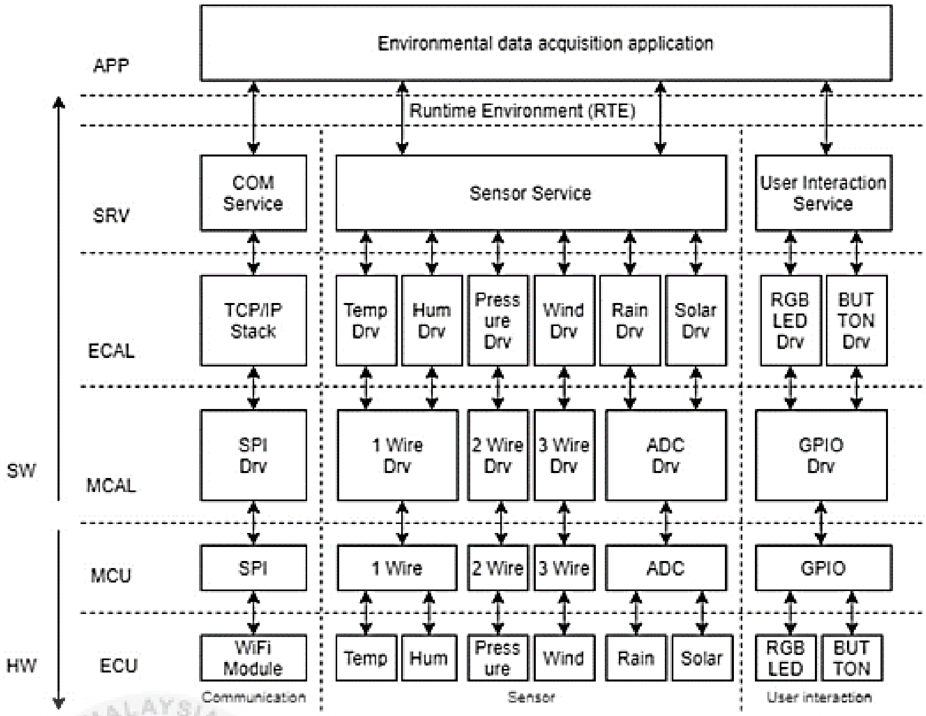


Figure 2.10 The layered architecture of hte IOT device.

Devices capture data about the environment and send it to the Internet of Things network. Temperature, humidity, air pressure, wind speed and direction (anemometer), raindrop, and light sensors will all be included in the system.

The acquisition server implementation is a one-of-a-kind system, especially in connecting sensors, because of how it was designed. Sensor services keep sensor components apart from the user. The acquisition server's communication module handles the data flow from the sensors, and data requests from the sensors are replaced with network requests. Through the Wi-Fi network, the acquisition server may communicate with the regular Internet network.

The sensor abstraction in the sensor service layer enables the application to use sensor data as if it were physically linked to the equipment it runs on. The message broker collects the data and distributes it to its subscribers, which in our system is the data processing layer, which filters, smooths, and analyzes the data. Figure 2.11 depicts the IoT server's whole data flow diagram.

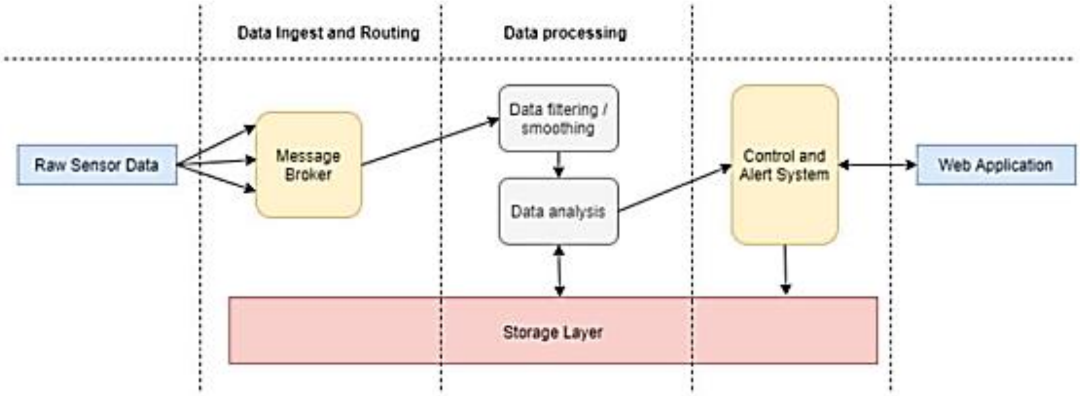


Figure 2.11 Functional and dataflow diagram for the IOT server

2.2.5 IOT Based Weather Reporting System Using Arduino and Node MCU

This project is proposed by Meera.S, Sharmikha Sree.R, Kalpana R.A, S.R.Manasvinii, Haritha.V, and Dr. K. Valarmathi [7]. The suggested system uses different sensors, an Arduino, and a Node MCU to read and display data on a server. The data obtained is also quite exact due to the high accuracy of the sensors. By connecting Node MCU to the server, the results are shown. The server is open source, and it offers a variety of ways to present data, either individually in feeds or in dashboards. The data can also be downloaded or fed straight into other programs using the server's interface.

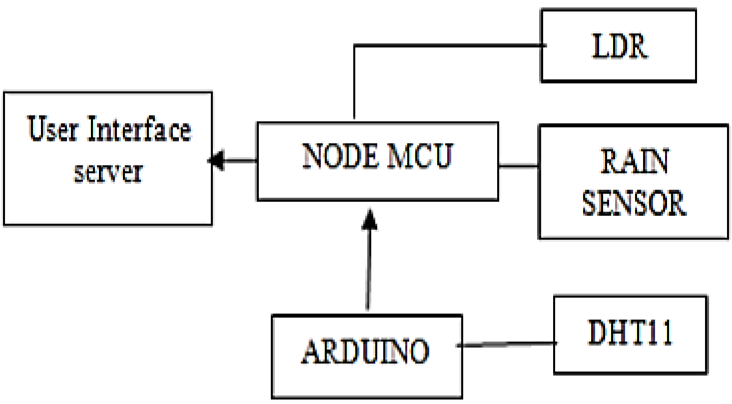


Figure 2.12 The weather reporting system IOT system architecture.

Historical datasets for the target system and statistics from IoT sensors for the test system are gathered for input into the system. The qualities of the data are identified once they have been obtained. The training and testing of the algorithm are the next steps in the process. The datasets are trained and tested using multiple linear regression and multi-target regression.

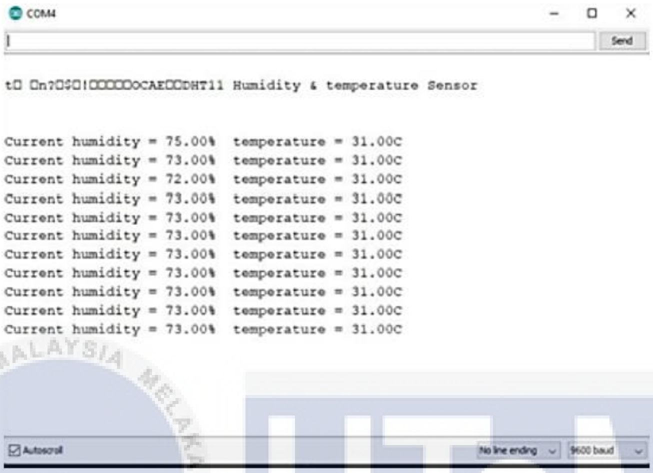


Figure 2.13 The Implementation in serial monitor of laptop server

Figure 2.14 The Implementation in serial monitor of laptop server.

As a result, this system is capable of both weather forecasting and monitoring. The Internet of Things is used for monitoring, while machine learning algorithms and methodologies such as multiple target regression are used for prediction. This strategy is highly effective for forecasting and hence aids in future planning. Whether it's an unforeseen natural disaster or an unexpected attack, this strategy aids in early preparation.

2.2.6 IoT based Data Logger System for weather monitoring using Wireless sensor networks

This project is proposed by Kondamudi Siva Sai Ram and A.N.P.S. Gupta [8]. The system may conduct control operations to react quickly to the sensor output. The microcontroller monitors and reads the sensors when the Wi-Fi module is set up in client mode. The temperature sensor and LDR will supply the data needed to continuously control the fan and light. The fan automatically turns on when the range temperature is raised to maintain the desired temperature. Furthermore, the LED lights will quickly turn on if the light dims. The other two sensors are likewise constantly monitored and updated. Still, since humidity and CO2 are unusual components in weather analysis, data from these two sensors are only received when the stated limit is exceeded. If the relative humidity in the air is greater than 35 °C, the sensor outputs logic 1 (high), otherwise logic 0. (low). CO2 sensors, on the other hand, provide data ranging from 1 to 0 based on the quantity of CO2 in the air.

A microcontroller (LPC2148) operates as the system's central processing unit and may be linked to all sensors and gadgets in the final system. The microprocessor can control and retrieve data from the sensors and perform data analysis before uploading it to the internet through the Wi-Fi module. The essential components of the proposed system are shown in the block diagram below.

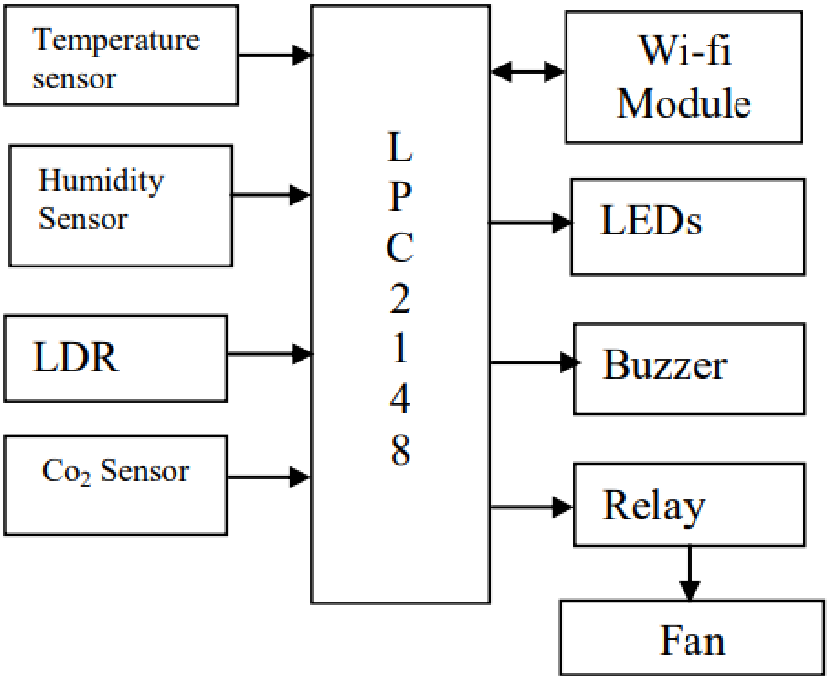


Figure 2.15 Block Diagram.

2.2.7 IoT Based Weather Monitoring System

This project is proposed by Prof. S.B. Kamble, P.Ramana P. Rao, Anurag S. Pingalkar, Ganesh S. Chayal [9]. System functionality refers to the entire operation of the system after all hardware and software have been connected. The system comprises three phases: one for receiving data from sensors; another for reading data from EEPROM; and finally for transferring data to the server (web page).

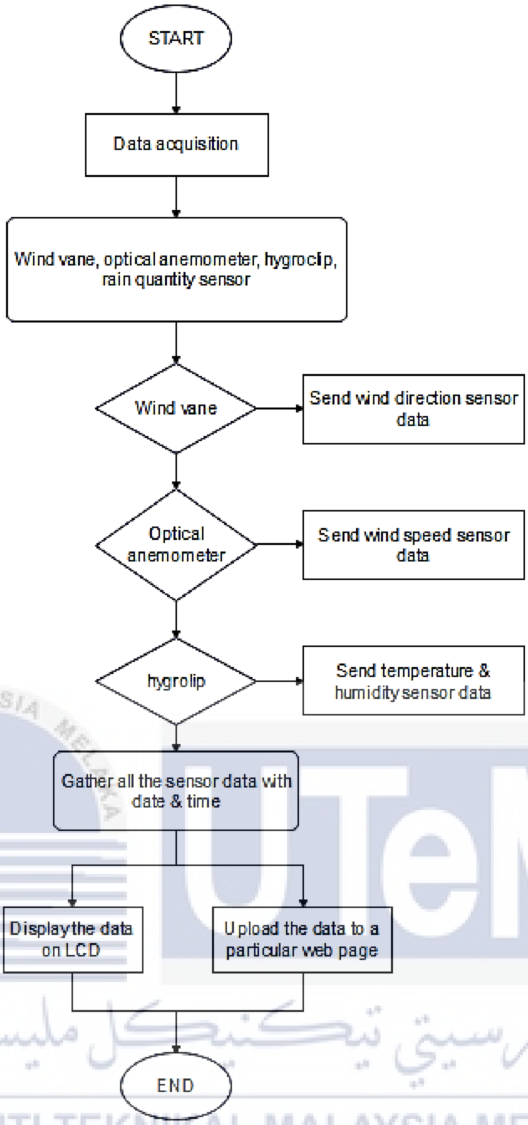


Figure 2.16 Flowchart system functionality.

S Some sensors produce analog outputs, while others produce digital outputs directly. The ADC converts the analog output of sensors into digital form. Our system's data acquisition block handles this part of the operation. Using a microcontroller, it collected all of this information and saved it in an EEPROM. The sensor data have been shown on the LCD and sent to the server, and a plot of the sensor values may be generated in the channel.

2.2.8 Smart weather monitoring and real time alert system using IoT

This project is proposed by Yashaswi Rahut, Rimsha Afreen, Divya Kamini [10]. The system comprises an Arduino Uno, which serves as the core processing unit and to which any sensors and devices may be connected. The microcontroller can operate the sensors, collect data from them, and analyze the data. The processed data may be posted and saved on a website to function as a database using Node MCU and Ubidots.

Humans were able to construct a more cost-effective design since the sensors in this product are far less expensive than those in earlier weather monitoring systems. These sensors gather data, which is then posted to a website and shown as graphical statistics. The content on this website may be accessed from anywhere on the globe. The information obtained on these web pages might be helpful in the future. Unlike the current system, which requires the physical delivery of data, this new method does not require physical data transmission.

2.2.9 IoT Based Weather Monitoring System for Effective Analytics

This project is proposed by Ferdin Joe John Joseph [11]. The hardware that connects to the Raspberry Pi, as well as the software that manages the data gathered by the sensors, Although the objective of this framework stores data on a cloud server, and the implementation tracks data availability throughout a subnet's intranet. The gathered data is available for download in CSV format, and the most current weather data is provided in JSON format for exchanging data over the network..

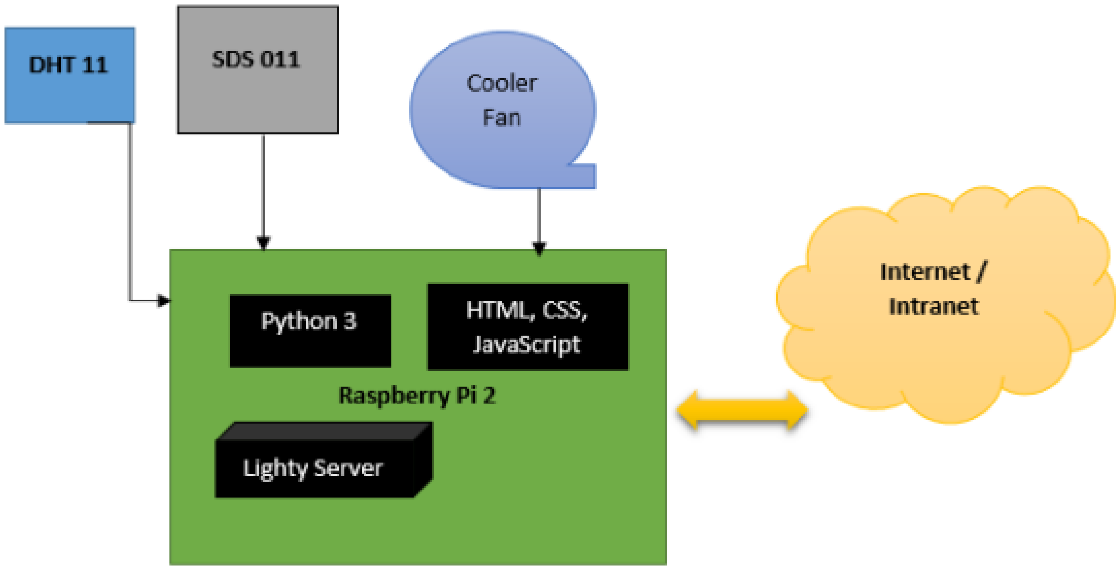


Figure 2.17 Architecture of Proposed System.

Figure 2.17 depicts the whole architecture of the system implementation. It shows the Raspberry Pi's hardware as well as the software that handles the data obtained by the sensors. Although the framework's purpose is to store data on a cloud server, the implementation only reports on data availability inside a subnet's intranet. The acquired data is accessible in CSV format for download, and the most recent weather data is available in JSON format for data exchange over the network.

2.2.10 IOT Based Weather Monitoring and Reporting System Project

This project is proposed by Anita M. Bhagat, Ashwini G. Thakare, Kajal A. Molke, Neha S. Muneshwar, Prof. V. Choudhary [12]. The Arduino Uno is used in an IoT-enabled weather monitoring system project to measure four weather parameters using four different sensors. Temperature, humidity, wetness, and rain level sensors are just a few alternatives. The Arduino Uno is directly connected to these four sensors. The analog-to-digital converter is included inside the Arduino Uno. These meteorological variables are calculated and displayed on an LCD screen using Arduino. The parameters are sent over the Internet using Internet of Things (IoT) technology. After a long time has elapsed, the practice of delivering data to the Internet through Wi-Fi is repeated. The user must go to a particular website to obtain the weather data. The project creates a connection to a web server and saves the data there. As a result, the user receives real-time weather information. For this IoT-based weather monitoring reporting system project, you'll need access to the Internet or a Wi-Fi connection.

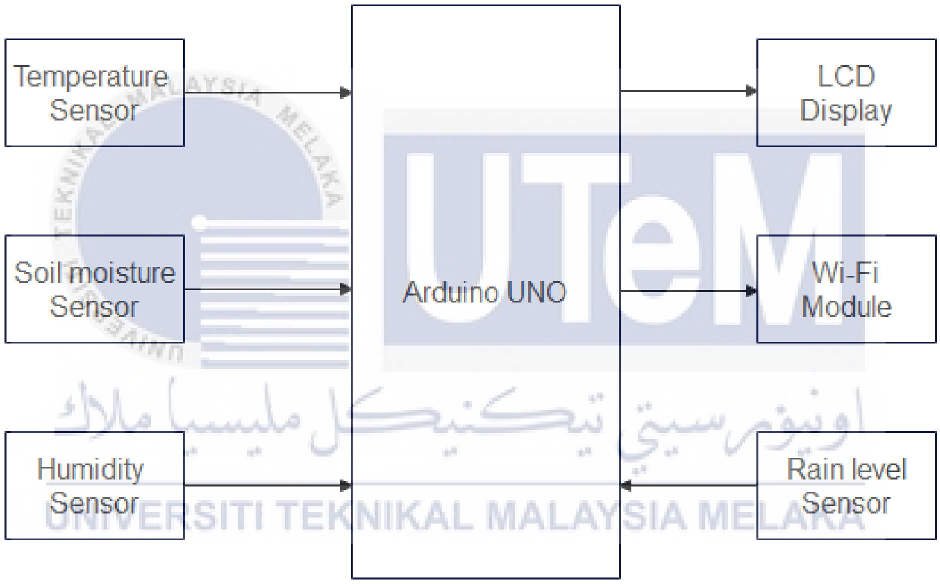


Figure 2.18 System Design.

2.2.11 IoT based- Advanced Weather Monitoring System

This project is proposed by Pranavi Yadav, Nimish Nigam, Ranjeeta Yadav and Sanjay Kr. Singh [13]. The project is a leading IoT-enabled weather monitoring system that allows users to get real-time data through a user-friendly interface provided by a thin client such as a web browser or a mobile application. As a result, users no longer need to rely on a weather forecasting service to verify the facts and weather conditions in their location online ahead of time. If the weather worsens, the system has a great feature that sends push notifications to the client's phone. This prompts the consumer to take the necessary actions right away.

Changes in meteorological factors such as wind speed, temperature, and humidity are tracked by the system. The weather data will be accessible in real-time through a customized website and a mobile app.

2.2.12 Weather Monitoring System Using Iot And Cloud Computing

This project is proposed by Mr. Mohit Tiwari, Deepak Narang, Priya Goel, Anupma Gadhwal, Abhinav Gupta And Ankush Chawla [14]. Weather monitoring is a critical use of the Internet of Things concept. It requires detecting and storing a wide range of meteorological variables and employing them for warnings, notifications, and long-term research. We'll also utilize graphs to attempt to detect and highlight patterns in the parameters. The instruments used for this purpose gather, organize, and display data. The Internet of Things is expected to change the world by monitoring and regulating environmental events using sensors and systems that can record, analyze, and transmit meteorological data.

Cloud computing refers to the ability to access resources such as data storage and processing power on a computer system without having to manage them directly. Once the sensor readings are uploaded to the cloud, the numbers are processed, and the user is informed if any of the parameter's particular values go outside their normal range. A graph is also created to show the trends.

2.2.13 Weather Reporting System using Internet of Things

This project is proposed by Dr. Ashpin Pabi.D J, Muneendra.D, Ramanath Reddy.N, Mohammad Yusuf.S, Kiran Kumar.D [15]. Carriers mainly use the Internet of Things (IoT) weather reporting technology. The majority of the time, this software will be used in agriculture. Weather plays a varied role in different scenarios most of the time. Thus, this system can get weather readings for agriculture and other tasks. It's also utilized to determine the weather in places like rainforests, volcanoes, and so on. With NodeMCU, this system is entirely automated. There is no requirement for human intervention.

In conclusion, this project presents a clear picture of a system that can monitor weather readings using wireless and IoT technology. The sensors will interact with the hotspot's Wi-Fi and surrounding areas to improve wireless connection. The sensor data can be shown on the Blynk App by the system. This can be downloaded through the Google Play Store or the App Store.

2.2.14 IoT-Based Data Logger for Weather Monitoring Using Arduino-Based Wireless Sensor Networks with Remote Graphical Application and Alerts

This project is proposed by Jamal Mabrouki , Mourade Azrour, Driss Dhiba, Yousef Farhaoui, and Souad El Hajjaji [16]. At three distinct locations with various traffic levels, this technology was employed to monitor meteorological and air information from their surroundings. The system was put near two roads, one with low traffic (Site 1) and the other with heavy traffic (Site 3). (Site 3). The sensors are capable of detecting and measuring a variety of items. The collected data is sent to an Arduino card for on-the-fly processing. After a basic treatment, the Arduino card sends the processed values to a computer with a database. A web page may be used to access the information remotely. When an unfavorable value is discovered, an email notice is sent to the end user.

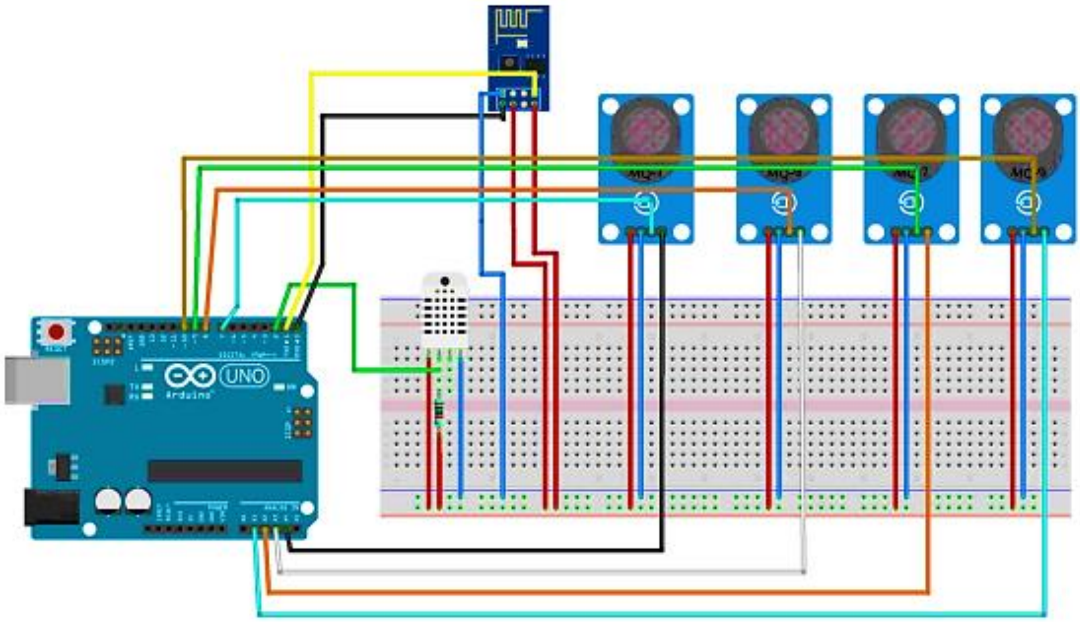


Figure 2.19 The proposed model hardware implement. .

This has resulted in the creation of a new application to monitor air and weather conditions, as shown in Figure 2.19. This program allows you to monitor both the weather and the qualities of the air. An Arduino card serves as the core management unit in the final construction. All sensors and equipment might then be attached to it, either directly or indirectly. Weather and air data may be collected through the sensors that are connected to the internet. The collected data is sent to an Arduino card for on-the-fly processing. After a basic treatment, the Arduino card sends the processed values to a computer with a database. A web page may be used to access the information remotely. When an unfavorable value is discovered, an email notice is sent to the end user.

2.2.15 Weather Monitoring System Using Arduino Uno

This project is proposed by Vaishnavi Gotmare, Rajesh Kolte, Rutwik Thengodkar [17]. This system was created to save data collected at pre-determined sampling intervals, as well as date and time stamps, for later retrieval and analysis of various environmental parameters such as temperature, humidity, atmospheric pressure, wind speed, wind direction, air quality, light intensity, rainfall amount, and location coordinates. It consists of an Arduino UNO (microprocessor) that acts as a gateway for data and information collection from numerous probes.

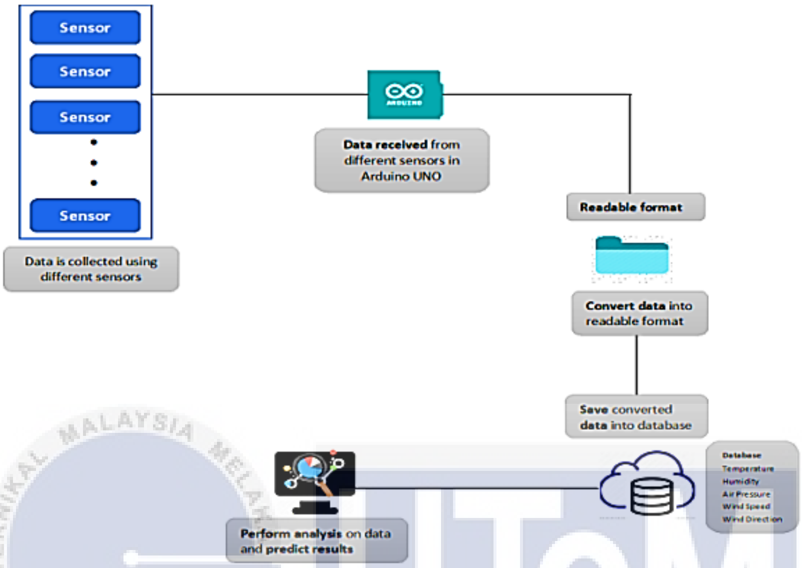


Figure 2.20 Architectural diagram of the system.

2.2.16 Development of an ESP-32 Microcontroller Based Weather Reporting Device

This project is proposed by T. E. Babalola, A. D. Babalola and M.S. Olokun [18]. With an ESP-32 microcontroller, the system is a new way to report the weather. The ESP-32 is the main microcontroller, and digital and analog pins connect the sensors to it. The readings from the sensors in the environment are sent to an ESP-32 microcontroller, which processes them and then shows the results on the ESP serial monitor. The display unit put the data from the sensor on the LCD screen. The solar power pack gives the LCD and the rest of the system the power they need to work. The plan's block diagram is shown in figure 2.21.

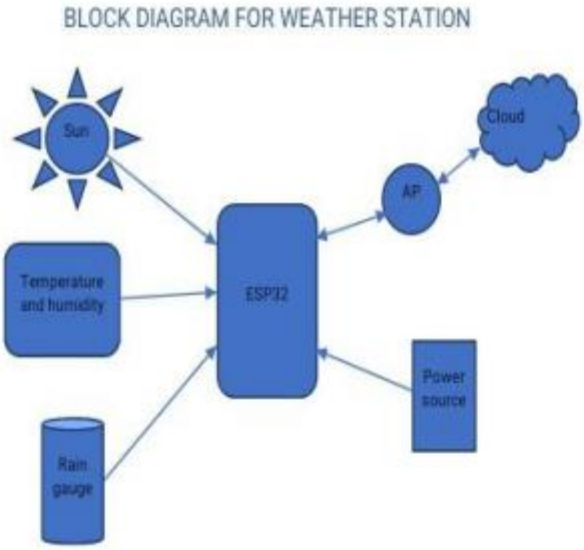


Figure 2.21 Block diagram for weather station.

A C-code generator for ESP-32 microchip was used to put the software design into action. This compiler has an improved C compiler program and better functions for a lot of microcontroller tasks. Figure 2.22 is a flowchart that shows how the algorithm is used to program the ESP32 microcontroller.

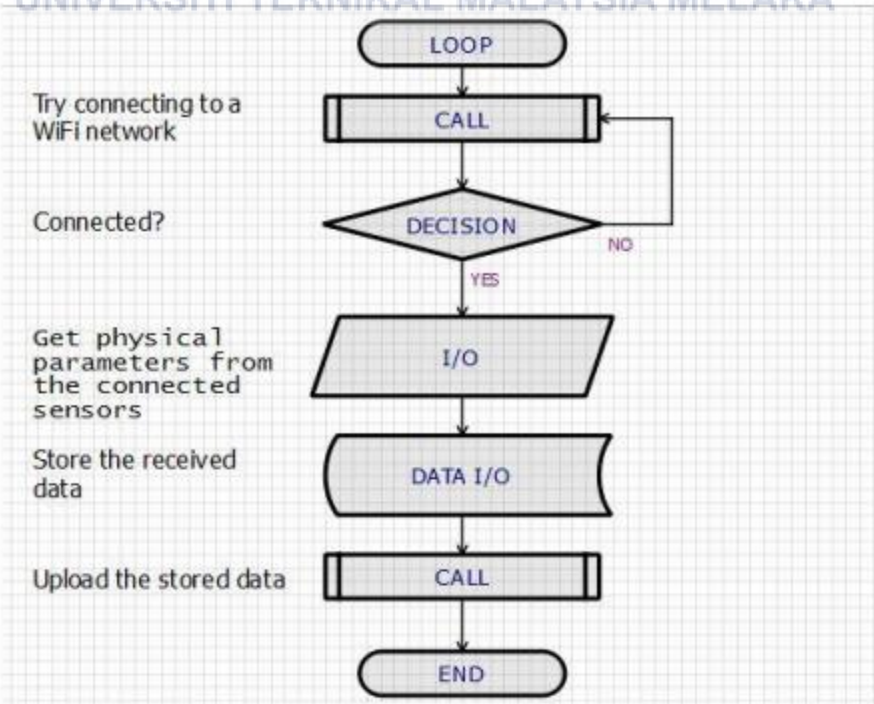


Figure 2.22 Diagram for flowchart weather station.

2.2.17 Real-Time Cloud based Weather Monitoring System

This project is proposed by Neha Kumari, Sakshi, Shivani Gosavi and Sandeep s. Nagre [19]. A weather monitoring system comprises a NodeMCU, a DHT11 sensor, a BMP180 pressure sensor, a light-dependent resistor, a rain sensor, and a gas sensor. These sensors can measure temperature and humidity, pressure, the amount of smoke in the air, the brightness of the light, and the amount of rain. This system uses the MQTT protocol to publish and subscribe and uses graphs to monitor environmental parameters. The collected data is sent to the cloud via the internet. The Thinkspeak platform looks at data in real time. Data can be accessed and controlled remotely from mobile computers and laptops connected to the internet.

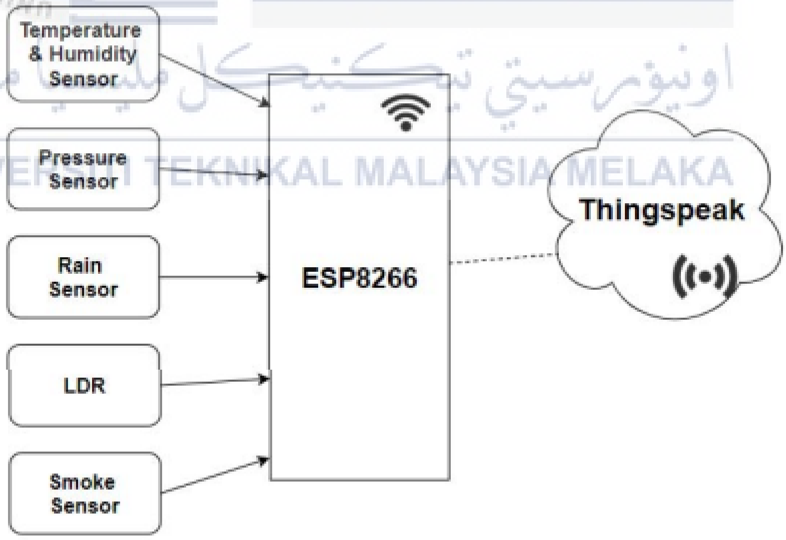


Figure 2.23 Architectural diagraam of real-time system.

The system is made up of different sensor nodes, an IoT cloud platform called Thinkspeak, and a microcontroller with Wi-Fi built in (esp8266). Sensor nodes measure the weather parameters, and the ESP8266 microcontroller sends the data to Thingspeak.

Temperature, rainfall, wind speed, wind direction, dew point, atmospheric pressure, humidity, light intensity, and the amount of smoke in the air are all weather variables. Out of the nine above parameters, only wind speed and direction are not connected in this journal. The ESP8266 microcontroller is used to process and sense information. This system uses Thinkspeak as its cloud platform because it is an open data platform for Internet of Things (IoT) applications.

All of the graphs below show how different Thinkspeak parameters are connected and how they change over time. The system is set up to keep track of different environmental conditions and to measure the necessary data, which is then shown on charts in a clear way. Updated data from sensors in real time on the Thinkspeak cloud and let users look at the parameters on a webpage.

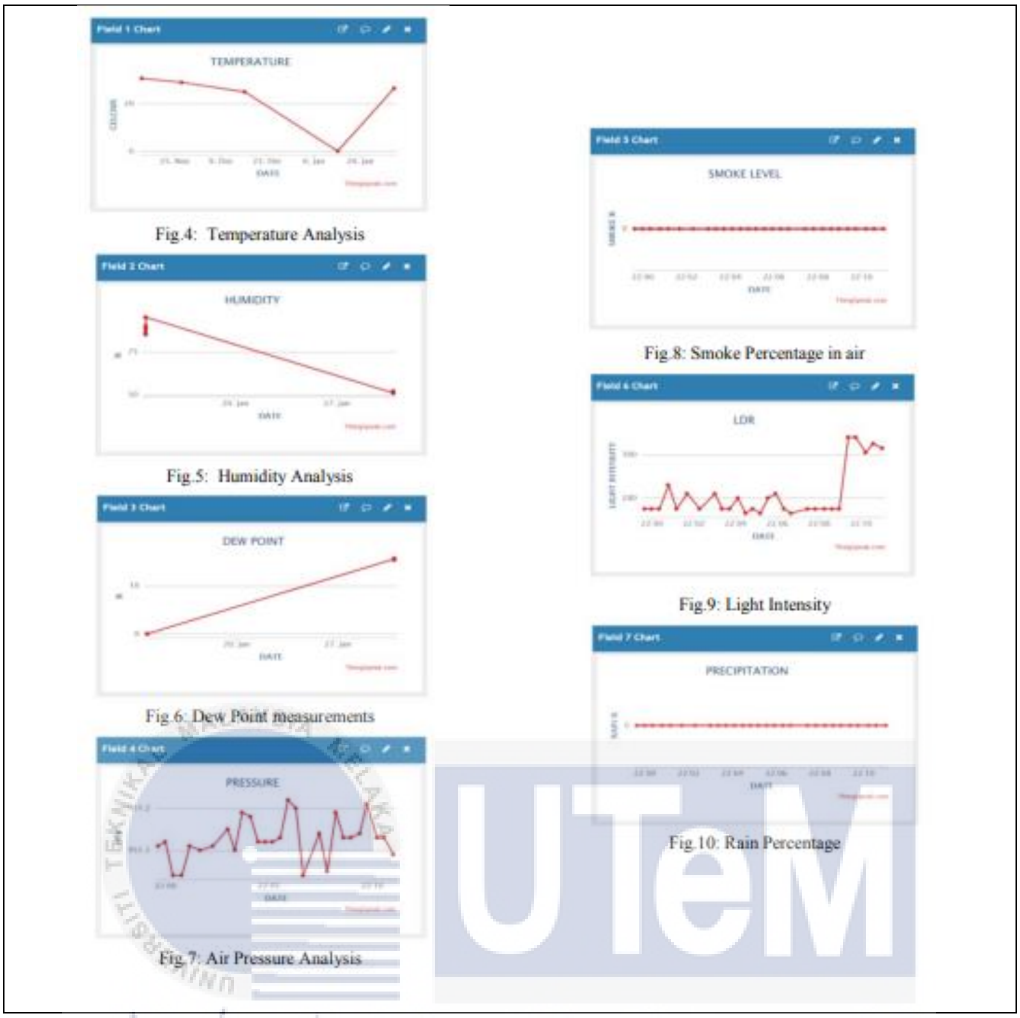


Figure 2.24 The result of Thinkspeak channel status.

2.2.18 Efficient IOT based Weather Station

This project is proposed by Abu Saleh Bin Shahadat, Safial Islam Ayon, and Most. Rokeya Khatun [20]. It created an affordable NodeMCU-based Wireless Weather Station with IOT-based graphical software applications for simple weather monitoring data. Temperature, pressure, humidity, rainfall, and altitude can all be measured. Comparing the results to those from other cost-effective weather stations, they showed good accuracy and stability.

It features a very low maintenance cost and an intuitive interface. Future updates could provide other enhancements, including wind direction, solar radiation, and precipitation. More research can be conducted to make the system as affordable as possible.

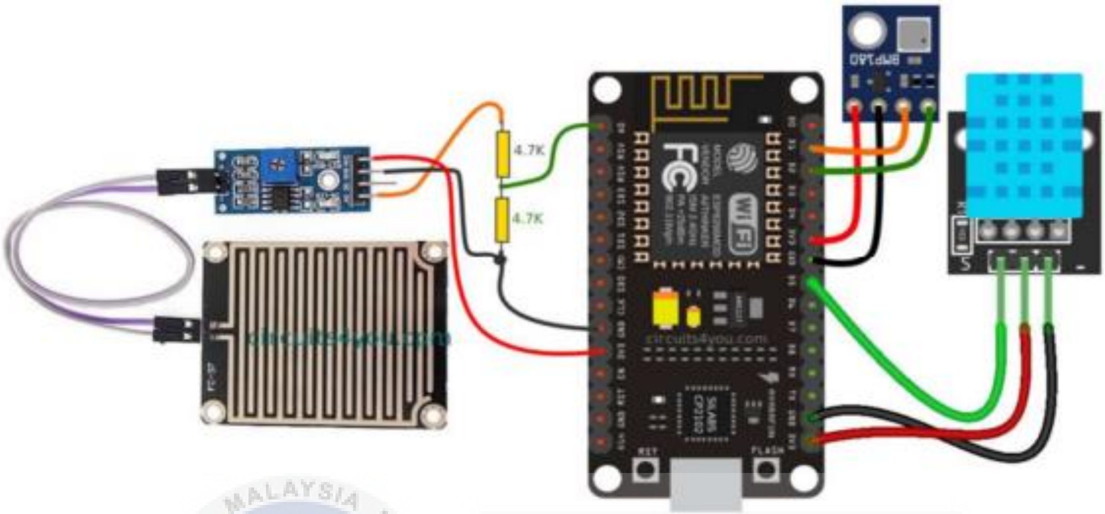


Figure 2.25 The proposed model hardware implement.

The recommended hardware design is shown in figure 2.25. The tasks are complete by using planned system's following step. Phase 1, after receiving an input signal, sensors first create a signal on an output pin. The ESP8266 module processes the sensor output in Phase 2 and sends it through a Wi-Fi connection to the cloud. Phase 3, the Blynk-IoT software collects the inputs through virtual terminals and displays them as desired.

This project has several additional benefits. First, the full automation of the IOT weather monitoring system. Second, it doesn't need any human involvement. Third, we are able to receive weather alerts in advance. Fourth, this system is more cost-effective and requires less work. Fifth, high accuracy. Sixth, clever method for examining the environment. Last but not less, effective.

2.3 Summary review of related works

Table 2.1 Comparision table of previous projects

Bil	Reference	Equipment/ components	Features
1.	Internet of Things (IOT) Based Weather Monitoring system Bulipe Srinivas Rao, Prof. Dr. K. Srinivasa Rao, Mr. N. Ome. (2016)	Arduino UNO, Temperature sensor, CO sensor, Sound sensor, LDR light sensor.	They found a good implementation model that includes a variety of sensors and other components. In this implementation paradigm, the Arduino UNO board with a Wi-Fi module was used as an embedded device for sensing and storing data in the cloud. On the Arduino UNO board, analogue input pins, digital output pins, an inbuilt ADC, and a Wi-Fi module connect the embedded device to the internet. The sensors are monitored using an Arduino UNO board. The relevant sensor reading will be transformed into a digital value, from which the linked environmental parameter will be determined.
2.	Arduino-Based Weather Monitoring System Ejodamen Pius Uagbae, Ekong, Victor Eshiet, Inyang, Udoinyang Godwin (2018)	Arduino UNO, Temperature and Humidity Sensor	The system's functionality refers to its overall operation once all peripherals and software have been integrated. The system comprises three phases: one for receiving data from sensors, another for reading data stored in EEPROM, and a third for delivering data to the server (web page).

Bil	Reference	Equipment/ components	Features
3.	Weather Station Design Using IoT Platform Based On Arduino Mega Medilla Kusriyanto, Agusti Anggara Putra (2018)	Arduino Mega, Rain Sensor, Air Pressure sensor, Air and Humidity Sensor.	An Arduino mega 2560 microcontroller acts as the data processing core for the weather forecast system, which also includes a DHT 22 temperature sensor, a rain detection sensor, a BMP 180 air pressure sensor, a DS3231 RTC, a 3.5" touch screen LCD, and an SD data storage card.
4.	IOT Based Weather Reporting System Vera Svet , Vasile Gurduza, Eduard Medinschi (2020)	Arduino UNO, Temperature and Humidity sensor.	The acquisition server implementation is a completely unique system due to the design approach, especially in terms of sensor connectivity. Sensor services are segregated from sensor components. The data from the sensors is sent via the acquisition server's communication module, and sensor data requests are replaced with network requests. The acquisition server may communicate with the normal Internet network through the Wi-Fi network.
5.	IOT Based Weather Reporting System Using Arduino and Node MCU Meera.S, Sharmikha Sree.R, Kalpana R.A, S.R.Manasvinii, Haritha.V and Dr. K. Valarmathi (2021)	Arduino UNO, Node MCU, Rain sensor, Temperature and Humidity sensor and Ldr light sensor.	The suggested system reads and displays data in real time using several sensors, an Arduino, and a Node MCU. Due to the high accuracy of the sensors, the data acquired are likewise highly precise. The findings are shown by connecting Node MCU to the server. The server is open source, and it provides a number of data presentation options, including individual feeds and dashboards. You may also use the server to get the data or import it straight into other programs.

Bil	Reference	Equipment/ components	Features
6	IoT based Data Logger System for weather monitoring using Wireless sensor networks Kondamudi Siva Sai Ram, A.N.P.S.Gupta (2016)	LCP2148, Temperature sensor, Humidity sensor, LDR sensor, CO sensor	By regulating tasks, the system may also take immediate action depending on sensor data. The Wi-Fi module will be configured in client mode at first. The microcontroller will then keep an eye on and read the sensors. The temperature sensor and LDR will continually provide the data required to regulate the fan and light. If the temperature rises over the set point, the fan will immediately switch on to maintain the desired temperature. If it begins to grow dark, the LED lamps will automatically switch on. The other two sensors are likewise examined and updated on a regular basis. We only read the data from these two sensors when they go over or below a specified limit since humidity and CO2 are not normally major weather elements. If the relative humidity in the air is more than 35 degrees Celsius, for example, the sensor outputs logic 1 (High) (Low). CO2 sensors, on the other hand, provide a 1 or 0 depending on the amount of CO2 in the air.
7	IoT Based Weather Monitoring System Prof. S.B. Kamble, P.Ramana P. Rao, Anurag S. Pingalkar, Ganesh S. Chayal (2017)	LPC2138, Wind vane, Optical Anemometer, Hygroclip and Rain Quantity sensor	The system functionality includes how the whole system works after all of the peripherals and software have been added. The system has three parts: reading the information from the sensors, reading the information from the EEPROM, and sending the information to the server (web page).

Bil	Reference	Equipment/ components	Features
8	Smart weather monitoring and real time alert system using IoT Yashaswi Rahut, Rimsha Afreen, Divya Kamini (2018)	Arduino UNO, NodeMCU, Dark sky.net, LDR light sensor, CO sensor, Humidity sensor, light sensor, anemometer.	The solution makes use of an Arduino Uno, which serves as the system's primary processing unit. All of the sensors and gadgets may be linked to the microcontroller. The sensors may be utilized by the microcontroller to gather data, and the data from the sensors can be used to do analysis. NodeMCU and Ubidots are both capable of uploading and storing processed data on a website, which may then be utilized as a database.
9	IoT Based Weather Monitoring System for Effective Analytics Ferdin Joe John Joseph (2019)	Python, Javascript, Temperature and humidity sensor.	The sensor data management software as well as the hardware that connects to the Raspberry Pi. Although the goal of this framework is to store data on a cloud server, the way it functions shows that data may be accessed through a subnet's intranet. The collected data may be downloaded in CSV format, and the most current weather data can be sent across the network in JSON format.
10	IOT Based Weather Monitoring and Reporting System Project Anita M. Bhagat, Ashwini G. Thakare, Kajal A. Molke, Neha S. Muneshwar, Prof. V. Choudhary (2019)	Arduino Uno, Temperature sensor, Soil moisture sensor, Humidity sensor, Rain level sensor.	In an IoT-enabled weather monitoring system project, an Arduino Uno monitors four meteorological parameters. Sensors for temperature, humidity, wetness, and rain. Four sensors are linked to the Arduino Uno. An ADC is built into the Arduino Uno. The weather parameters are shown on the LCD via Arduino. These parameters are sent to the Internet through IOT. Data is sent through Wi-Fi at regular intervals. A user must go to a website to view weather data. The data for the project is stored on a web server. Users can obtain real-time weather updates. The Internet or Wi-Fi is required for this IoT-based weather monitoring reporting system.

Bil	Reference	Equipment/ components	Features
11	IoT based- Advanced Weather Monitoring System Pranavi Yadav, Nimish Nigam, Ranjeeta Yadav and Sanjay Kr. Singh (2020)	Arduino ATmega 328, ESP 8266 Wi-Fi Module, Temperature and Humidity sensor.	The recommended system is a prominent weather monitoring solution that incorporates IOT, allowing real-time data to be accessed through a thin client such as a web browser or mobile app. Without using a weather forecasting agency, the user may verify the data and weather conditions of their area online. If the weather worsens, the system sends push alerts to the client's phone, urging them to take precautionary measures.
12	Weather Monitoring System Using Iot And Cloud Computing Mr. Mohit Tiwari, Deepak Narang, Priya Goel, Anupma Gadhwal, Abhinav Gupta And Ankush Chawla (2020)	Node MCU, Temperature and Humidity Sensor, Barometric Pressure Sensor, LDR sensor, Raindrop Module Sensor.	Weather monitoring is a practical use of the Internet of Things concept that entails detecting and recording a variety of weather characteristics for warnings, messages, altering appliances, and long-term research. We'll also attempt to see how parameter patterns change over time. Data is collected, organized, and displayed using these devices. By using sensors/devices that can record, analyze, and broadcast meteorological characteristics, the internet of things will alter the globe by monitoring and regulating environmental occurrences. Cloud computing refers to the availability of data storage and processing capacity without the need for user administration.
13	Weather Reporting System using Internet of Things Dr. Ashpin Pabi.D J, Muneendra.D, Ramanath Reddy.N, Mohammad Yusuf.S, Kiran Kumar.D (2021)	Node MCU, Temperature and Humidity sensor, Raindrop sensor.	Most of the time, formers use the IOT weather reporting system. This app will be used in agriculture most of the time. Most of the time, the weather plays a different role in each situation, so this system can be used to get a reading on the weather while farming or doing other work. It is also used to find out the weather in places like rain forests, volcanoes, and so on. Node MCU is used to control every part of this system. No need any help from people.

Bil	Reference	Equipment/ components	Features
14	IoT-Based Data Logger for Weather Monitoring Using Arduino-Based Wireless Sensor Networks with Remote Graphical Application and Alerts Jamal Mabrouki , Mourade Azrour, Driss Dhiba, Yousef Farhaoui, and Souad El Hajjaji (2021)	Arduino UNO R3 card, Temperature sensor, Humidity sensor, pressure sensor.	This system was created to monitor the weather and air quality at three separate locations with varying levels of traffic. The system was installed near a low-traffic road (Site 1) and a busy road (Site 2). (Site 3). The sensors can detect and measure a variety of items, including The data is transferred to the Arduino card, where it will be processed. The Arduino card transfers the values that have been modified to a computer with a database after a simple step. Using page online, you may access the information you've stored from away. When an undesirable value is discovered, an email warning is delivered to the end user.
15	Weather Monitoring System Using Arduino Uno Vaishnavi Gotmare, Rajesh Kolte, Rutwik Thengodkar (2021)	Arduino Uno, Temperature and Humidity sensor, Pressure sensor, Anemometer sensor, CO sensor, Light Intensity sensor, Rain gauge sensor, co-ordinate sensor.	This system was created to save data collected at pre-determined sampling intervals, as well as the date and time, in order to retrieve it later with real-time alerts for monitoring and analyzing various environmental parameters such as temperature, humidity, atmospheric pressure, wind speed, wind direction, air quality, light intensity, rainfall amount, and location coordinates. It is made up of an Arduino UNO (a microcontroller) that acts as a gateway for collecting data and information from numerous probes.
16	Development of an ESP-32 Microcontroller Based Weather Reporting Device T. E. Babalola, A. D. Babalola, M. S. Olokun (2022)	ESP32, Temperature and Humidity Sensor, Tipping Bucket, Light Sensor, LCD, Resistor, Capacitor,Solar Panel, Voltage Regulator	This explains the techniques used to design the hardware and software components. The main part of the hardware is the DHT22, TS2651 sensor and how it connects to the microcontroller and display unit. The research methods considerations and specifications of the weather reporting system were also written down. It is made with software called Proteus. Once turned on, the microcontroller is set up to get signals from the sensors.

Bil	Reference	Equipment/ components	Features
17	Real-Time Cloud based Weather Monitoring System Neha Kumari, Sakshi, Shivani Gosavi and Sandeep s. Nagre (2020)	ESP8266, Temperature and Humidity Sensor, Pressure Sensor, Rain Sensor, Light Dependent Resistor, Gas Detector	NodeMCU, DHT11, BMP180, LDR, rain, and gas sensors make a weather monitoring system. These sensors measure pressure, temperature, humidity, smoke, light, and rain. This system monitors environmental parameters using graphs and the MQTT protocol. The cloud receives data gathered online. Thinkspeak appears to analyze data live. Internet-connected laptops and mobile devices may access and control data.
18	Efficient IoT based Weather Station Abu Saleh Bin Shahadat, Safial Islam Ayon, Most. Rokeya Khatun (2020)	Esp8266, Pressure Sensor, Temperature and Humidity Sensor, Rain Sensor	Wireless Weather Station with IOT-based graphical software for easy weather monitoring. It calculates altitude, pressure, humidity, rainfall, and temperature. The results were accurate and stable. It is easy to use and low-maintenance. Add wind direction, sun radiation, and precipitation later. Research can make the system cheaper.

.

2.4 Summary

According to research findings that have been conducted as part of previous projects that are also relevant, we are able to declare that there are various methods that can be used to create weather stations. During the life cycle of the project, the information collected may prove useful in meeting the main goals of the project. There are many aspects that need to be taken into account, such as the type of component that will be used, the cost of the component, and the way it will be used. As a result of this, we can see that every project has its positives and negatives, but as long as it has the potential to serve the community, it should be carried out.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology is a comprehensive recommendation collection that provides tool models for completing tasks across the software development life cycle. This set of guidelines is also known as the methodology. This chapter is focus on the approach employed in this project's development, including the block diagram and flow chart. In addition, the procedures for joining the networks be included, as well as the circuit diagrams, which be covered in detail. When designing a system, selecting an appropriate methodology is essential because it will allow for more efficient creation and operation. Rapid Application Development is the approach taken with this project's methodology (RAD). During the process of developing the Weather Station using the Internet of Things, Rapid Application Development was utilized (IoT).

3.2 Project Workflow

Developing a workflow chart that is well-organized and user-friendly is an essential step in determining and ensuring that a project is effective and goes smoothly. Putting together a strategy that is well thought out and organized is one of the most important steps toward achieving one's goals. After have finished all of planning, the next step is to perform research of some kind. It is necessary to carry out a project in order to make things simpler. The completion of an in-depth study enables the identification and avoidance of any and all potential problems during the implementation of a project.

As a direct consequence of this, the design of the project is developed, and then its actual execution follows. After the project was finished, it was evaluated to see how successful it was in meeting its objectives.

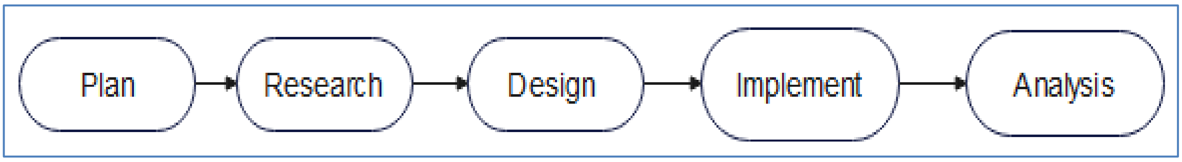


Figure 3.1 Project flow

3.2.1 Planning

The first thing that needs to be done in order to get things off to a good start with the planning process is to figure out what the goals of the project actually are. The purpose of the project is develop an advanced network of weather stations. Several elements are put into consideration and given critical analysis in order to ensure that all of the objective criteria are satisfied. Gantt charts were developed to assist in the completion of projects. The scope of the project is also constructed in such a way that it allows for the identification of project scope.

3.2.2 General Block Diagram

The project's block diagram is shown in the figure 3.2. This project has input from the sensors such as temperature and humidity sensor (DHT22), barometric pressure sensor (BMP180), rain sensor and light dependent resistors sensor (LDR). The microcontroller ESP32 is the processing device used in this project. The blynk program is utilized as the output device in this project.

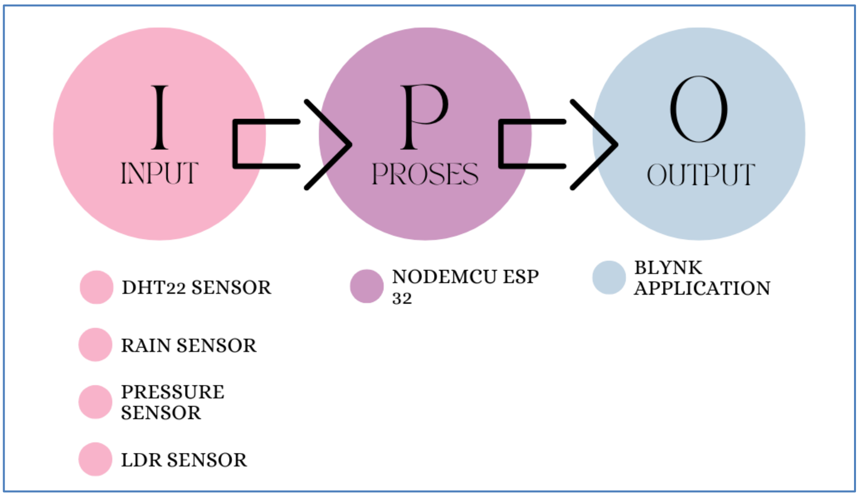


Figure 3.2 General block diagram

3.2.3 Gantt Chart

From the day one briefed on the PSM2 until the day of the presentation, when everything will be arranged and exhibited in the gantt chart, the gantt chart will aid in the completion of this PSM2. This has been the situation from the moment about the PSM2 until the day of the presentation.

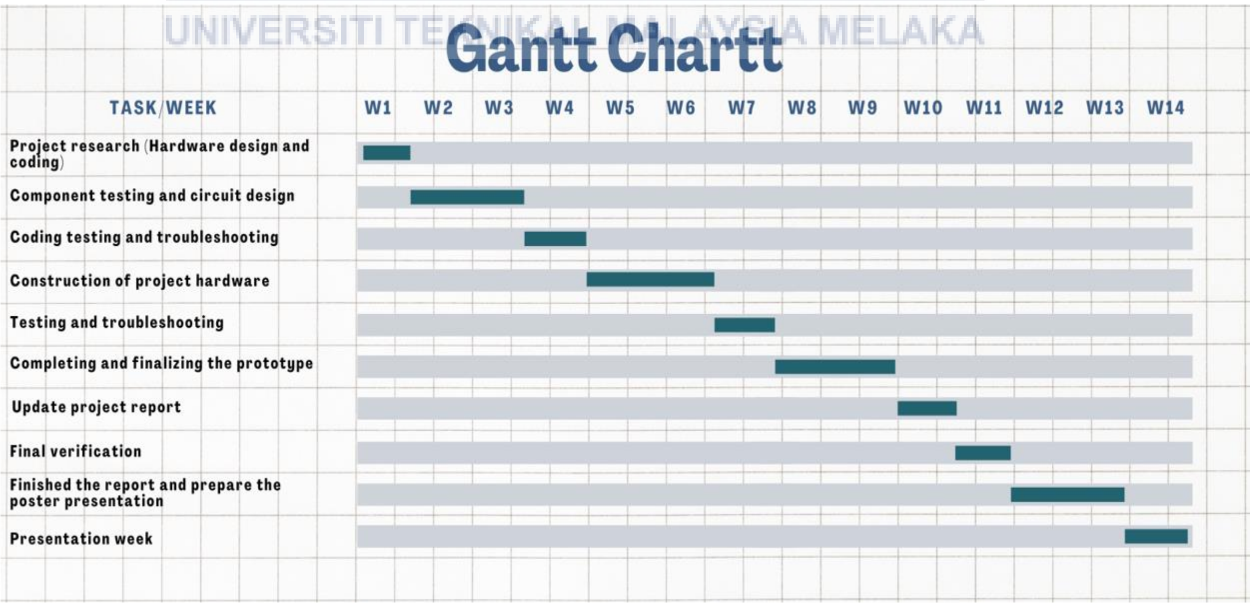


Figure 3.3 Gantt chart of project

3.3 Project Design

System design refers to the process of developing a software system's architecture as well as its modules, interfaces, and data in order to fulfill a set of standard requirements. The process of applying the ideas of system theory to the steps involved in the production of new products is referred to as system design. This weather station that is based on the Internet of Things is comprised of a framework, an architecture design, a process model and a flowchart.

3.3.1 Framework

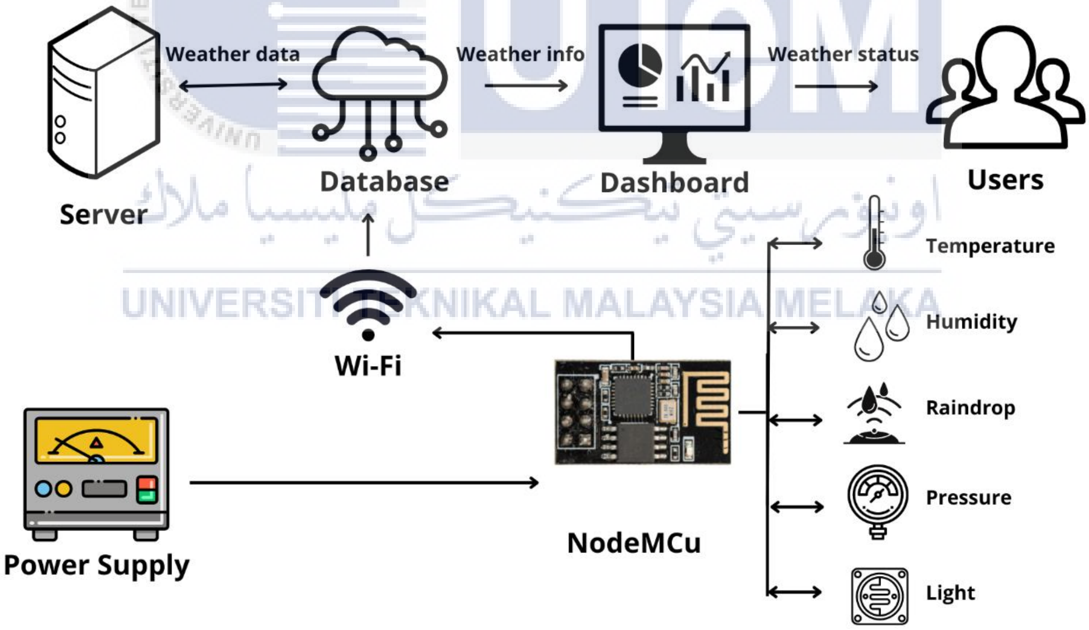


Figure 3.4 Framework of the project

In this framework, the weather station will send data to the database every 5 minutes, including information on the temperature, humidity, pressure, light, and volume of raindrops that have fallen. The weather data will be saved in the database, and the dashboard application will receive the saved weather data. The dashboard application gives customers the ability to monitor the weather data at any time.

3.3.2 Process Model

A context diagram and a data flow diagram compose the detail process model for a Weather Station utilizing the Internet of Things. Figure 3.4 depicts the context diagram for a weather station that uses the Internet of Things (IoT). In a one-way system, the user may only request weather data from the system.

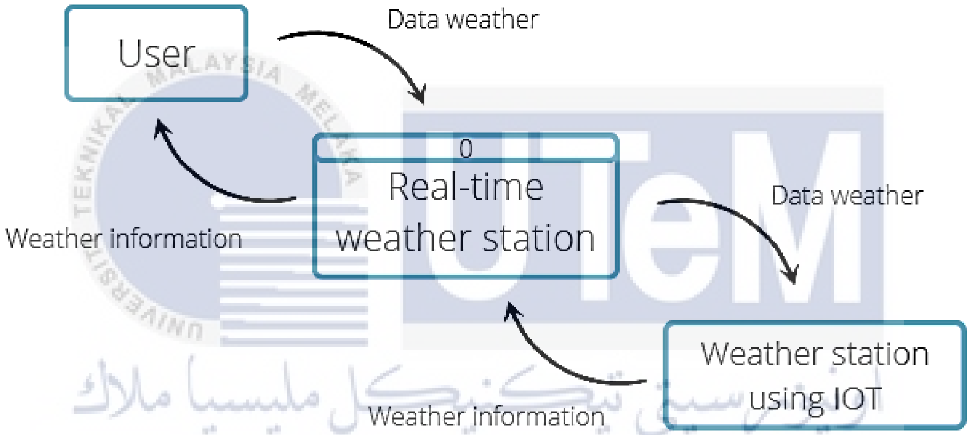


Figure 3.5 Context diagram of the system

A data flow diagram is a two-dimensional graphic that depicts the processing and transport of data inside a system. Entities, actions, and files make up the data flow diagram (DFD), which depicts how data travels through the system.

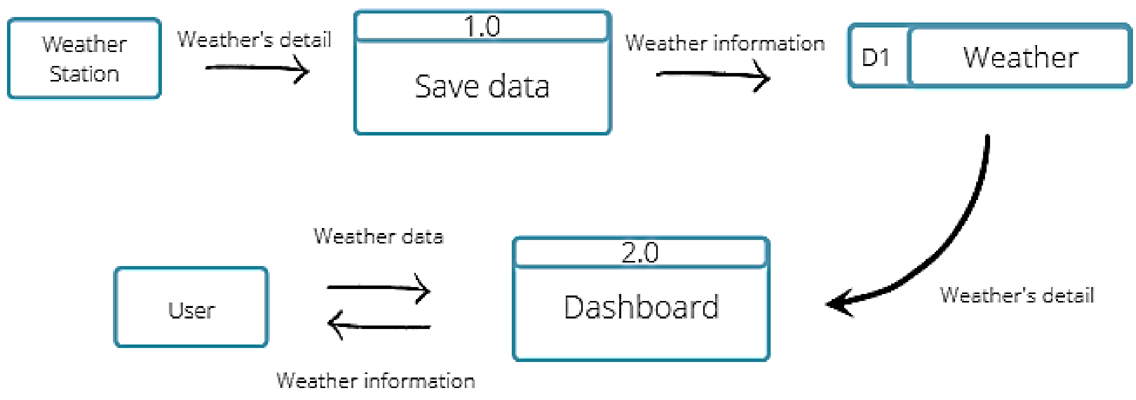


Figure 3.6 Data flow diagram of the system

The system has two procedures, as shown in Figure 3.5, which are saving data and dashboard. The first process involves transferring weather data from a weather sensor to a weather data database via the save data process. The second phase involves the user being able to check the data by dashboard, which is then provided to the user via dashboard.

3.3.3 Flowchart

The method for information about product users' progress must be designed into the project flow/sequence so that the project can function smoothly with all of the users' implementation. The project's flowchart is shown in figure 3.7.

The project starts with a wi-fi connection. When the user successfully connects with wi-fi, the weather station will send data to the database every 5 minutes. The weather data has been saved in the database, and the dashboard application will receive the saved weather data. This is to say that the dashboard of mobile applications (Blynk Apps) has value. Last, the Blynk application allows customers to monitor the weather data anytime.

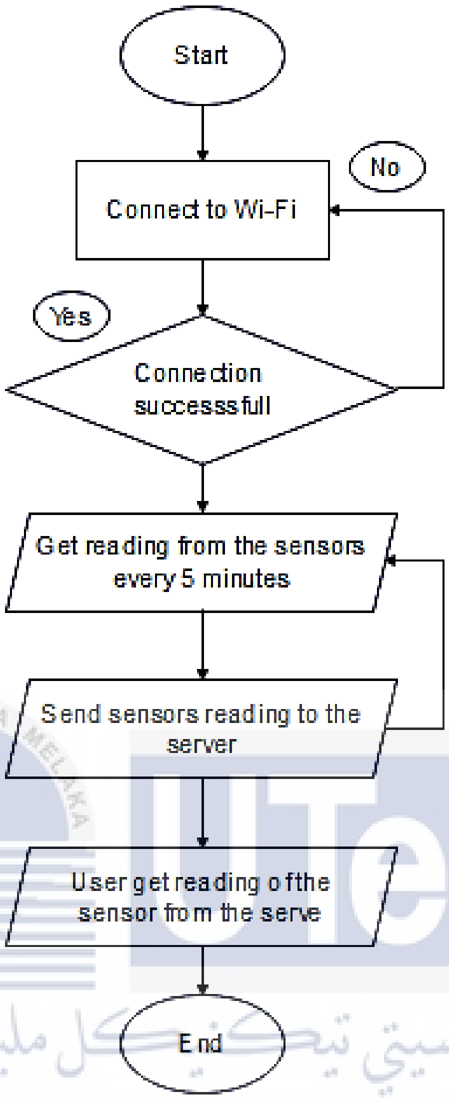


Figure 3.7 Flowchart of the project

3.4 Hardware and Software

The physical component is known as hardware. It is for the Development Of IOT-Based Weather Monitoring And Reporting System Using NodeMCU because the project cannot be completed without hardware and may rely on simulation to demonstrate the process. It is also difficult to maintain when only dependent on software because the user should understand and monitor input and output operations by wiring them to the microcontroller.

3.4.1 NodeMCU ESP 32

The ESP32, which is a single 2.4 GHz Wi-Fi and Bluetooth chip, was created using TSMC's ultra-low-power 40 nm technology. The firmware for the AT command set is pre-programmed into the ESP 32 module. The ESP32 is designed for use in mobile, IoT, and wearable devices. The ESP8266 System on Chip has been replaced with Tensilica's 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth (SoC). It is available in single-core and dual-core configurations. This project will utilize the ESP 32 Wi-Fi board as the microcontroller and gateway, allowing for more network coverage and communication.

Figure 3.8 ESP 32 Microcontroller & Wi-Fi Module

3.4.2 Temperature and Humidity (DHT 22) Sensor

The DHT22 is a simple temperature and humidity sensor with a digital signal. It measure the surrounding air, it utilizes a capacitor-based humidity sensor in connection with a temperature sensor component. The sensor's output is a digital signal that must be transmitted to the relevant information pin. This sensor is accurate and good at measuring a wide range of temperature and humidity parameters.

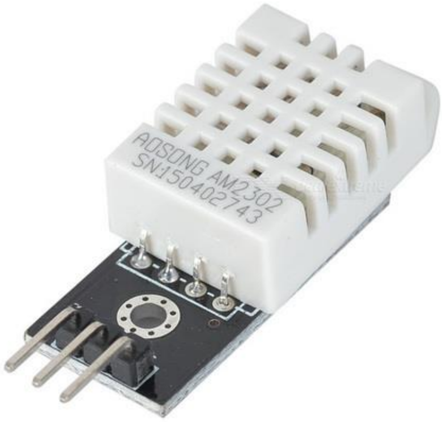


Figure 3.9 Temperature and Humidity Sensor (DHT22)

3.4.3 Rain drop Sensor

The rain drop sensor module is a low-cost rain sensor with an innovative design. There are two parts to it: a rain sensor pad and a control board. The control board reads and sends signals while the sensitive sensor pad detects any water on it. It may be used to monitor the weather and notify the weather conditions when rain is detected. A rain drop sensor is nothing more than a piece of wood with nickel lines painted on it. It is built on the idea of resistance. If there isn't even a speck of rain on board. As a result of the high resistance, we receive a high voltage from V=IR. Because water is an electrical conductor, raindrops reduce resistance, allowing nickel lines to be connected in parallel by decreasing resistance and lowering the voltage drop across them.

Figure 3.10 Rain drop Sensor

3.4.4 Light Intensity Module Sensor

The most common uses for photoresistors, also referred to as light dependent resistors (LDR), are to detect the presence or absence of light or to gauge the intensity of the light. An LDR is used to measure light intensity. An LDR is a part whose resistance (variable) varies according to the amount of light it receives. A light-controlled inconstant resistor is referred to as an LDR. This exhibits photoconductivity because its resistance decreases as the intensity of the incident light increases. The component that measures the value of the lights falling over the photo resistor fitted at the top of the module piece as exposed is known as a photocell sensor or light dependence resistance (LDR). The suggested sensor has four input pins, which are designated by the letters VCC, GND, digital pin D0, and analog pin A0. For instantaneous signaling reactions, the digital pin D0 is in charge of transferring the signal from the sensor to the microcontroller. The suggested connection scheme for the photocell sensor-based Arduino microcontroller is taken into consideration, with the VCC of the sensor being connected to the Arduino's 5V, the sensor's GND being connected to the Arduino's GND, and the sensor's D0 being connected to pin 2 of the Microcontroller.

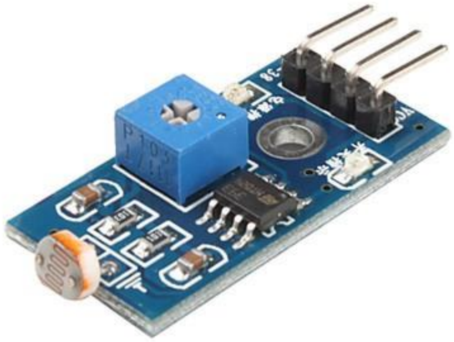


Figure 3.11 Light Intensity Module Sensor

3.4.5 Barometric Pressure Sensor (BMP180)

The I2C bus is intended to be used by the BMP180 to connect it directly to a mobile device's microcontroller. The BMP180's E 2PROM's calibration data must be used to adjust the pressure and temperature data. Piezo-resistive sensor, analog to digital converter, and control unit with E2PROM and serial I2C interface make up the BMP180. The uncompensated value of pressure and temperature is provided by the BMP180. A total of 176 bits of unique calibration data are stored in the E2PROM. This is employed to correct for offset, temperature dependence, and other sensor-related parameters. UP = data on pressure (16 to 19 bit)

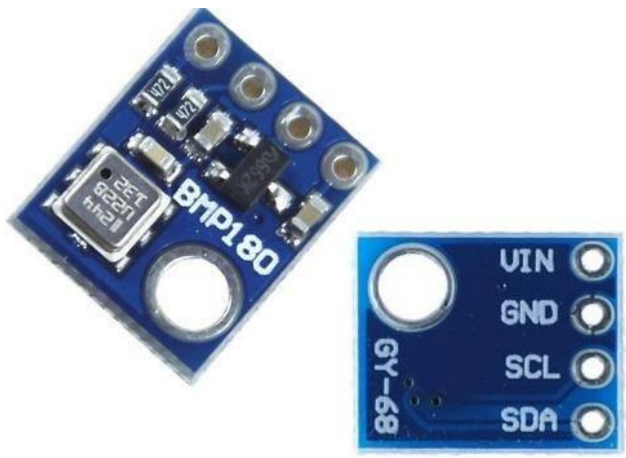


Figure 3.12 Barometric pressure sensor

3.5 Hardware Connection

The connection of hardware used in weather monitoring and reporting systems using NodeMCU ESP32. This project comprises a microcontroller, which serves as the "brain" of the system and directs the behaviour of the other components based on how the system is designed. The input part of this project consists of four sensors, one of which is a rain sensor, each with three pins. The analogue pin is attached to the ESP32's D23 pin, the GND pin is connected to the ground of the supply, and the VCC pin is connected to the supply voltage VIN of the ESP32. The next component is the LDR sensor. The digital pin of the LDR sensor is connected to D18 of the ESP32, the GND pin is attached to the ground of the supply, and the VCC pin is connected to the supply voltage VIN of the ESP32. The DHT22 is the following sensor, and its input pin is attached to the D4 of the ESP32. The GND pin is connected to the supply's ground, and its VCC pin is connected to the VIN pin of the ESP32. The BMP180 sensor is connected to the D22 pin for SCL and D21 pin for SDA on the ESP32, the GND pin is connected to the ground on the supply, and the VCC pin is connected to the ESP8266's supply voltage VIN.

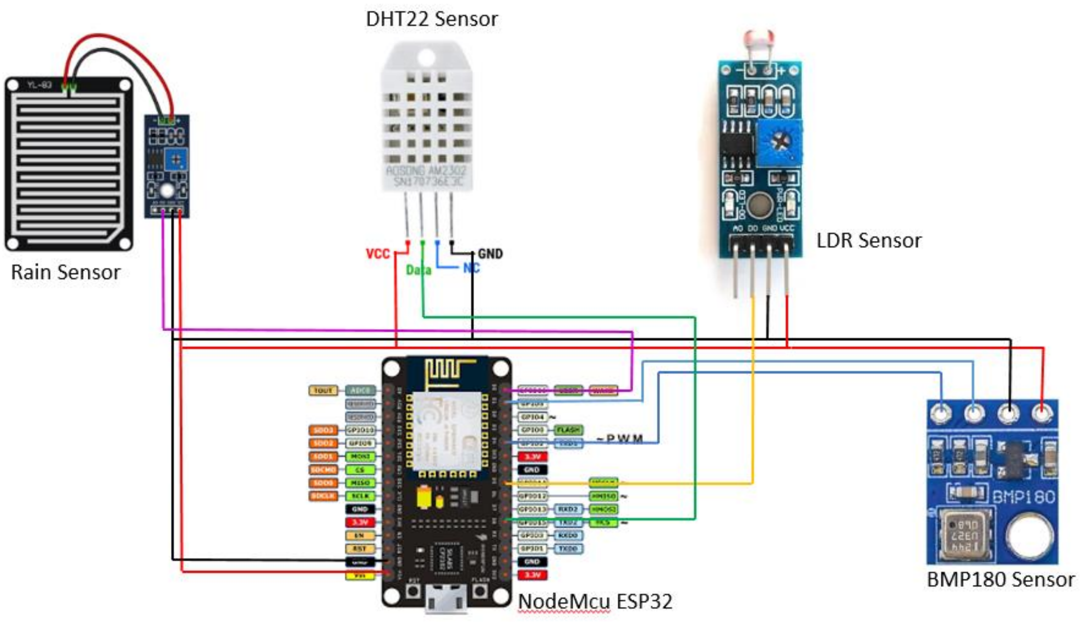


Figure 3.13 Weather monitoring and reporting system connection

3.6 Software Connection

The framework for this project is being developed using software like the Arduino IDE and the Blynk server application. Better programming customization and system simulation are made possible by this software.

3.6.1 Arduino IDE

The IDE (Integrated Development Environment) stands for Integrated Development Environment for Arduino. The Arduino IDE is a free and open-source software that allows you to write and compile code for Arduino modules. The standard Arduino software simplifies code compilation. It is compatible with MAC, Windows, and Linux operating systems and runs on the Java Platform. Built-in commands and techniques for debugging, editing, and compiling code in the environment are included. On the board of each of them is a microcontroller that has been programmed and accepts data in code. As a consequence, we'll use the Arduino IDE to develop code that will control the microcontroller and another device in this project.

For this project, a code was developed in the Arduino IDE and uploaded to the ESP32 to enable Wi-Fi connectivity. In addition, code can be developed using the Arduino IDE to link the Blynk server and ESP32 so that users of Blynk Apps can operate the weather monitoring and reporting system. The software interface for Arduino is shown in figure 3.12 below.

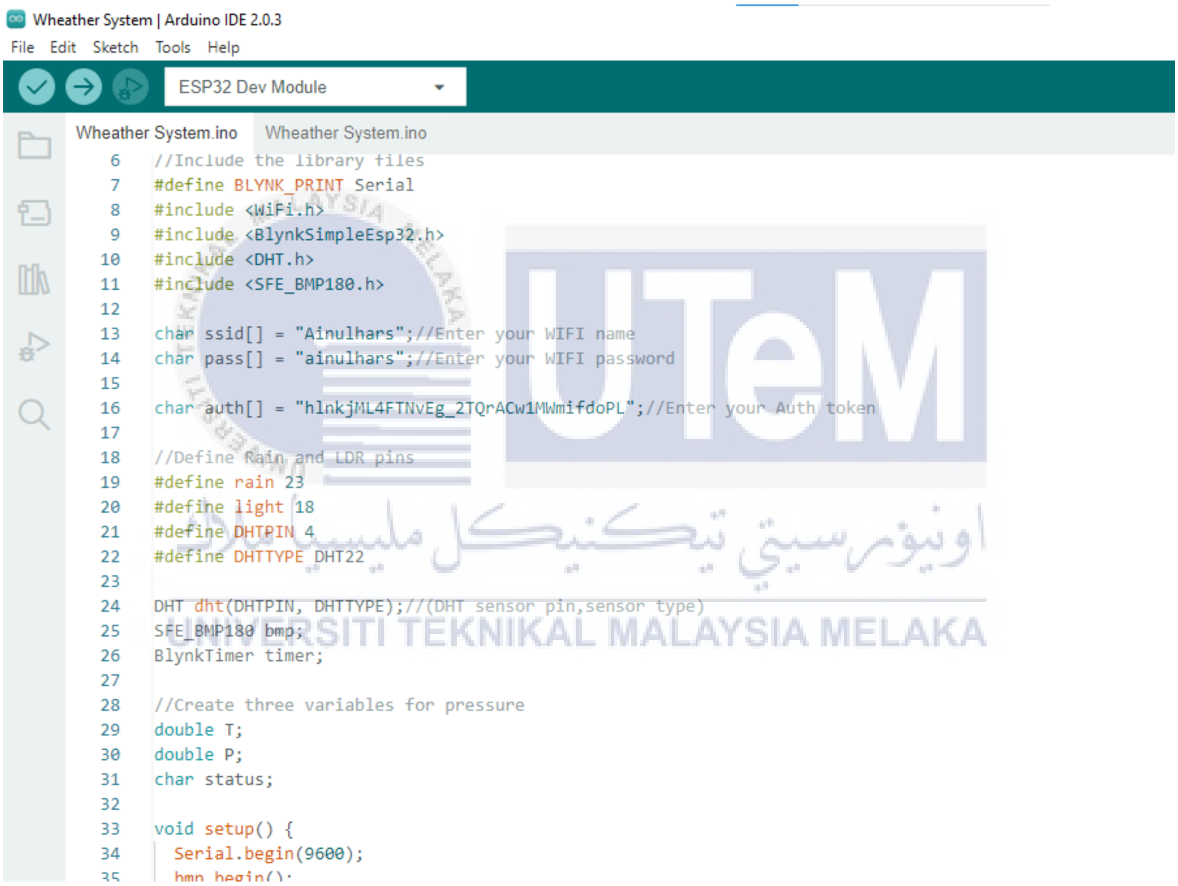


Figure 3.14 Arduino IDE

3.6.2 Blynk Application

Android devices are used to control this weather monitoring and reporting system, which makes it more user-friendly. The microcontroller is configured to accept input signals from Android mobile devices. The Android smartphone and the ESP-32 microcontroller are connected by a Wi-Fi or hotspot, allowing the user to control the device from any Android smartphone within a specific range. The Android application used in this project to manage the weather reporting and monitoring system is called Blynk server. In the following figure 3.13, the application interface is shown.

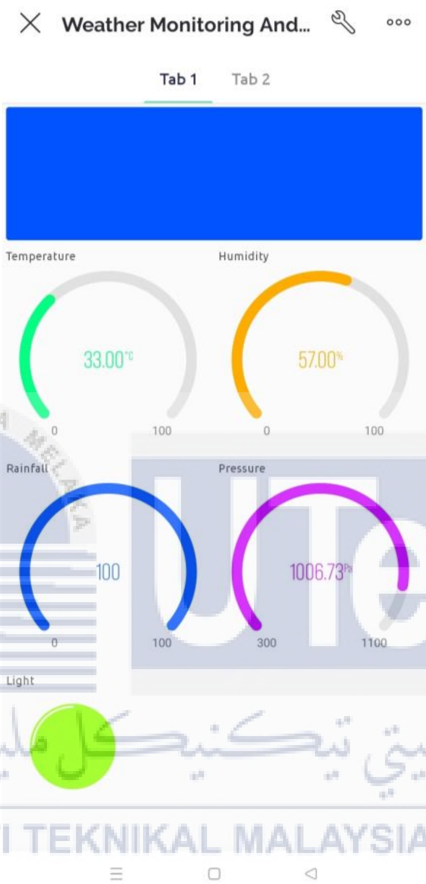


Figure 3.15 Blynk application interface

3.7 Software Implement

This part is focused on the implementation of a Weather monitoring and reporting system using Blynk apps.

3.7.1 Configure nearest IP adress of Blynk Server.

To identify the Blynk server's local IP address at the current location, ping blynk-cloud.com. Webhooks will send a web request to Blynk apps using the IP address of the Blynk server. The closest Blynk Server for the current locations is located at IP address **14.192.208.191**.

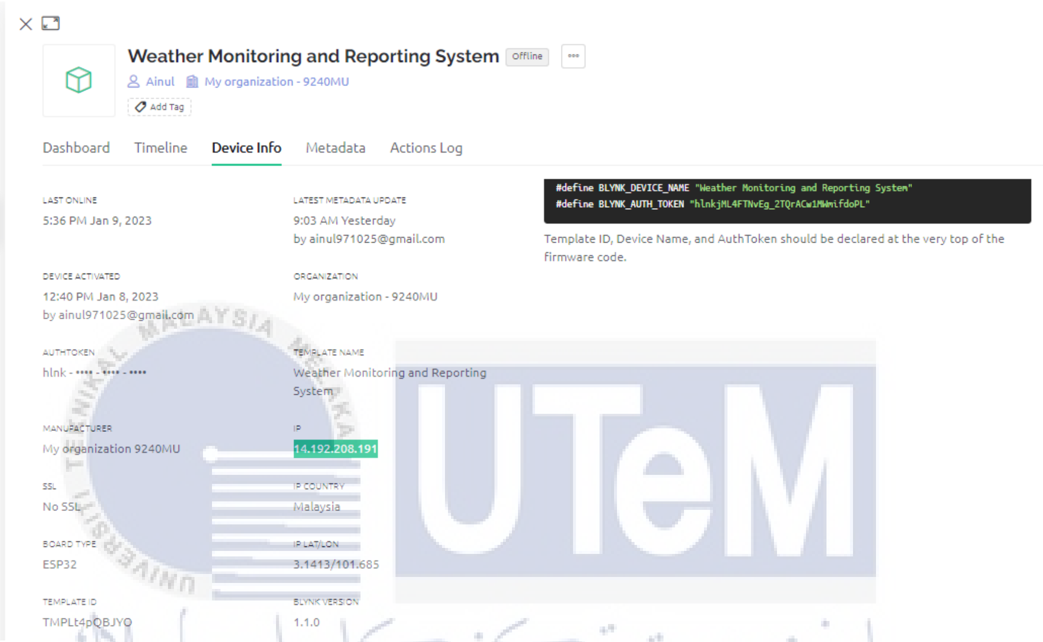


Figure 3.16 IP address of Blynk server

3.8 Summary

This chapter outlines the suggested method for creating a contemporary, successful, and integrated weather station system. The primary purpose of the proposed technique is to arrive at a simple, less rigorous, and effective estimate that does not result in a significant loss of accuracy in the results. The strategy also attempted to make use of network and load data from power providers, both publicly available and limited. The method's ultimate purpose is to achieve efficiency, simplicity of use and manipulation, and the flexibility to deploy on a large-scale distribution network, rather than perfect accuracy.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The results and analysis of the development of a weather monitoring and reporting system are described in this chapter. According to data completion, all simulation test outcomes, operational circumstances, and data analysis are presented. As a guide for determining whether the project's goal has been achieved, the final findings of these reviews and evaluations are used to simulate the project's functionality using the Arduino IDE and Blynk Application

4.2 Project Hardware Prototype

To make it simpler to finalize the design, a rough sketch was first prepared of the draft prototype in figure 4.1 and figure 4.2. In order for the microcontroller and the sensors to operate as planned, it is important to choose their position. To make it simpler to open and close in order to view the interior of the circuit box.

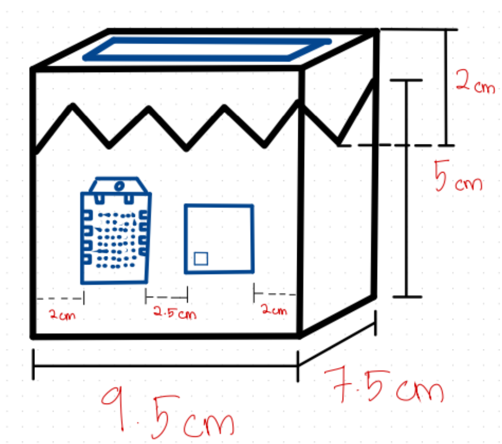


Figure 4.1 Front View

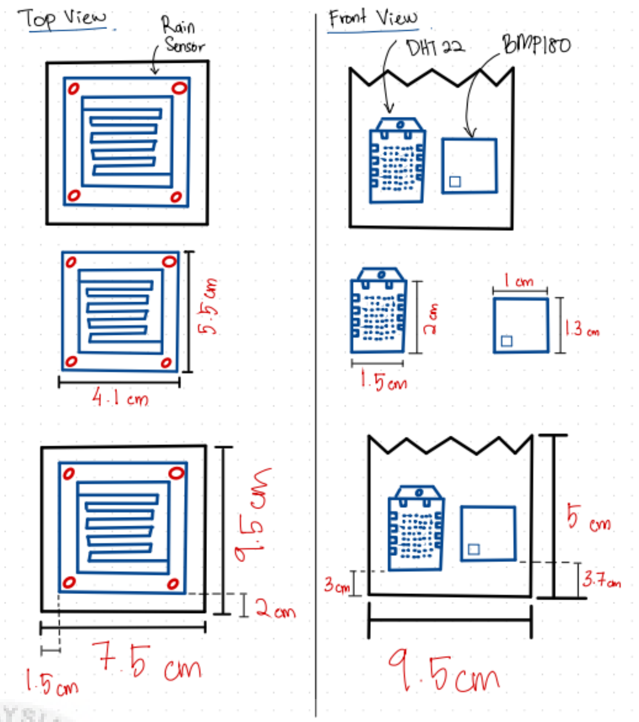


Figure 4.2 Top view and right view

The Weather reporting and monitoring system prototype. The top view is shown in the first image, and the right side view is shown in the second image. Figure 4.3 and figure 4.2 shown the real hardware prototype.

Figure 4.3 Top view of the project

Figure 4.4 Right side view of the project

4.3 Project Integration

4.3.1 Connection of NodeMcu ESP32

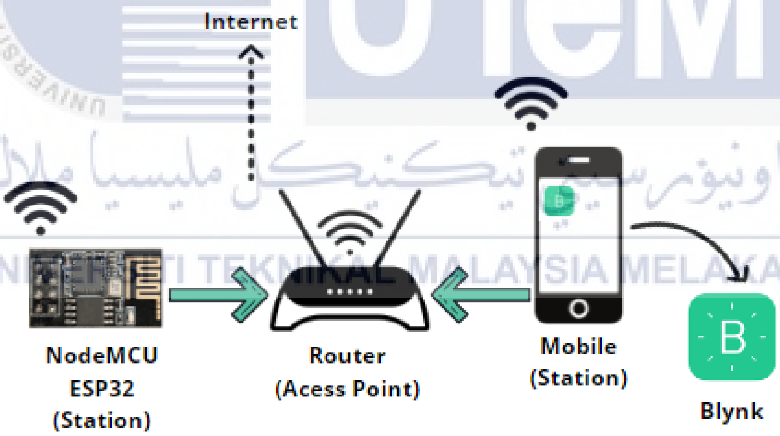


Figure 4.5 Block diagram of internet connection of ESP32

Figure 4.5 demonstrates how the NodeMCU ESP32 is connected to the internet in order to communicate with the Blynk App on the user's mobile device. The microcontroller, NodeMCU ESP32, will read all the data from the input sensors and then transmit it to the user via IoT.

4.3.2 Login to Blynk application

User need to log into their Blynk account with their email and password for security purposes as shown in figure 4.6.

Figure 4.6 Login page on Blynk application

4.3.3 Blynk LCD display

The project's rain sensor's function is shown. When the rain sensor collects a raindrop larger than 0.1 milliliters, the LCD display shows this to the user. This LCD display can be shown to the user's mobile device, informing them whether or not it is raining outside. Based on figure 4.7 and figure 4.8.

Figure 4.7 LCD display when no rain drop

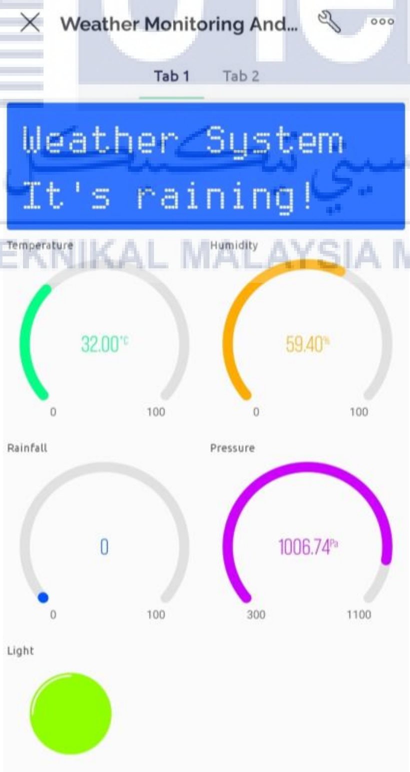


Figure 4.8 LCD display when detect rain drop

4.3.4 Chart

Figure 4.9 demonstrates that the user can reporting the weather by this chart. The user can choose the specific range time to monitor and reporting the weather based on the time. This chart contain for 4 data which is temperature, humidity, pressure, and rain drop.

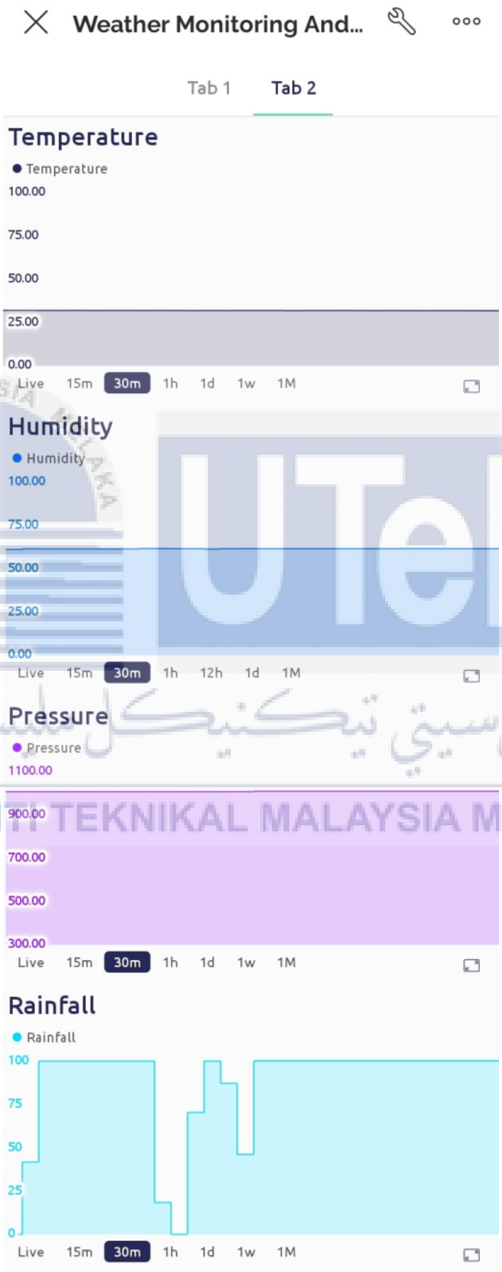


Figure 4.9 Chart display

4.4 Data Analysis

4.4.1 The value of rain based on the volume of raindrops

Using a syringe, water is sprayed onto a rain sensor to simulate rain in order to test it. The experiment involved the number of water droplets used. The sensor can be considered to be accurate if it informs the user when there are water droplets present and informs them when there aren't.

Table 4.1 Rain sensor volume test result

Volume (ml)	Rain Value	LCD Display
0	100	Have a nice day!
0.10	0	It’s Raining
0.20	0	It’s Raining
0.30	0	It’s Raining
0.40	0	It’s Raining
0.50	0	It’s Raining
0.60	0	It’s Raining
0.70	0	It’s Raining
0.80	0	It’s Raining
0.90	0	It’s Raining
1.0	0	It’s Raining

4.4.2 The accuracy of temperature and humidity sensor by comparing the data value with weather forecast and measured data in different cities

In this experiment, to measure the temperature and humidity, the sensors are located at the same spot and period on five or more days. To know that the sensors are accurate and functioning, we need to make sure that the percentage error is low. To see the range of outcomes over the course of a week, data are collected from two different cities.

Table 4.2 Temperature and Humidity test results in Ayer Keroh, Melaka

Day	**Weather component**	**Weather forecast**	**Measured data**	**Percentage Error**
1	Temperature	28°c	27.31°c	2.46%
	Humidity	89%	88.6%	0.45%
2	Temperature	30°c	29.45°c	1.83%
	Humidity	84%	82.44%	1.86%
3	Temperature	30°c	29.55°c	1.5%
	Humidity	84%	83.56%	0.52%
4	Temperature	30°c	29.32°c	2.27%
	Humidity	83%	78.52%	5.39%
5	Temperature	30°c	30.46°c	1.53%
	Humidity	83 %	79.73%	3.94%

Table 4.3 Temperature and Humidity test results in Durian Tunggal, Melaka

Day	**Weather component**	**Weather forecast**	**Measured data**	**Percentage Error**
1	Temperature	30°c	28.90°c	3.67%
	Humidity	74%	73.26%	1%
2	Temperature	31°c	30.15°c	2.74%
	Humidity	68%	65.50%	3.68%
3	Temperature	32°c	31.70°c	0.94%
	Humidity	67%	64.50%	3.73%
4	Temperature	30°c	28.78°c	4.07%
	Humidity	75%	71.80%	4.27%
5	Temperature	31°c	30.50°c	1.61%
	Humidity	73 %	69.60%	4.66%

4.4.3 The accuracy of pressure sensor by comparing the data value with weather forecast and measured data in different cities

In this experiment, to measure pressure sensors are located at the same spot and period on five or more days. To know that the sensors are accurate and functioning, we need to make sure that the percentage error is low. To see the range of outcomes over the course of a week, data are collected from two different cities.

Table 4.4 Pressure test results in Ayer Keroh, Melaka

Day	Weather forecast	Measured data	Percentage Error
1	1011 Pa	1009 Pa	0.2%
2	1010 Pa	1010 Pa	0%
3	1009 Pa	1008 Pa	0.1%
4	1009 Pa	1007 Pa	0.2%
5	1010 Pa	1008 Pa	0.2%

Table 4.5 Pressure test results in Durian Tunggal, Melaka

Day	Weather forecast	Measured data	Percentage Error
1	1010 Pa	1009 Pa	0.1%
2	1011 Pa	1009 Pa	0.2%
3	1010 Pa	1009 Pa	0.1%
4	1010 Pa	1008 Pa	0.2%
5	1009 Pa	1008 Pa	0.1%

4.4.4 The distance of light from the LDR sensor

A flashlight is used to examine the light sensor during testing. To achieve different exposure levels, the flashlight is moved closer or further from the LDR sensor.

Table 4.6 LDR sensor test result

Distance (cm)	LED Indicator
1	ON
3	ON
5	ON
10	ON
20	OFF
30	OFF
40	OFF

4.5 Summary

In conclusion, during this project's successful completion, several web dashboards were suitable for use. Among them are web dashboards for Arduino cloud IoT, ThinkSpeak, and Cayenne. Each web dashboard has its advantages and disadvantages.

Arduino cloud IOT web dashboard is straightforward because each widget connected to the datastream has its own code, and there is no need to create new code in the Arduino IDE software. Widget selection is also relatively easy, and there are many options according to the modern style of the current era. But it has a weakness where only five data streams can be used while the sensor produces more than five reading values. So had to find another web dashboard.

In addition to the ThinkSpeak Web Dashboard, many tutorials can be found on the website and are easy to follow. But it needs to be expanded using widgets and sensor data streams.

After that cayenne web dashboard it has just introduced, and only a few people are familiar with this web dashboard. Cayenne is straightforward to use for notification extensions. No need to change the original coding in Arduino IDE; add the coding in the Cayenne application only. The problem is that not all sensor values can be displayed. In this issue, the pressure value cannot be displayed, and most of them have the same problem. Even after changing the code in Arduino IDE, it still can't help. Without hesitation, switch to the Blynk web dashboard.

Lastly, the Blynk web dashboard still has some limitations. Throughout learning to use this Blynk application, many widgets have been used to find the appropriate dashboard. Even though it has limitations, the Blynk application is still stable for the weather monitoring and reporting system.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the design of the weather monitoring and reporting system works as expected and can perform better when designed into an actual product using the right components for each part. A weather monitoring and reporting system have been developed to make it easier for users to monitor and check current uncertain weather conditions. For user convenience, a mechanism or tool for the Blynk application monitor has been used as an application for monitoring the status of temperature, humidity, air pressure, light, and rain.

This project was created to make it easier for users to access information instantly and anywhere. The user has access to the sensors' measurements at any time. As a result, users always have access to current weather information whenever they need it. The user is then given real-time warnings to organize their day accordingly. Understanding these signals can help prevent emergencies.

According to current trends, the components are also cheap, simple, and portable. This is simple but still keeps the project's fundamental goal of analyzing the weather station—a cost-effective wireless weather station with IOT-based graphical application software for quick weather information checks. Temperature, pressure, humidity, rainfall, and light can all be calculated. Comparing the results to those from other cost-effective weather stations, they showed good accuracy and stability. It has a meager maintenance cost and an intuitive interface.

5.2 Future Works

For future work, the results of this project could be enhanced as future updates could include additional enhancements like wind direction, solar radiation, and precipitation. Can create more notification alert for each sensor. It may also be used in hospitals or other medical facilities for research and studies on the Effect of Weather on Health and Diseases, improving warnings for precautions. Additionally, this weather monitoring system may be utilized for many other automation projects, such as smart city projects.

REFERENCES

[1] Faqihah Binti Adnan, “Weather Station Using Iot Fatin Nur Faqihah Binti Adnan (Software Development) With Honours Faculty of Informatics and Computiing,” 2019.

[2] S. Banara, T. Singh, and A. Chauhan, “IoT Based Weather Monitoring System for Smart Cities: A Comprehensive Review,” *2022 Int. Conf. Adv. Technol. ICONAT 2022*, pp. 1–9, 2022, doi: 10.1109/ICONAT53423.2022.9726106.

[3] B. S. Rao, K. S. Rao, and N. Ome, “Internet of Things (IOT) Based Weather Monitoring system,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, no. 9, pp. 312–319, 2016, doi: 10.17148/IJARCCE.2016.5966.

[4] E. P. Uagbae, V. Eshiet, and U. Godwin, “Arduino-Based Weather Monitoring System Arduino-Based Weather Monitoring System,” *m. Int. J. Eng. Res. Gen. Sci.*, no. February, pp. 835–846, 2018, [Online]. Available: https://www.researchgate.net/publication/330220305

[5] M. Kusriyanto and A. A. Putra, “Weather Station Design Using IoT Platform Based On Arduino Mega,” *ISESD 2018 - Int. Symp. Electron. Smart Devices Smart Devices Big Data Anal. Mach. Learn.*, pp. 8–11, 2019, doi: 10.1109/ISESD.2018.8605456.

[6] V. Șveț and V. Gurduza, “IOT BASED WHEATHER REPORTING SYSTEM,” pp. 1–3, 2020.

[7] S. Meera, R. Sharmikha Sree, R. A. Kalpana, S. R. Manasvinii, V. Haritha, and K. Valarmathi, “IOT based weather reporting system using Arduino and Node MCU,” *Adv. Parallel Comput.*, vol. 38, pp. 209–213, 2021, doi: 10.3233/APC210038.

[8] K. S. Sai Ram and A. N. P. . Gupta, “IoT based Data Logger System for weather monitoring using Wireless sensor networks,” *Int. J. Eng. Trends Technol.*, vol. 32,

no. 2, pp. 71–75, 2016, doi: 10.14445/22315381/ijett-v32p213.

[9] P. S. B. Kamble, P. R. P. Rao, A. S. Pingalkar, and G. S. Chayal, “IoT Based Weather Monitoring System,” *SSRN Electron. J.*, no. 2, pp. 2886–2891, 2017, doi: 10.2139/ssrn.3918133.

[10] Y. Rahut, R. Afreen, and D. Kamini, “Smart weather monitoring and real time alert system using IoT,” *Int. Res. J. Eng. Technol.*, vol. 848, pp. 848–854, 2008, [Online]. Available: www.irjet.net

[11] F. J. J. Joseph, “IoT based weather monitoring system for effective analytics,” *Int. J. Eng. Adv. Technol.*, vol. 8, no. 4, pp. 311–315, 2019.

[12] A. M. Bhagat, A. G. Thakare, and K. A. M. | N. S. M. | P. V. Choudhary, “IOT Based Weather Monitoring and Reporting System Project,” *Int. J. Trend Sci. Res. Dev.*, vol. Volume-3, no. Issue-3, pp. 365–367, 2019, doi: 10.31142/ijtsrd21677.

[13] P. Yadav, N. Nigam, and R. Yadav, “IoT based- Advanced Weather Monitoring System IoT based- Advanced Weather Monitoring System,” 2020.

[14] M. Tiwari and D. Narang, “WEATHER MONITORING SYSTEM USING IOT AND CLOUD COMPUTING,” *Lect. Notes Electr. Eng.*, vol. 788, no. June, pp. 247–253, 2020, doi: 10.1007/978-981-16-4149-7_21.

[15] A. J. PabiD, R. ReddyN, M. YusufS, and K. KumarD, “Weather Reporting System using Internet of Things,” *Int. Res. J. Eng. Technol.*, no. July, pp. 2135–2138, 2021, [Online]. Available: www.irjet.net

[16] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. El Hajjaji, “IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts,” *Big Data Min. Anal.*, vol. 4, no. 1, pp. 25–32, 2021, doi: 10.26599/BDMA.2020.9020018.

[17] V. Gotmare, R. Kolte, and R. Thengodkar, “WEATHER MONITORING SYSTEM USING ARDUINO UNO,” no. June, 2021, doi: 10.17605/OSF.IO/R8XWP.

[18] T. E. Babalola, A. D. Babalola, and M. S. Olokun, *Development of an ESP-32 Microcontroller Based Weather Reporting Device*, no. July. 2022, pp. 27–38. doi: 10.9734/jerr/2022/v22i1117577.

[19] N. Kumari, Sakshi, S. Gosavi, and S. S. Nagre, “Real-Time Cloud based Weather Monitoring System,” *2nd Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2020 - Conf. Proc.*, no. Icimia, pp. 25–29, 2020, doi: 10.1109/ICIMIA48430.2020.9074848.

[20] A. S. Bin Shahadat, S. I. Ayon, and M. R. Khatun, “Efficient IoT based Weather Station,” *Proc. 2020 IEEE Int. Women Eng. Conf. Electr. Comput. Eng. WIECON-ECE 2020*, pp. 227–230, 2020, doi: 10.1109/WIECON-ECE52138.2020.9398041.

APPENDICES

Appendix A (Coding Arduino)

```
/*Weather monitoring and reporting system with the New Blynk app
   Home Page
*/

//Include the library files
#define BLYNK_PRINT Serial // Libary Blynk
#include <BlynkSimpleEsp32.h> // Libary Blynk
#include <WiFi.h> // Libary ESP
#include <DHT.h> // Libary DHT22
#include <SFE_BMP180.h> // Libary bmp180

/*Put SSID & Password*/
char ssid[] = "Ainulhars";//Enter your WIFI name
char pass[] = "ainulhars";//Enter your WIFI password
char auth[] = "hlnkjML4FTNvEg_2TQrACw1MWmifdoPL";//Enter your Auth token

//Define Rain, LDR & DHT pins
#define rain 23
#define light 18
#define DHTPIN 4
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);//(DHT sensor pin,sensor type)
SFE_BMP180 bmp;
BlynkTimer timer;

//Create three variables for pressure
double T;
double P;
char status;

//-------------------------Setup------------------------
void setup() {
  Serial.begin(9600); // The serial monitor is beginning
  bmp.begin();
  pinMode(light, INPUT);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  Blynk.virtualWrite(V5,"Weather System");
  Blynk.virtualWrite(V6,"");
  dht.begin();
```

```
  //Call the functions
  timer.setInterval(100L, DHT22sensor);
  timer.setInterval(100L, rainSensor);
  timer.setInterval(100L, pressure);
  timer.setInterval(100L, LDRsensor);
}

//-------------------------DHT22--------------------------
void DHT22sensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);
}

 //-------------------------Rain Sensor-----------------------
  bool alert = false;     // Flag to track if an alert has been sent for
rain event
  float threshold = 100;  //Send an alert if rain is less than this value

void rainSensor()
{
  int value = digitalRead(rain);
  value = map(value, 0, 1, 0, 100);
  Blynk.virtualWrite(V2, value);

  if (value < threshold && alert == false)
  {
    Blynk.virtualWrite(V6,"It's raining!");
    Blynk.logEvent("rainfall","It's raining!");
    alert = true;
  }
  else
  if(value > threshold)
  {
    alert = false;
  }
  Blynk.virtualWrite(V6,"Have a nice day!");
}
```

```
  //--------------------------BMP180------------------------
void pressure()
{
  status =  bmp.startTemperature();
  if (status != 0)
  {
    delay(status);
    status = bmp.getTemperature(T);

    status = bmp.startPressure(3);// 0 to 3
    if (status != 0)
    {
      delay(status);
      status = bmp.getPressure(P, T);
      if (status != 0)
      {
      }
    }
  }
  Blynk.virtualWrite(V3, P);
}

 //---------------------------------LDR Sensor---------------------------
void LDRsensor()
{
  bool value = digitalRead(light);
  if (value == 0)
  {
    WidgetLED LED(V4);
    LED.on();
  }
  else
  {
    WidgetLED LED(V4);
    LED.off();
  }
}

void loop()
{
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
```