Faculty of Electronics and Computer Technology and Engineering

DEVELOPMENT OF AUTOMATED WEATHER MONITORING SYSTEM USING LoRa

NURAFIQAH BINTI MUKHTAR

Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

2024

DEVELOPMENT OF AUTOMATED WEATHER MONITORING SYSTEM USING LoRa

NURAFIQAH BINTI MUKHTAR

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek: Development of Automated Weather Monitoring System Using LoRa

Sesi Pengajian: 2023/2024

Saya NURAFIQAH BINTI MUKHTAR mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☒ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)

Alamat Tetap:

NO.41 TAMAN CHENERAS JAYA,

27200, KUALA LIPIS, PAHANG

(COP DAN TANDATANGAN PENYELIA)

MOHD FAIZAL BIN ZULKIFLI
Jurutera pengajar
Jabatan Teknologi Kejuruteraan Elektronik & Komputer
Fakulti Teknologi Kejuruteraan Elektrik & Elektronik
Universiti Teknikal Malaysia Melaka

Tarikh: 12.1.2024

Tarikh: 22.2.2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development of Automated Weather Monitoring System Using LoRa" 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 : Nurafiqah Binti Mukhtar

Date : 12 January 2024

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 with Honours.

Signature :

Supervisor Name : Ts. Mohd Faizal Bin Zulkifli

Date : 22/2/2024

DEDICATION

I would like to thank my family and friends for supporting me throughout this process. I would like to express my deep appreciation to my family, for their unwavering support and encouragement. My sister, Nurul Nabilah and brothers, Muhamad Nabil & Muhamad Afiq have always been there for me and are truly remarkable. I would also like to thank my many friends & my surrounding for their continued support throughout this process.

I would also like to express my gratitude to my many sponsors who have helped me along the way.

Ts. Mohd Faizal Bin Zulkifli is my supervisor who has been instrumental in developing my technology skills and coding programming.

And my amazing friends, who has supported me throughout this entire this program.

They have been my biggest cheerleaders.

ABSTRACT

This project focuses on the creation of an Automated Weather Monitoring System using LoRa (Long Range) technology. Conventional weather monitoring systems sometimes need extensive infrastructure and substantial human engagement, making them unsuitable for remote areas.A wireless sensor network is used in this project, which includes weather sensors such as temperature, humidity, rainfall, wind speed, rain volume detector level, and noise sensors. These sensors capture real-time weather data and send it wirelessly through the LoRa communication protocol. LoRa technology was chosen for its long-range capabilities, low power consumption, and ability to perform in low signal-to-noise settings.A LoRa gateway installed in a central monitoring station serves as the system's brain. Weather stations receive weather data sent by sensors and send to a server for data storage and analysis and a user-friendly interface is developed to display weather data, allowing users to access real-time information and historical records.

ABSTRAK

Projek ini memberi tumpuan kepada penciptaan Sistem Pemantauan Cuaca Automatik menggunakan teknologi LoRa (Jarak Jauh). Sistem pemantauan cuaca konvensional kadangkala memerlukan infrastruktur yang luas dan penglibatan manusia yang banyak, menjadikannya tidak sesuai untuk kawasan terpencil. Rangkaian penderia wayarles digunakan dalam projek ini, yang merangkumi penderia cuaca seperti suhu, kelembapan, hujan, kelajuan angin, tahap pengesan volum hujan, dan sensor bunyi. Penderia ini menangkap data cuaca masa nyata dan menghantarnya secara wayarles melalui protokol komunikasi LoRa. Teknologi LoRa dipilih untuk keupayaan jarak jauhnya, penggunaan kuasa yang rendah dan keupayaan untuk berprestasi dalam tetapan isyarat-ke-bunyi yang rendah. Gerbang LoRa yang dipasang di stesen pemantauan pusat berfungsi sebagai otak sistem. Stesen cuaca menerima data cuaca yang dihantar oleh penderia dan dihantar ke pelayan untuk penyimpanan dan analisis data dan antara muka mesra pengguna dibangunkan untuk memaparkan data cuaca, membolehkan pengguna mengakses maklumat masa nyata dan rekod sejarah.

ACKNOWLEDGEMENTS

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

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and my siblings for the financial support which enables me to accomplish the project.

My highest appreciation goes to my parents and family members for their love and prayer during the period of my study. An honourable mention also goes to me for all the motivation and understanding. And to my friends around me, thanks for always be there for me to complete my project.

Finally, I would like to thank all the classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Ω	-	Ohm
$^{\circ}C$	-	Degree Celcius
%	-	Percentage

LIST OF ABBREVIATIONS

V	-	Voltage
dBm	-	Decibal miliwatts
Hz	-	Hertz
mA	-	MiliAmpere
uA	-	MicroAmperes
hpa	-	Hectopascal

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

According to an article from Kosmo written by (Muhammad Farid, 2021)flash floods that occur in urban areas are not a new problem that we are facing nowadays. Many incidents of flash floods happen without any warning causing the community to bear the loss when they cannot save their property because it happens so quickly.

LoRa technology has the advantage of monitoring the weather over long distances and various communications. With this feature it enables the collection of data from remote locations that may be impossible to access and other advantages of LoRa are such as low power consumption, long battery life and low maintenance costs. Real-time data from sensors makes predictions and warnings accurate and timely.

Therefore, LoRa is a technology used in automatic weather monitoring systems because it is a cost-effective and efficient way to monitor the weather. The device can be used over long distances, uses little power, and collects data in real-time.

1.2 Problem Statement

With climate change happening in the world now is one of the problems for people who work without shelter outside. settled climate science predicts extreme weather events will become more severe, more frequent, and cover larger and larger areas. This project uses low-cost components compared to the previous journal that uses Raspberry pi which is quite expensive, and the sensor determines whether it is raining or not, but it cannot detect the level of water in some places.

The effect on users that there is no early warning of the increase in rainwater at the minimum or maximum level for urban and rural areas which causes flooding problems that occur. The problem of flooding for users is not a small problem, especially for rural residents who are close to the river because they need to be ready to face flooding in the event of continuous rain. With that the problem of seeing the water level at the minimum or maximum level is important to avoid being late from being ready to move.

Currently available weather monitoring systems are subject to some limitations in terms of range, especially previously created projects that use IoT technology because the range cannot exceed the limits determined by Wi-Fi itself. Because of this, it will be difficult to expand the system's coverage or handle higher data requests in the future. Also depending on the weather monitoring system, it may be difficult to expand coverage or handle larger data volumes over larger areas.

The problem statement that has been mentioned above will be overcome with a project that will be built using LoRa to monitor the weather with this system automatically sending notifications to users.

1.3 Project Objective

Based on the problem statements discussed above, the objectives of this study are:

I. To study and propose the method and the component used of weather monitoring system.

II. To develop a Lora-based system to monitor the weather.

III. To analyze all data of parameters Using LoRa.

1.4 Scope of Project

This project focuses on weather readings that will be monitored through this system and use LoRa (Long-Range) technology in developing an automatic weather monitoring system. It offers cost-effective, wide-area coverage with real-time data collection and analysis capabilities. A barometric pressure sensor predicts rain based on its effect on air pressure and raindrops are used to detect whether or not it is raining.

1.5 Expected Result

The expected result of this project is Real-Time Monitoring, which means the system will monitor weather conditions continuously and in real time. The LoRa technology used in the system provides scalability and enables the use of multiple monitoring stations across large areas and it is scalable, so it can be expanded to cover larger areas or add additional sensors to collect comprehensive data by providing instant updates and warnings in significant change events or adverse weather events to users.

1.6 Organization

Chapter One discusses the problem statement, objectives, and scope of the project and provides a brief overview of how it will be carried out. reviewing the literature about existing methods and technologies from previous projects has been told or written in Chapter Two and the difference in parameters has been stated in this chapter. In continuing this work in Chapter Three, details about the components and methods that will be used during the completion of this project will be presented, as well as the type of approach that we will use and, in this chapter, and will also get a clear picture of how the project will be managed from beginning to end, so that it can plan accordingly.

1.7 Summary

The purpose of this chapter is to give an overview of the project. Project objectives, problem statement, and scope are outlined in the background project. State the problems in this project that can be improved and the purpose to improve them in the objective. Technology used in this project and the expected results are the focus of the project scope.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, Literature reviews summarize and analyze existing research and literature on a specific topic. This report gives a brief overview of current knowledge, identifies gaps and areas requiring further research, and highlights key findings and conclusions. The purpose of a literature review is to provide a comprehensive understanding of the topic under research and to synthesize the existing literature. It also assists in establishing the context and background of the research question.

There are several steps that are taken while conducting a literature review, which include systematic searches and reviews, followed by critical evaluation and synthesis to establish the current state of knowledge in an area. Even though previous research had identified inconsistencies and gaps, future research needs to address these issues.

In summary, a literature review lends support to researching questions, hypotheses, and methodology, and provides a framework for analyzing and interpreting the results.

2.2 Internet of Things (IoT) based Weather Monitoring System

The project was conducted by(Girija C, n.d.). Presented in this paper is a new method for displaying weather information anywhere in the world based on the research conducted for the title. The Internet of Things (IoT) is a technology that creates a network of connected things. It connects the world of things. Electronic items include cameras, sensors, and automotive electronics. Environmental conditions such as temperature, relative humidity, and CO level are monitored and controlled by sensors. The implemented system plots graphs based on sensor data and allowed access from anywhere on the planet through the internet.

Among the latest financial and technological waves in the worldwide data industry, the internet of things (IoT) is viewed as a potential financial and technological wave. It is a smart system that interconnects all things to the Internet to exchange data and convey through data-detecting gadgets. Sensors are placed at different locations in an area of interest to predict its behavior.

This paper will discuss the design and implementation of a remote monitoring system that captures the sensor data in the cloud and stores it remotely. We will be able to view the estimated trend remotely through a web browser. This paper proposes that wireless embedded computing systems can be used to monitor the environment for temperature, humidity, and carbon dioxide levels. As an example, CO levels exceeding normal levels in the air are a parameter crossing its threshold values. By integrating sensor devices and wireless communication, embedded systems allow users to access various parameters remotely and store the data in the cloud.

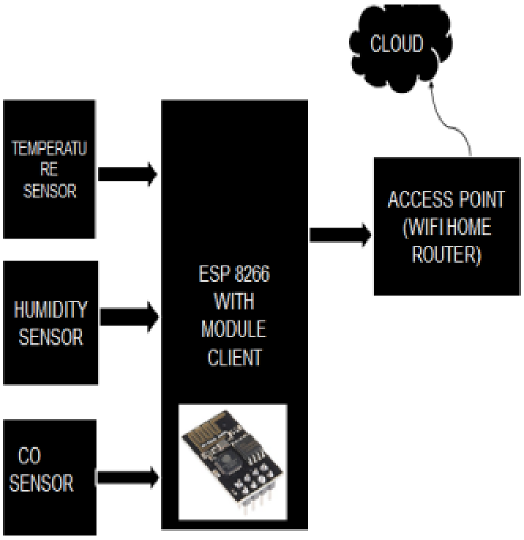


Figure 2.1: Block Diagram of Internet of Thing (IoT) based Weather Monitoring System

The component used is the pre-programmed system-on-chip known as ESP8266 interacts with micro-controllers through the UART interface. The DHT 11 temperature and humidity sensor and the MQ 6 CO sensor are its two sensors. The microprocessor transforms the analogue voltage from the sensors into digital data, and the sensor readings might be as much as two seconds old. An open-source Internet of Things (IOT) programmed called Thing Speak can build a social network of objects with applications for sensor recording, location tracking, and status updates.

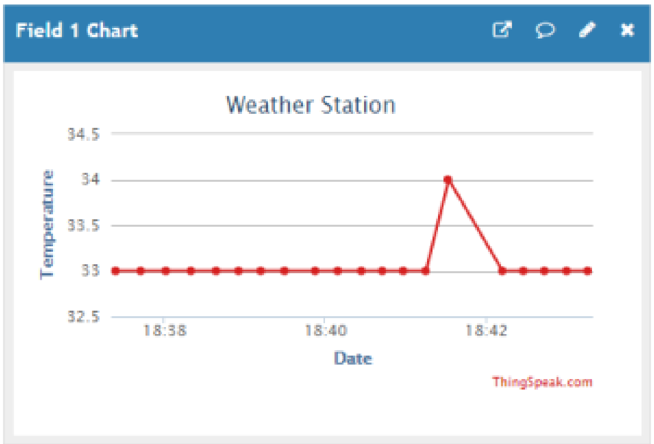


Figure 2.2: Result of Temperature Simulation Vs Time

Simulation results for this project include data collected from sensors placed in targeted areas. A connection will be established with the server device and the sensed data will be automatically sent to the web server. Monitoring and controlling the system will be possible through the web server page. By displaying temperature, humidity, and CO levels in a specific region, the embedded monitoring system displays data. The sensed data will be stored in cloud storage (Google Spread Sheets). Cloud data can be analyzed and continuously monitored for a variety of purposes. Air temperatures, humidity, and CO levels are measured periodically.

There are some circuits that can include an alarm to notify the user if there is an excessive smoke condition, that is, smoke detectors are proposed for future projects. Temperature, humidity and smoke parameters can be sent via SMS to customers, informing them of the current situation.

2.3 Development of IoT Weather Monitoring System

In Development of IoT Weather Monitoring System, (Rusyaidi et al., 2022) conducted this project. The weather and climate change are key factors determining society's development. Essentially, the weather describes the state of the atmosphere, whether it's hot or cold, dry or wet, clear skies or cloudy. An average atmospheric condition can be defined as climate if it is averaging over an extended period.

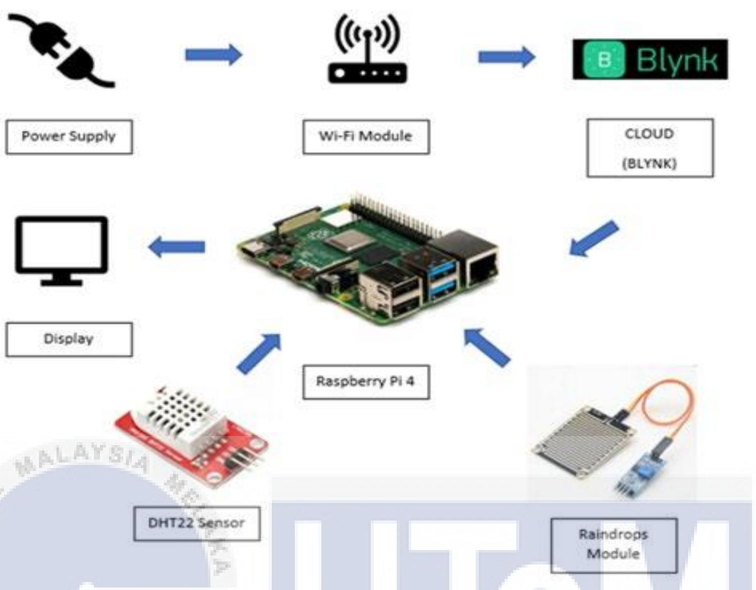


Figure 2.3: Block Diagram of Development of IoT Weather Monitoring System

IoT enabled weather monitoring system based on Raspberry Pi contains power supply, Wi-Fi module, Cloud Database (Blynk), Raindrop Module, DHT22 Temperature and Humidity Sensor, and LCD Display. It will automatically start looking for a Wi-Fi connection after being activated. Either Wi-Fi, a hotspot, or Ethernet are among the options that can be chosen. The Blynk app starts pairing with the Raspberry Pi and begins sending data once it is connected. Data will be collected on rainy conditions from the raindrop module, whereas temperature and humidity data will be collected from the DHT22 sensor. The Raspberry Pi and Blynk will then have access to these data, which can then be displayed on the LCD or mobile device through a graphical user interface.

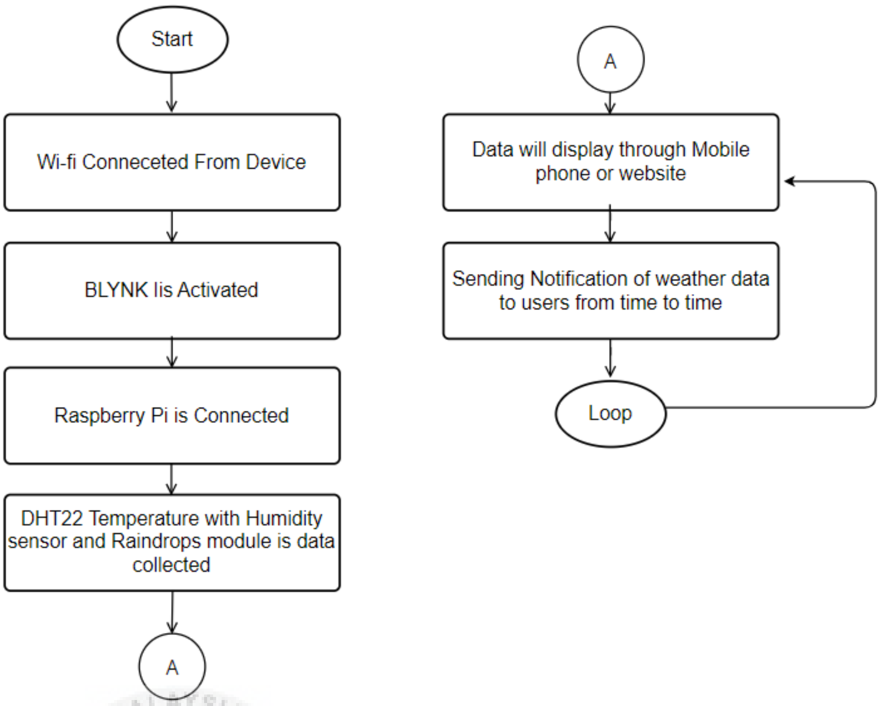


Figure 2.4: Flowchart of IoT Weather Monitoring System

According to the flowchart above, when the device is turned on, it searches for a network connection automatically. Blynk needs to be activated and paired with the same network connection once the device has been connected to it. Once the Raspberry Pi has been connected to Blynk, Blynk will then connect with it and begin transmitting data from the Raspberry Pi to Blynk. As soon as the raindrop module and the DHT22 humidity and temperature sensor are connected to the Raspberry Pi, they will begin to collect data on the weather conditions and deliver it to the Raspberry Pi.

Depending on your choice, the data will be displayed on any mobile devices you wish to use or on any LCD device that you choose, and the device will continue to keep sending you notifications about the weather conditions on a regular basis. There will be no end to this process until the user shuts down the device by closing the app. If water is detected, it will display "raining". Temperature and Humidity have a clear relationship based on the results. Their relationship is inverse. In high temperatures, humidity levels are low and in low temperatures, the humidity levels are high.

When the weather options are selected, the device provides accurate weather information about the current weather conditions in the options area as expected. This system is mainly intended to aid people in determining the current weather conditions around the area in order to prepare for the different weather events. User preparation for future weather conditions will be improved and any problems will be avoided, and This system also generates new ideas for making improvements or creating new devices that are more advanced and attractive.

Future recommendations could be to add on a GSM module or other device that can send data without the need for internet connections to send current weather information to users. In some rural or urban areas, internet connections are poor. Aside from improving the range, it could also increase the number of areas the device can cover, thereby making it able to detect and collect more weather data efficiently and at a larger scale. The final recommendation would be to add an anemometer to determine the speed of the wind in order to make future recommendations more accurate. Users might be able to avoid hurricanes and other destructive weather conditions by using this system.

2.4 Weather Monitoring and Forecasting System Using IoT

The authors of this article (Balakrishnan Sivakumar & Chikkamadaiah Nanjundaswamy, 2021) describe the real-time data is readily accessible over a broad range using IoT technology to make it an advanced weather monitoring system. Multiple sensors monitor the weather and climate conditions, including temperature, humidity, wind speed, soil moisture, light intensity, UV radiation and even carbon monoxide levels in the air. A web page is accessed to see the sensor data, which is then plotted as graphical statistics. Almost anywhere in the world can access the data uploaded to the web page. It is also possible to refer to the data gathered on these web pages in the future.

Alert systems are included as a means of ensuring people who are at risk of sudden and drastic weather changes are informed. Using an API, sensors collect data and predict weather forecasts that are not possible to predict directly. APIs allow access to data from anywhere and anytime, and they can be stored for later use. This project makes use of solar panels to power several components which consume very little electricity.

A suitable implementation model has been identified based on the framework, which includes a variety of sensor devices and other modules. The Node MCU is used in this implementation model for sensing and storing data. An embedded device connected to the internet is equipped with an inbuilt ADC and WiFi module. Monitoring is done with sensors connected to the Node MCU board. Analog sensor readings are converted into digital values by the ADC, and environment parameters are calculated.

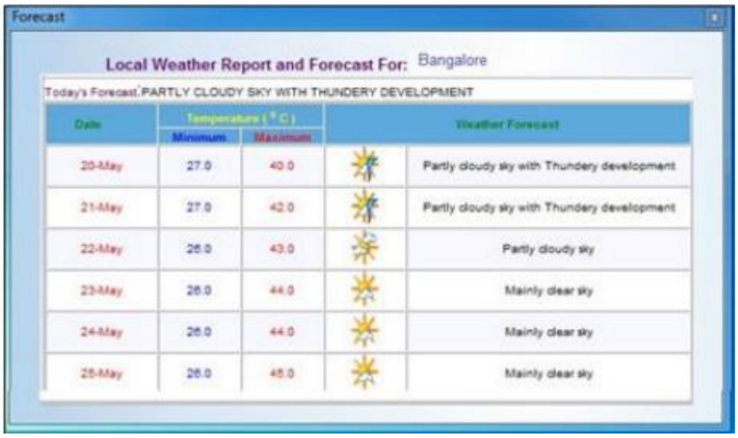


Figure 2.5: Result of Forecasting System

It is necessary to make sure that SMS service and GSM port are handled by a web interface gateway or GSM modem, depending on the result. Upon receiving SMS, the SMS will display the current weather statistics on the user's/receiver's mobile phone. This wireless weather monitoring system can be implemented practically using a microcontroller. The project utilizes an ATMEGA32 microcontroller to communicate with a GSM modem and display the temperature simultaneously on an LCD screen. By experimenting and implementing, found that LM35 (temperature sensors) can be used to monitor temperature in an isolated place and sent via SMS to a distant user's phone.

The implementation of this strategy requires the deployment of sensor devices in the environment for collecting the data as well as analysing it. With the deployment of sensor devices within the environment, it will be possible to record data in real time. This object can collaborate with other objects in the network with the network. As soon as the data has been collected and analysed, it is then made available to the end user with Wi-Fi.

In this paper, different models are presented to demonstrate the efficiency, low-cost entrenched system that can monitor the environment in a smart way. The weather station is a device that uses science and technology to determine and predict the weather conditions at a specific place based on the information provided by the weather station. Weather stations enable you to study and predict weather conditions by collecting quantitative data about the conditions in an area. There are many factors that affect human activities, including whether they occur or not. There is nothing like a short-term weather forecast that can help you plan for a day's worth of weather in advance, but forecasting a short-term weather can only be done as quickly and accurately as possible.

It is suggested that the system be improved by including an exterior cover for it, which can function as a protective cover for the system and enable the device to work in a harsh climate without causing damage to it.

2.5 An IoT-Based Real-Time Weather Monitoring System Using Telegram Bot and Thingsboard Platform

A study (Bestari & Wibowo, 2023) was done by Technological advancements, such as those in IoT, have enabled meteorologists to automate their work and observe real-time data. Mining regions are especially dependent on meteorological data. Data such as this can enhance workers' safety, pollution control, and water quality, such as dust and poisonous gases, runoff during rainy seasons, and chemical discharge into drainage systems.

By using an ESP32 microcontroller and SHTC3, BMP180, wind speed, and wind direction sensors, anyone can build IoT monitoring devices that measure temperature, humidity, air pressure, wind speed, and wind direction. Thingsboard and Telegram Chatbot use MQTT, Web Sockets, and HTTP requests to transfer data. Weather forecast information is enhanced in real-time using sensor data and BMKG's National Digital API. From a mobile phone, a user can obtain detailed weather information in real-time.

A recent study examined how instant messaging apps are used in people's daily lives and found that they give users access to local weather information and daily forecasts in real-time. The system's status can be accessed by managers and developers directly with the help of these APIs, without having to physically access the device. The chatbot is more efficient and can save a lot of time than Thingboard since it can access real-time information on mobile devices and respond to users in real-time. Typically, when a user uses the Thingsboard chatbot, he or she can only access selected data when using the tool.

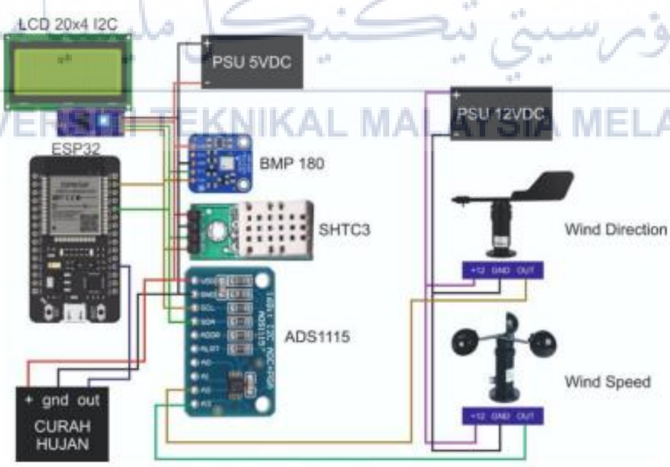


Figure 2.6:Circuit Diagram

Sensors will be used in IoT devices so that data can be collected from them. There are many types of sensors available to measure a variety of various temperatures, pressures, humidity levels, wind speeds and directions and precipitation amounts. A MCU node will process the data collected from the sensor then send it over the internet to be processed by the MCU node.

It is necessary to consider the energy capacity, distance, and bandwidth required for different features and functionality of IoT devices. Having stable internet connectivity is one of the most important requirements for the successful implementation of IoT technology. After the data has been collected, it is processed by software. The response of the device during this process can be affected by several factors.

Using IoT and data transfer systems, this research developed a real-time weather monitoring system. Changes in each weather parameter are captured by sensors. MQTT, WebSocket, and HTTP request protocols are supported by the Thingsboard platform and Telegram bot. Using the Telegram dashboard, you can monitor data at any time and call the Telegram chatbot via mobile phone.

This research can be further developed by adding database data. Farms, mines, and plantations can benefit from the research conducted in this area that can be integrated with smart farming, smart homes, and weather monitoring. With artificial intelligence, a connected device can predict and alert, for example, for high temperatures or with machine learning algorithms.

2.6 Real Time Weather Monitoring System Using IoT

A research article (Sharma & Prakash, 2021) is the system should contain a variety of sensors that monitor regional temperature, humidity, rain values, and system pressure. Sensors capture data and send it to MCU controllers in the nodes.

Sensor data is uploaded using Arduino. A serial monitor connects sensors with the cloud. The data is rejected by the sensor on the monitor. As part of Siri's monitoring, IP addresses are tracked.

A web server's data is viewed using the HTTP protocol. By using environmental parameters or sensors, real-time weather data can be displayed on a web server. Any web server can be used to monitor weather conditions from anywhere without any apps or websites. There is public access to data. The proposed system was used to measure the weather conditions in Gorakhpur. It has been observed that we achieve better results with the proposed weather model than with standard weather parameters based on a variety of sensor results.

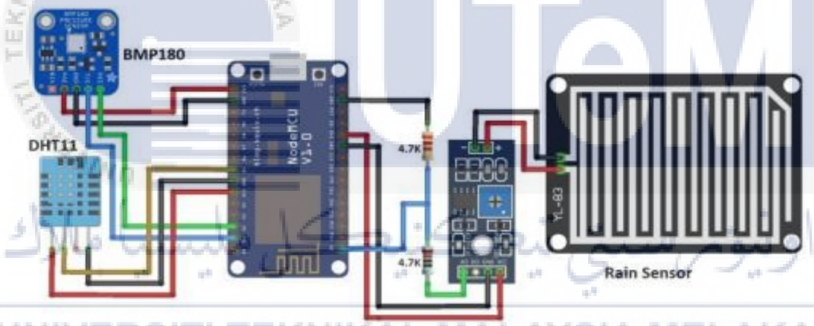


Figure 2.7: Circuit Diagram

It was developed using Arduino IDE with inbuilt libraries. We installed the latest versions of the DHT11 and Adafruit libraries. Temperature and humidity were obtained from a DHT11 sensor using an Arduino program. Using Python programming, it was connected to an open-source data collection tool that was connected to a MySQL database server where data was stored at specific intervals in a database. Python is used to interact with the COM4 port, and in the program is the database name that will be used for storing data. The task scheduler performs the task for the number of days over which the data needs to be monitored in specified intervals.

A system that can monitor the weather parameters at a minimum cost is proposed in this proposal. By using multiple sensors, it is possible to observe various environmental information at the same time. Compared to the existing model, the system that was designed has used a smaller number of sensors. It is their main goal here at VISIO to create a cost-effective, affordable model that aims to achieve the main objective. By using this proposed system, multiple sensors data will be collected and sent to the webpage on the web server by using HTTP requests protocol on the web server in order to capture the data. This paper presents the proposal for a system that can perform its duties in the region of Gorakhpur.

Their propose that the proposed model conducts not only a systematic collection of data, but also makes decisions according to the data observed. In terms of accuracy, the proposed model is nearly accurate with the real data collected. HTTP suffers from the problem of Head-of-Line blocking i.e., if in the transmission of data one packet fails then all other packets wait for the retransmission of the failed packet. Then it is suggested to use a simple application.

2.7 IoT Based Low-Cost Weather Station and Monitoring System for Smart Agriculture

This project by (Marwa et al., 2020) uses Internet of Things to monitor environmental parameters and Medenine region (Tunisia) will be monitored for changes in climate parameters. There are three main components of our system: the central node (gateway), the nodes (field-installed), and the cloud storage. With the

ZigBee communications module, the sensor measures fields and sends them to the gateway either one-time or repeatedly. Sensor nodes send miniature records to the cloud through the 4G / LTE module.

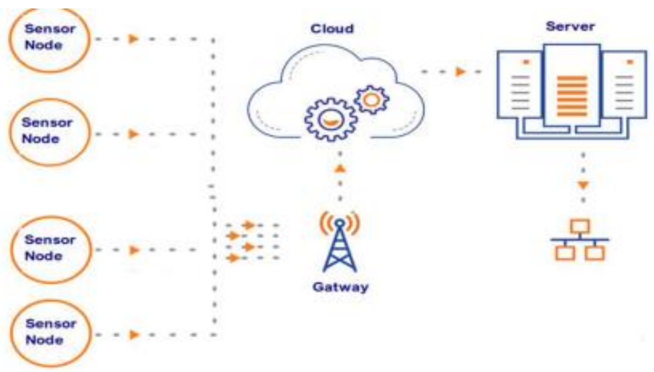


Figure 2.8: Iot Architecture System

A pilot field has been equipped with environmental equipment because of the analysis. The cloud-based data repository stores, transfers, and processes all relevant parameters as of the installation date. Every hour, 9691 weather reports are received for the following parameters: date, air temperature, relative humidity, and rainfall.

As the authors describe in their article, they are working on a monitoring station based IoT system as a key element of their climate monitoring systems are being implemented and research is being conducted intended for agricultural use. Temperature, humidity, and the amount of rain that is falling are all monitored by the system. As a result of using this system, they are able to conserve natural resources, save energy, as well as make it easy and inexpensive to operate.

A future extension of the system can be accomplished through the integration of other nodes equipped with environmental sensors, so as to expand the scope of the spatial network coverage and include more measures of agricultural parameters.

2.8 Arduino Based Weather Monitoring System

(Krishnamurthi et al., 2015) recommends using three sensors to measure weather/environmental parameters like temperature, humidity, light intensity, dew point, and heat index. A text file is created to analyze sensor readings processed by Arduino microcontrollers. A LCD display is also available on board for convenient viewing of readings. All these observations may be analyzed to determine an area's weather characteristics and to record weather patterns. It may produce a weather chart of a given area over time using these variables as input.

The prescribed actions are carried out based on the current weather conditions and preset settings. Activating the heating system occurs when the temperature drops below the set value; activating the cooling system occurs when the temperature rises beyond the set value. Databases may also store the serial output from the Arduino microcontroller, which is the sensor value.

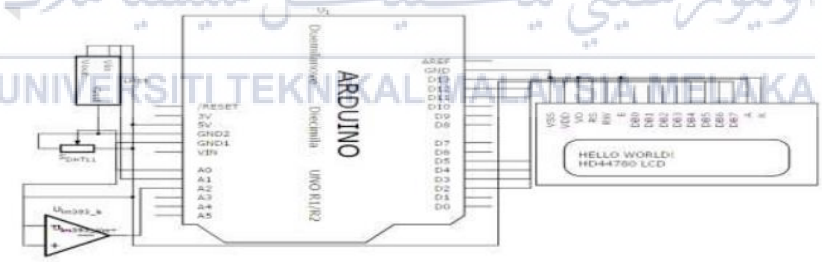


Figure 2.9: Circuit Diagram

For accurate temperature measurement, two temperature sensors, LMT35 and DHT11, are used. These two readings are averaged. A specified threshold temperature is maintained by delivering hot or cold air based on the temperature. A blasted stream of hot air is used to raise the temperature if it is too low in a particular location. When temperatures are too high, cold air is pumped to bring them down.

In this way, the temperature can be adjusted. Based on light intensity, the second kind of LDR functions. When the plant receives too much or too little sunlight, servo motors are used to open and close the glass door based on the LDR readings. We are able to document the amount of natural light present in a specific space in this way. The intensity of natural light may fluctuate over time. Applications in agriculture are crucial because certain plants cannot survive in low light, and light is necessary for plant development.

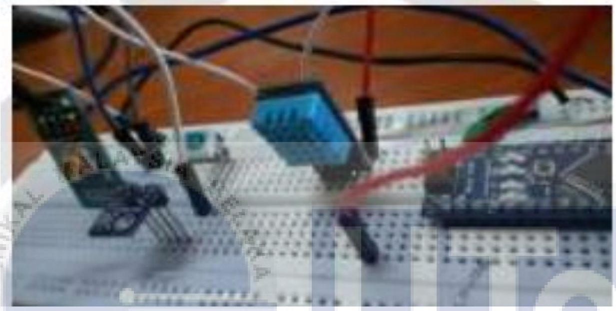


Figure 2.10: Sensor Array

The Arduino device communicates with a computer via the com port, and the acquired data is serially fed into a text file on the computer. The use of macro functions in Excel can make it easy to import text files into Excel documents. The imported data is then sorted and structured before being plotted onto a chart. The chart depicts a graphical representation of the data, displaying weather trends across the recorded time. Visual patterns depict the behavior of the weather in a certain location. This is the primary goal of this effort.

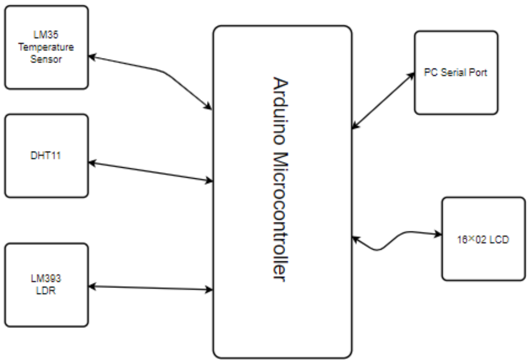


Figure 2.11: Architecture Design

The data is then used to create a graphical chart that provides a great analytical picture of the weather patterns based on the sensor readings. As a result, further tests in a setting more like real-world weather conditions are required. The findings of the experiment are shown in the table below.

Test Point	Module	Data
Tp0	Temperature	22.6 C
Tp1	Humidity	27%
Tp2	Dew Point	10.2 C
Tp3	Heat Index	23 C
Tp4	Light Intensity	675 Lux
Tp5	Temperature 2	23.2 C
Tp6	Power	+5v

Table 2.1: Result Of Experiment

In conclusion, this work has proven to be successful and is an efficient method of recording real-time weather data that will assist farmers whose livelihoods depend on it to produce better crops in countries like India. Each area's needs can be gathered over time by using this tool. Based on the collected information, farmers can determine which conditions are optimal for plant growth and adjust the environment accordingly. As well as agriculture, farmers around the world may see a substantial impact from it as well. This is a very limited weather system, as it is only designed for small areas, and it does not work on the web.

Future versions of the system can include sensors that analyse the quality of the air using gas detectors. The data can also be fed directly to the Internet using a web interface or service.

2.9 Raspberry Pi Based Weather Monitoring System

This study by (Bharadwaj, 2021)discusses a climate hypothesis framework designed for specified goals. The framework for estimating air temperature, air stickiness, light force, precipitation, and soil dampness can be made with less costly materials. This module can be used to screen the climate, anticipate worker data, and occasionally conduct information investigations using less costly materials.

The Raspberry Pi board was used to build this IoT-based climate monitoring framework. It is possible to limit the framework's equipment by using the Raspberry Pi board. Thus, this project does not require an external microcontroller, ADC, or communication module. Sensors utilized by the framework include Water Level Measuring Sensors (ULN2803), Temperature and Humidity Sensors (DHT11), Light Intensity Sensors (LDR), and Pressure and Altitude Sensors (BMP180). GPIO headers on the Raspberry Pi board are used to connect this collection of sensors. A network of Ethernet devices ensures that information from sensors is constantly checked.

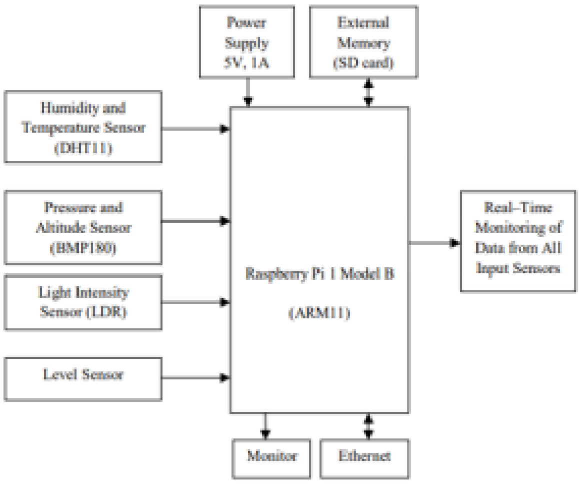


Figure 2.12: Block Diagram

Figure 2.13: Component Used

In terms of framework functionality, the Raspberry Pi board operates both as an information procurer and as a web worker. A number of sensors are used to gather data, including temperature, humidity, pressure, altitude, light power, and rainwater level. This data is then sent back to the consumer using the HTTP protocol. On the client side, thingspeak.com provides constant information from anywhere on the earth. By using an Ethernet connector or a USB dongle, you can connect to the website via LAN or USB. Using thingspeak.com, you will be able to see your results arranged graphically. Every field on this site is placed in a channel, which is created for each field.

This IoT-based platform provides continuous monitoring of ecological limits. Temperature, moisture, pressure, height, light force, and rainstorm water level are all monitored by this framework. Information may be accessed from any location on the earth. By using this framework, consumers can continually screen. No link between ecological limits and extra workers. Work is performed by Raspberry Pi. Raspbian's working foundation handles this excellently. Based on Raspberry Pi, this weather monitoring framework has a low cost, small size, low force requirement, rapid data transfer, outstanding execution, and remote checking. There are some limitations to this framework, including no built-in Wi-Fi or Real-Time Clock.

System management requires direct web access. A GPIO header must be connected to all sensors. It will be possible to use a more advanced Raspberry Pi board framework in the future. The framework can be expanded with more sensors and can be provided with further capacity with a sunlight-based board and a wind factory.

With the use of a Wi-Fi network, a viable climate estimating framework is constructed. Using economical components was the main goal in order to obtain the most exact framework that could continuously screen the climate application on rural grounds. The Raspberry PI was used to create a model of air temperature, moisture, light, soil moisture, and rain. A cutoff point routes information from the sensors to a general point where it can be seen, making it essentially accessible to everyone.

2.10 A Conceptual Design of LoRa based Weather Monitoring System for Smart

The Internet of Things (IoT)-based smart farming technologies offer new possibilities for agricultural field monitoring, according to research by SOY and DILAY (2021). In actuality, IoT-enabled devices can be connected using a variety of communication protocols. Low-power, wide area network (LPWAN)-based protocols are perfectly suited to the unique requirements of agricultural applications due to their energy-efficient transmission of small data packets over long distances. This study outlines the conceptual design of a LoRa-based weather monitoring system that could be used to notify farmers when unfavorable weather conditions could potentially cause crop damage.

The equipment plan of a low-cost climate station has been altogether point by point. In expansion, utilizing the Xirio Online radio arranging apparatus, the anticipated scope region of the application to be carried out to diminish the hazard of apple blooming ice harm chance in Karaman Area, Turkey was assessed. The information collected approved the common sense of the proposed climate observing framework.

Then there is the system model. The proposed system comprises of several low-cost weather stations put in various geographical places over a vast agricultural field to collectively monitor meteorological conditions. Figure 1 depicts the system model of the LoRa-based weather monitoring application. According to the LoRa network architecture, each weather station acts as an end device and communicates collected data to the network server via a gateway. The LoRaWAN protocol enables for many gateways to accept messages provided by any end device. The gateways collect data packets from end devices and forward them to the network server via an IP connection.

This study's primary commitment is a LoRa-based weather conditions station that best addresses the issues of horticultural applications. In our weather conditions station equipment plan, we utilized the TE Availability MS8607 series advanced sensor, which consolidates three ecological boundaries, specifically pressure, relative stickiness, and temperature. The MS8607 sensor's operating ranges are listed as 10 to 2000 mbar, 0 to 100 percent relative humidity, and -40 to 85 °C by the manufacturer. This large number of climate related boundaries are caught as 16-cycle values by the MS8607 sensor and shipped off the microcontroller through the I2C interface. It ought to be noticed that the MS8607 sensor is great for weather conditions stations because of its super low power utilization, high accuracy, and minimization (QFN bundle 5 3 1 mm3) (TE, 2017).

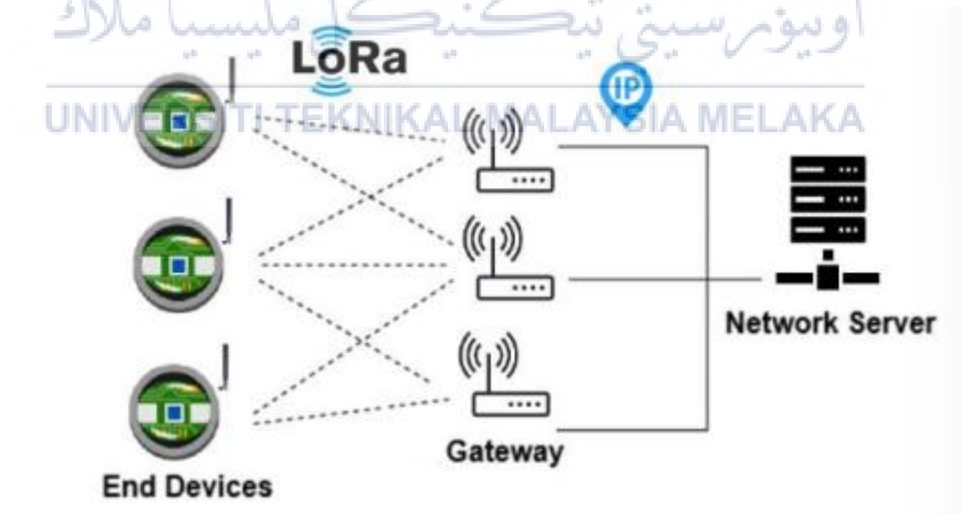


Figure 2.14: System Model of The Agricultural Weather Monitoring

Several LoRa deployments in agricultural applications that focus on remote monitoring of farming activities have recently been realized. Low cost, increased flexibility, and expanded coverage are essential needs for these applications to improve operational efficiency while preserving sustainable agricultural output. A revolutionary LoRa-based weather station prototype design for constructing a private agricultural LoRa network is provided in this research. Furthermore, the coverage area of the proposed LoRa network was evaluated using the Xirio Online radio planning tool. The apple orchards constructed around Akçaşehir region in Karaman province were chosen as the target. It is obvious that the finished conceptual design research will simplify the development process soon.

2.11 Prototype Weather Station Uses Lora Wireless Connectivity Infrastructure

(Murdyantoro et al., 2019) uses LoRa wireless infrastructure to build a prototype weather station. LoRa is a remote correspondence innovation that upholds the web of things (IoT) framework. This innovation is an option in contrast to other normal remote correspondence modules, like GSM modules, WiFi modules, and Bluetooth (BLE). The LoRa network is utilized to expand the scope of remote cells, which might venture out as much as 5 kilometers while consuming little power.

A single LoRa arrange cell can interface conclusion gadgets to hundreds of hubs. Temperature, mugginess, discuss weight, rain discovery, and wind speed are all measured climate parameters. The Arduino UNO, DHT11 sensor, raindrop sensor, anemometer sensor, BMP180 sensor, LoRa shield, LoRa Door, and ThingSpeak web application contain the model. Whereas the number of conclusion gadget hubs built in one cell is three. As a result, the framework can work ordinarily. With a few sorts of sensors connected, this framework has the potential to be connected in both urban and provincial areas. As a bigger sensor organize, cells can be overseen to make a Moo Power-Wide Region Organize.

This study was completed in the Unsoed Electrical Designing Research center, Jalan Mayjend Sungkono KM. 5 Blater, Kalimanah Area, Purbalingga Rule, Focal Java. A PC, Arduino programming (IDE), Proteus programming, thingspeak on the web, a mustimeter, thermometer, hygrometer, mark, and computerized anemometer are required. Lora passage LG01 915 MHz, lora watch 915 MHz, Arduino UNO Rev3, power bank, anemometer locator, downpour drop, DHT11, BMP180 muumuu links, PCB sheets, and USB connectors.

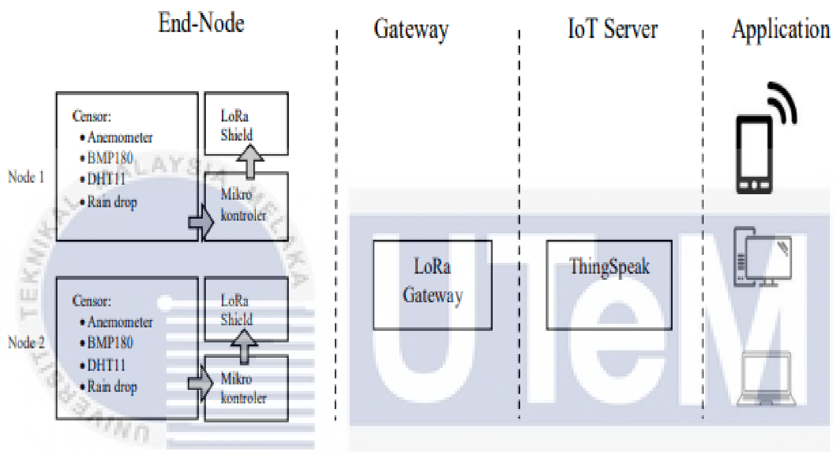


Figure 2.15: System Architechture

Figure 2.16: Result of Weather Measurement Data On Thingspeak Web

In light of the discussions, we can derive the following observations. The successful performance of a weather station prototype, which utilizes LoRa technology, is evident through its accurate and dependable display of sensor readings on the ThingSpeak interface. The LoRa platform offers an advantageous and viable option for creating sensor networks that necessitate numerous nodes within a single cell while maintaining low power consumption. By employing LoRa-based systems, it becomes feasible to establish sensor networks with expansive coverage and a higher node capacity.

2.12 LoRa based Wireless Weather Station with WebServer

This project by (Sai Manasa et al., 2023) involves developing a "LoRa Based Wireless Weather Station" concept. LoRa is a detached organisation creation that capacities the Internet of Things (IoT) construction. This change offers an alternative to famous detached organisation modules to a degree GSM modules, Wi-Fi modules, and Bluetooth modules. The LoRa network expands the range of detached containers, that concede possibility climb to 8 km while absorbing little capacity. Temperature, pressure, temperature when dew forms, peak, precipitation, moisture, and light force are the feeling edges that we estimate for this work. This model form use of the Arduino UNO, a barometric sensor, a rain sensor, a light force sensor, and LoRa modules.

Furthermore, the Thingspeak Web application allows clients to share information. This architecture could be used in metropolitan and provincial locations with a variety of sensors. This project can be used to build a Low Power-Wide Area Network as part of a broader sensor organisation. Gateway-Node LoRa receives communication information from LoRa and sends it to the internet using the ESP32

Wi-Fi Module. The client can access/notice the information via the Web or Thingspeak.

A LoRa-based weather station must include a Sender and Receiver circuit to communicate data remotely. We can keep the Weather Station framework on top of your house or at another place just a few km away. Sensors include the BME280 Barometric Pressure Sensor, the BH1750 Light Sensor, and the Rain Sensor. Essentially, this weather station may show environmental data including temperature, humidity, pressure, altitude, dew point, rainfall, and light intensity.

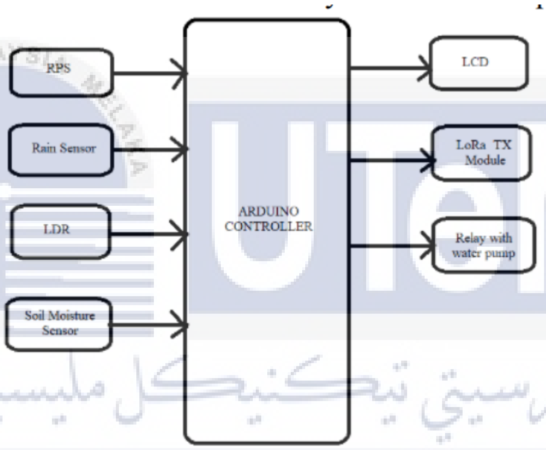


Figure 2.17: Block Diagram of transmitter

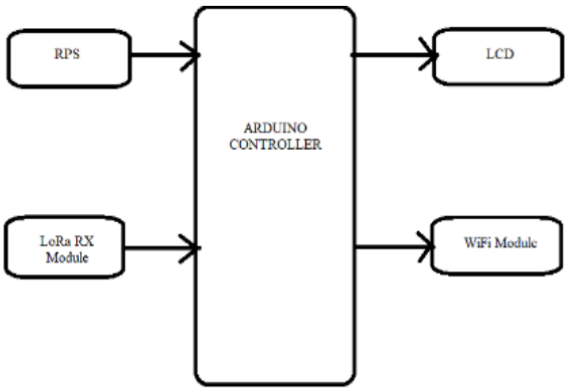


Figure 2.18: Block Diagram of transmitter

After all the connections have been completed, the Arduino nano is connected to the computer, and the programme written on the Arduino nano is dumped. The Arduino mini is meant to transfer collected data to a web server, which is monitored. They can keep track of the pressure, altitude, temperature, light intensity, and rainfall on the web server.

The reason for this task is to consistently screen and measure different factors, including temperature, pressure, downpour fall and level, and light power. The data can be sent to a web server, where it can be used to forecast weather and eventually analyse climate patterns, among other things. The system uses an effective combination of analogue and digital sensors in both wired and wireless modes of operation. As a result, a proof of concept for a weather monitoring system using an Internet of Things device has been created.

2.13 Development of Blynk IoT Platform Weather Information Monitoring System

The Internet of Things (IoT) is a modern technology that connects physical equipment and other devices to the Internet network for a specific use. The demand for weather monitoring apps prompts a smart IoT development drive. (Kamarudin et al., 2022) needs to confirm ecological circumstances prior to participating in open air exercises might require an earnest weather conditions alert. Subsequently, this article portrays a climate data observing framework gadget that can proficiently screen weather patterns continuously and give notices by means of the versatile Blynk application, SMS, and email. The gadget is involved an ESP8266 microcontroller, a temperature-dampness sensor (DHT11), a raindrop and carbon monoxide (CO) identification sensor, and ongoing information move over GSM and Wi-Fi organizations. The proposed framework is worked with an easy-to-understand

GUI portable application and is Online, permitting clients to get weather conditions observing framework refreshes whenever and from any area. To start, a 5 V, 1200 Mama power supply is important to instate the framework.

The associated sensors should be saved dynamic for information from the sensors to be conveyed to the NodeMCU ESP8266 microcontroller. The microcontroller is customized utilizing the Arduino UNO SMD Rev3. All sensor information is shipped off the Blynk server over a GSM and Wi-Fi organization. Blynk is an ongoing stage that permits clients to make interfaces fundamentally for the iPhone working framework (iOS) and Android PCs to track and screen the important applications. The gained information is examined and utilized in information perception empowering end clients to get climate related data refreshes.

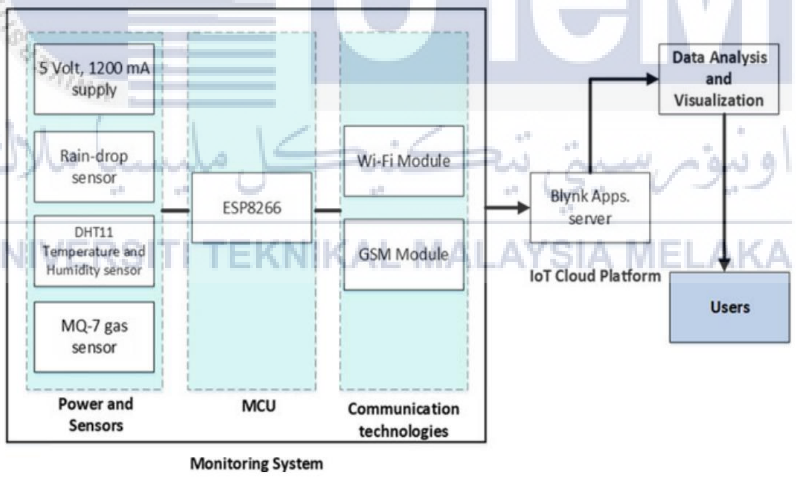


Figure 2.19: Block Diagram of The Weather Monitoring System

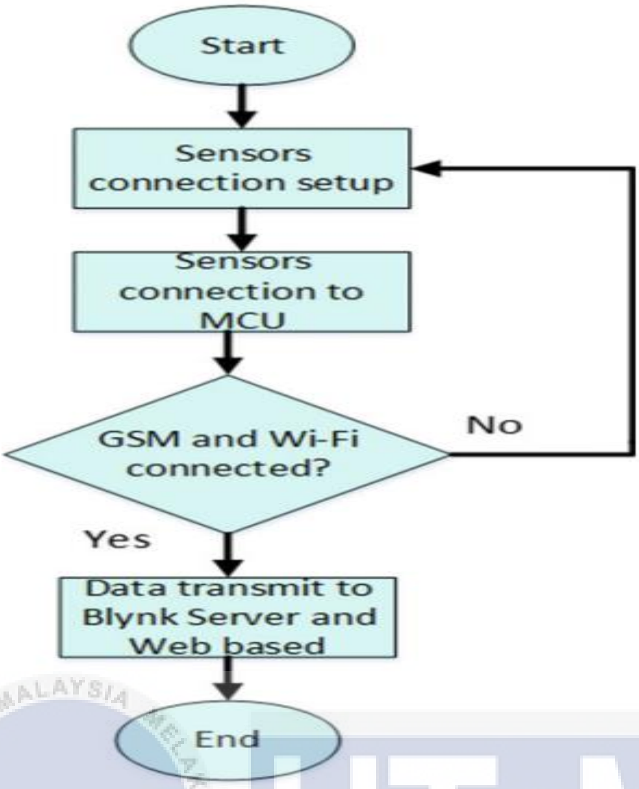


Figure 2.20: Flowchart of System

If the air quality file, or AQI, score surpasses 300, the climate is considered hazardous, and a crisis alert is probably going to be given. In ordinary room temperature settings, the typical temperature, moistness, and downpour drop rate are 31 °C, 71%, and 1.07%, separately, when the downpour drop sensor is dry and not covered by any downpour drop. The downpour drop level of 40% is assigned as a weighty downpour condition, hence a caution warning is sent through versatile applications as an initial step.

Significant developments in hardware design can be included in the final model of the device system to achieve superior outcomes and commercialised on a global scale. Future work should be done to improve sensor sensitivity and adopt long range (LoRa) communication technology connectivity as low power and long-range technological capabilities are preferred.

2.14 Development Of Wind Monitoring Systems with Lora Technology

The primary goal of this project by Purnomo et al. (2022) is to build and implement a wind monitoring system using LoRa (Low Range) technology. This gadget estimates wind speed, air temperature, and moistness. This information should be handled to acquire data about the ongoing status of the air. Existing information can be utilized as a source of perspective point for navigation or as an advance notice indication of a danger. This checking framework notices cools utilizing remote LoRa (Long Reach) innovation as an information move system. In this examination, two LoRa hubs and one Lora Passage are utilized.

In this instance, an anemometer and a BME280 sensor are utilised. The LoRa hub's sensors measure wind speed, air temperature, and stickiness. The LoRa hub sends all information to the LoRa entryway. A LoRa passage associated through Wi-Fi sends information to a server (The Things Organization or Firebase). As per the consequences of these analyses, the typical deferral for sending LoRa hub information to Firebase is 0.25 seconds. LoRa gadgets with a 3 cm long radio wire can impart up to 400 meters away.

The wind speed monitoring system's architecture consists of LoRa nodes coupled to various sensors, LoRa gateways, LoRa WAN network servers, and databases. The LoRa gateway connects the LoRa wireless network to the IP network via Wi-Fi and Ethernet [16]. As a LoRa WAN network server, this system employs The Things Network (TTN). The Things Network is an open-source infrastructure that provides a free and open set of tools and networks for the development of IoT applications. The sensor data is subsequently stored in a database via HTTP interaction with The Things Network. The wind speed monitoring system's architecture is depicted in Figure 2.21.

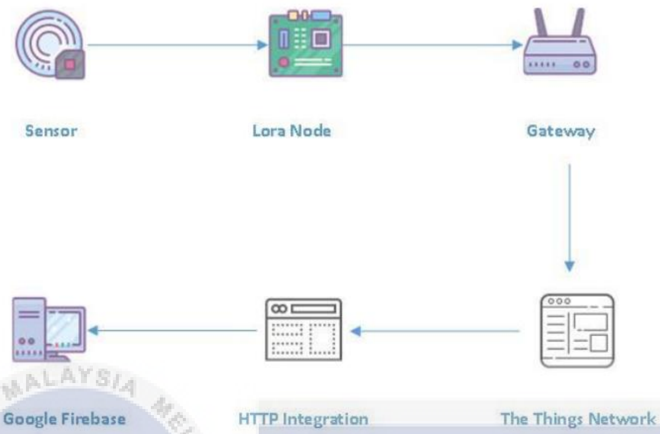


Figure 2.21: System Architechture of Wind Speed

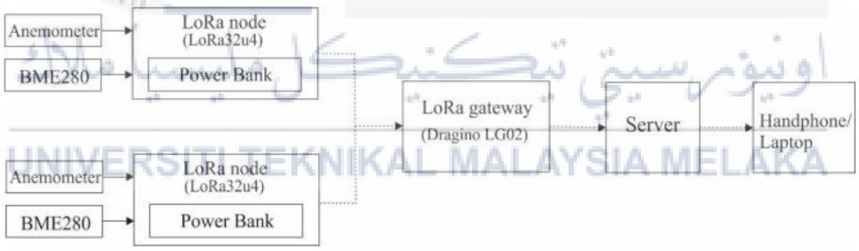


Figure 2.22: Block Diagram of Wind Speed, Temperature And Humidity Monitoring System

The time disparity between sensor readings and data received by the database is being tested on the node. The out Voltage multiplied by the number 6 determines the accuracy of the anemometer sensor in reading speed. And the output voltage is calculated by multiplying the sensor value by (5.0/1023.0).

NO	Distance (Meters)	Delivery Status (per 1 minute)
1	200	Sent
2	300	Sent
3	400	Sent
4	500	Not sent
5	> 500	Not sent

Table 2.2: Test Results of Data Delivery Against Distance

No	*Counter*	Serial Monitor (hour:minute:second)	Firebase (hour:minute:second)	Time Delay (second)
1	0	04:18:20,9	04:18:21,2	0,3
2	1	04:19:23,9	04:19:24,2	0,4
3	2	04:20:26,8	04:20:27,0	0,2
4	3	04:21:29,7	04:21:29,9	0,2
5	4	04:22:32,6	04:22:32,8	0,2
6	5	04:23:35,5	04:23:35.8	0,3
7	6	04:24:38,4	04:24:38,6	0,2
8	7	04:25:41,3	04:25:41,5	0,2
9	8	04:26:44,2	04:26:44,4	0,2
10	9	04:27:47,1	04:27:47,4	0,3
		Average		0,25

Table 2.3: Time Difference Between Sensor Readings And Data Received By The Database.

The delay between the information read by the sensor and sent by the hub and the information got in the data set is determined for dormancy testing. This information dormancy is critical on the grounds that it decides if the information shows up in the data set when the sensor assembles it. The sensor reading time from the Serial monitor is used to calculate the data transmission time. The timestamp of the data collected in the Firebase interface is used to calculate the time of receipt.

Table 2.2 displays the data delay. The average latency between data transmission and data reception in Table 2.3 is 0.25 seconds. Wind speed, temperature, and humidity can all be measured using nodes in the wind monitoring system. The transmitted data is characterized as having near-zero data latency or being near real-time data.

As indicated by the outcomes, the LoRa Hub moves all information to the LoRa Passage. The LoRa passage's Wi-Fi association will be shipped off the server (The Things Organization and Firebase). As per the test discoveries, the typical inactivity for conveying LoRa hub information to Firebase is 0.25 seconds. LoRa gadgets with a 3 cm long receiving wire can convey up to 400 m away.

2.15 LoRa Based Smart City Air Quality Monitoring System

This proposed solution by (Jayasree et al., 2023) is built on IoT and embedded devices. This system also uses electronic devices such sensors to detect ambient variables and the Arduino Uno board to improve system performance. This system includes a LoRa module for long-distance communication. Communication modules such as Bluetooth, Zigbee, and Wi-Fi can also be used, but LoRa is used in this system to communicate over great distances. In comparison to the LoRa module, Bluetooth, Zigbee, and Wi-Fi are typically utilized for short-range communications. This project is a LoRa-based smart city air quality monitoring system. Its primary function is to monitor smart city environmental parameters such as temperature, humidity, rain, and pressure, as well as detect light and contaminants in the air. The values of these environmental parameters are saved on a cloud server that communicates with the LoRa Gateway. The user can monitor data from the cloud server, and results are displayed in graphs or tables.

The proposed study is about creating and implementing a smart city air quality monitoring system based on LoRa. A few sensor nodes in the framework measure natural qualities like temperature, stickiness, downpour, light opposition, and airborne poisons. By connecting all the sensors, a NodeMCU board, and a LoRa module, a transmitter is formed.

Pollutants in the environment are detected by these sensor nodes. These sensor nodes are distributed across the environment. Data from each sensor node is sent via the LoRa module to the LoRa gateway, transferred to the cloud, sent to Blynk IoT's web server, and viewed via smartphones via the internet.

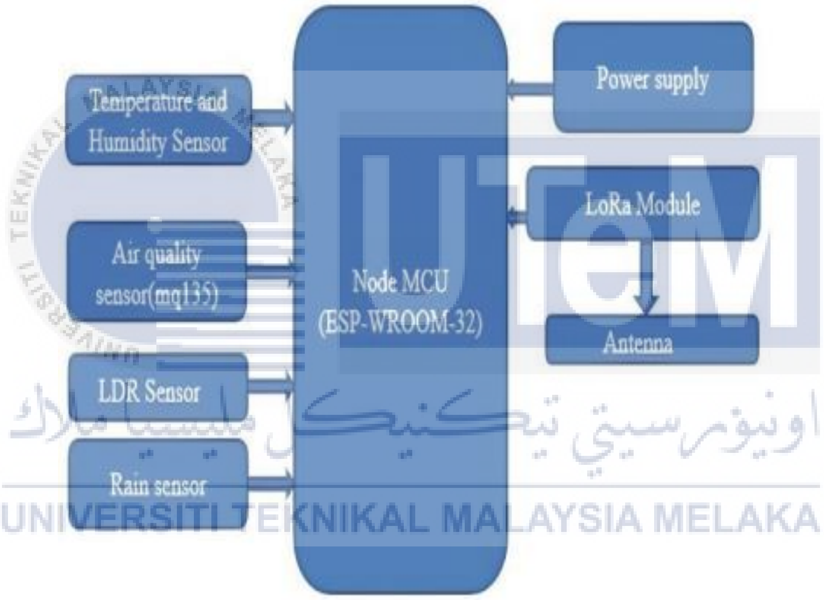


Figure 2.23: Block Diagram of Transmitter Section

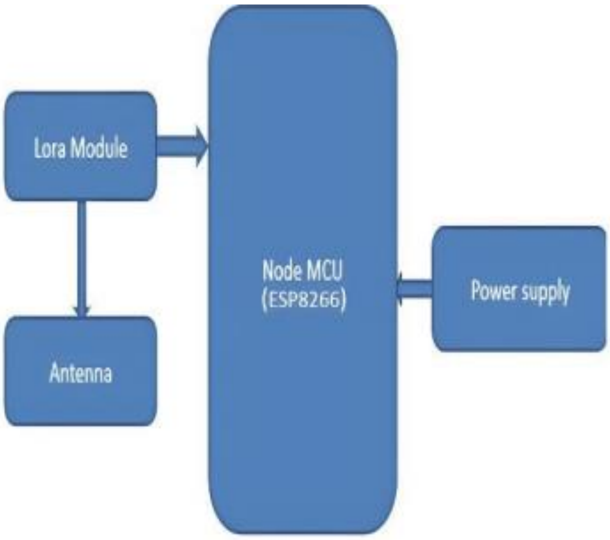


Figure 2.24: Block diagram of Receiver Section

There have been several devices for monitoring air quality. The reliability of the utilized devices is not sufficient in all circumstances, and they function differently in laboratory and real-life scenarios. LoRa devices with gateways and servers have been introduced, and they have helped to fully cover the city area for monitoring air quality. This widespread networking of various types of devices demonstrates its scalability. Furthermore, Lora WAN technology devices are made affordable to clients. The usage of wireless sensor networks via the LoRa module increased the flexibility of the applications for remote monitoring. In this way, LoRa technology outperforms other technologies.

The received data is delivered to the reception station from the transmitter part via the Lora module. From the receiver section, the received data is sent to the user's mobile device and saved in the cloud. The data can be seen on mobile and online dashboards in the BlynkIoT application.

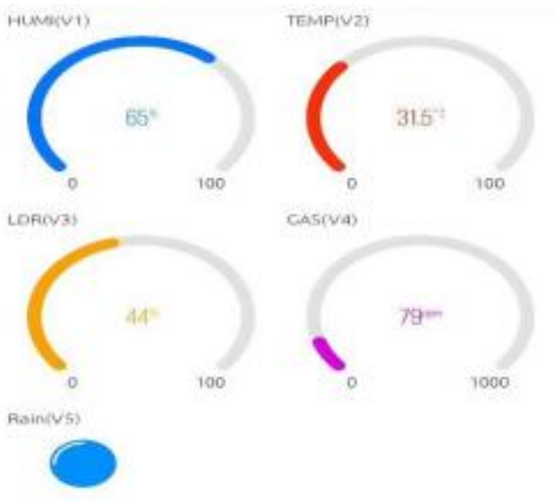


Figure 2.25: Result Sensors of Gauge Representation

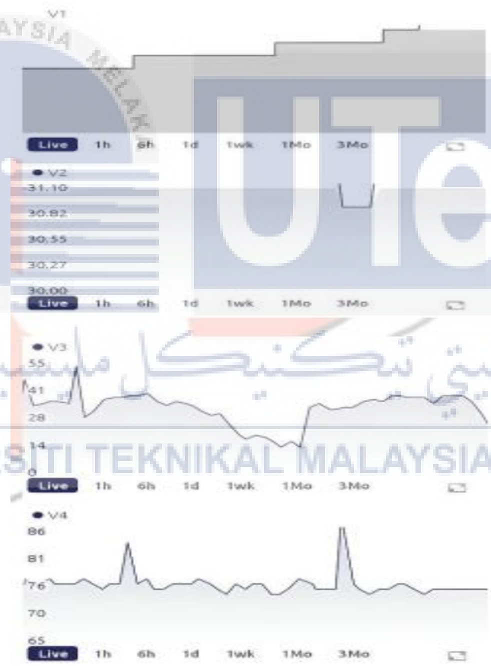


Figure 2.26: Result Sensor of Graphical Representation

This LoRaWAN-based air quality monitoring device We have done well in this system to detect atmospheric changes. Change in CO, CO2, and PM exposure. We have also noticed a shift in air temperature and humidity. The atmospheric pressure, rain, and environmental light intensity are also measured. Data from the sensor node is transferred to the gateway and then to the Blink IoT server. The communication method between transmitter and receiver has been tested in both outdoor and interior environments. During the testing phase, the transmitter was placed outdoors, the receiver was placed indoors, and the transmitter was placed indoors to evaluate the interior environment characteristics. Both the transmitter and receiver devices can communicate over extended distances of up to 3 km.

Future research may concentrate on increasing coverage. LoRa technology can be used in high-traffic regions by adding a gateway. New types of sensors can also be added to detect other sorts of gases.

2.16 Low-Cost Air Quality Monitoring System Using LoRa Communication Technology

A study (Fadhli et al., 2022) for this project uses LoRa-based communication technology to propose a low-cost air quality monitoring system. The suggested system consists of many sensor nodes and a gateway. Sensor nodes use the MQ-7 sensor for CO gas, the MQ-135 sensor for CO2, the GP2Y1010 sensor for Particulate Matter (PM), and the DHT-22 sensor for air temperature and humidity.

The data processing centre is an Arduino UNO R3. The LoRa RF96 module transmits sensor data to the gateway. The gateway uses NodeMCU V3 to deliver sensor data to the Thingspeak server, which users can access over the internet. The results of the testing show that the proposed system can detect contaminants in the air and communicate them to the Thingspeak server. Up to 32 metres, an interior gateway could successfully receive data from an outdoor sensor node. At 50 metres, the gateway had a data reception success rate of 99.17%, whereas at 70 metres, it was 78.3%.

The air quality monitoring system described in this research consists of multiple sensor nodes and a single gateway. Sensor nodes serve as pollution exposure readers in a variety of places. The impurities detected include CO, CO2, PM, as well as air temperature and humidity data. Data from each sensor node is routed to the gateway via the LoRa module, and then to the Thingspeak server, which is accessible via the internet network.

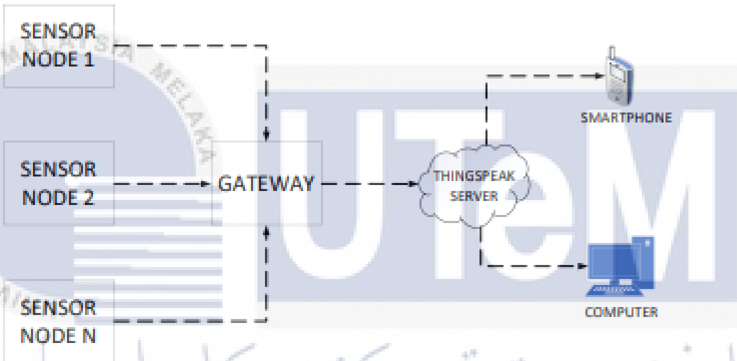


Figure 2.27: Block Diagram of Overall System

Table shows the results of testing the communication distance of sensor and gateway nodes. The table displays the number of packets that were successfully received without problems, the number of packets that were received with errors, and the number of packets that were lost. All packets sent could be received by the gateway without error at distances of 18 and 32 meters. 119 packets were successfully received while at 50 meters. Only 94 packets were successfully received at 70 meters.

Distance (Meter)	Sent Packets	Received Packets	Error Packets	Lost Packets
18	120	120	0	0
32	120	120	0	0
50	120	119	0	1
70	120	94	17	9

Table 2.4: Result of Communication Distance Measurement

The gateway passed data to the Thingspeak server. It was entered in the relevant field, along with the sending node's address, so that it could be accessible via the internet network. All packets received by the gateway could be routed to the Thingspeak server based on the test findings. The Thingspeak page is depicted in Figure 2.28.

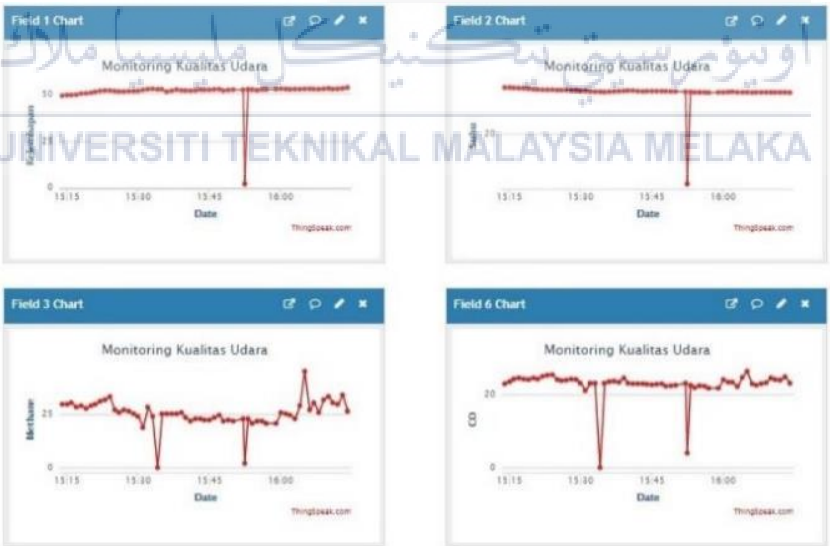


Figure 2.28: Page of Thingspeak

In this study, a LoRa-based low-cost air quality monitoring system performed effectively in detecting changes in CO, CO2, and PM exposure. It is also effective in detecting changes in air temperature and humidity. Sensor node data can be transmitted to the gateway and routed to the Thingspeak server. An outdoor-indoor scenario was used to test the communication mechanism between the sensor node and the gateway.

2.17 A Low-Cost Microcontroller-based Weather Monitoring System

This paper(Ariffin Noordin et al., 2006) describes the improvement of a weather conditions checking gadget that utilizes simple and computerized parts to screen and show the temperature, tension, and relative mugginess of the climate. The sensors' simple results are coupled to a microcontroller through an ADC for computerized signal change and information logging. The microcontroller is also linked to an LCD display, which displays the measurements. The data can be sent to a PC with a graphical user interface programmed via a USB link for analysis and preservation.

The connection point modified permits you to characterize inspecting boundaries, for example, the date and season of the information logging action. As far as size, immense memory limit, on-gadget show, lower cost, and higher versatility, the contraption enjoys various upper hands over past weather conditions checking gadgets.

The sensor circuit, information logging circuit, time-keeping circuit, and USB communicating circuit are the four vital parts of the framework. The sensor circuit includes an integrated circuit temperature sensor, a resistive humidity sensor, and a barometric pressure sensor. An ADC converts the analogue outputs of these sensors into digital signals before feeding them into the data-logging circuit, which includes a microcontroller. The time-keeping circuit provides the current time for data logging purposes, while the USB interface circuit permits data transfer between the data logger and a PC.

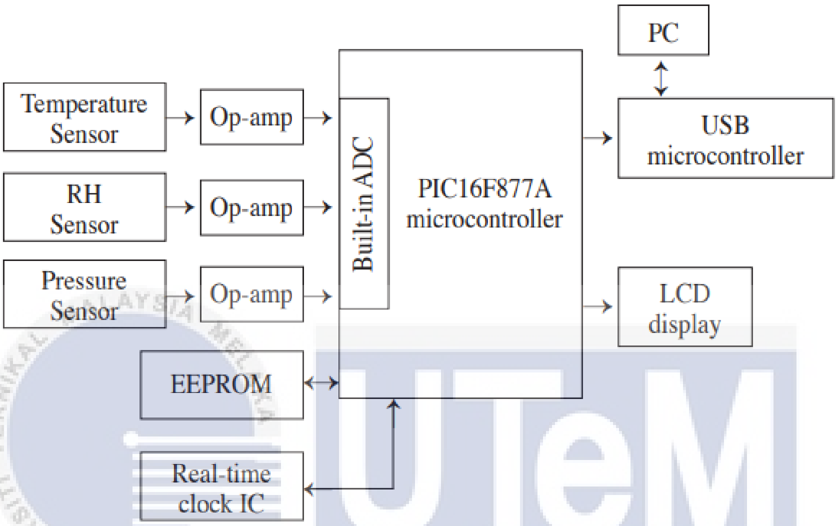


Figure 2.29: Block Diagram of The System

Extensive trials have been done to validate the proposed system's accuracy. The measurements were compared to those taken with more advanced equipment used by the Malaysian Meteorological Services Department (MMS), which included a thermometer, a relative humidity sensor with chart recorder, and a Vaisala PTB220A1 pressure transmitter for temperature, relative humidity, and pressure measurements, respectively.

	JDC Instruments SKYWATCH	Kestrel 4000 Pocket Weather Tracker	Grant Mini-met Weather Station	Oregon Scientific WM-918	Proposed system
Temperature Accuracy	±0.5°C	±1°C	±0.4°C	±2°C	±0.7°C
RH Accuracy	±3%	±3%	±2%	±6%	±2%
Pressure Accuracy	±0.2 kPa	±0.3 kPa	±0.03 kPa	±0.7 kPa	±0.2 kPa
Memory Capacity	-	250 measurements	64000 measurements (separate data logger)	-	512 Kb
Backup power	CR2032 lithium battery batteries	2 AAA alkaline batteries	12 V rechargeable battery and solar power supply	8 AAA batteries	Rechargeable 9V battery
Price (US$)	380	330	Not available	90	< 70

Table 2.5: Comparisons of Other Weather Station Devices With The Proposed System

In terms of features and accuracy, several portable weather station devices on the market are compared to the proposed system. Table summarizes the comparisons. It Table shows that the suggested system has a substantially higher memory capacity for 38 CMU. Journal (2006) Vol. 5(1) 39 data-logging purposes when compared to other weather station devices, one of which requires a separate data logger.

A framework has been proposed that combines the usage of sensors in the development of a low-cost, high-accuracy weather monitoring system that employs both analogue and digital components. Extensive trials were conducted on the proposed system, and the results demonstrated its accuracy and reliability. Furthermore, a comparison of the features of various types of monitoring systems was performed, and the results demonstrate that the suggested system is a better choice in terms of cost, portability, memory capacity, and logging interval-setting capabilities.

2.18 Internet of Things (IOT) Based Weather Monitoring System

This study's conduct by(Nerella, n.d.) proposed system is a sophisticated approach for monitoring weather conditions in each region and making the data available elsewhere in the world. The Internet of Things is the technology underlying this. The Internet of Things (IoT) is a more advanced and efficient technology for linking devices to the internet and the complete universe of things in a network. Electronic devices, sensors, and vehicle electronic equipment might all be found here. The system uses sensors to monitor and adjust environmental factors such as temperature, relative humidity, light intensity, and CO level, and then sends the information to a web page to plot the sensor data as graphical statistics. The data updated by the established system is available on the internet from anywhere in the world.

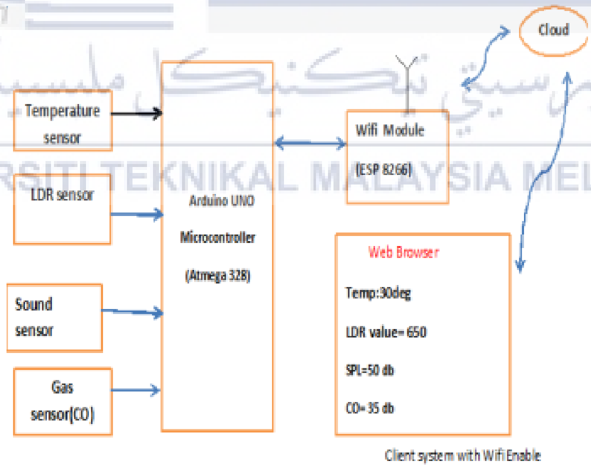


Figure 2.30: Block Diagram of The Project

The designed system includes a microcontroller (ATmega328) as the primary processing unit for the entire system, to which all sensors and devices can be connected. The microcontroller can operate the sensors to retrieve data from them, execute the analysis using the sensor data, and upload it to the internet via the Wi-Fi module that is attached to it.

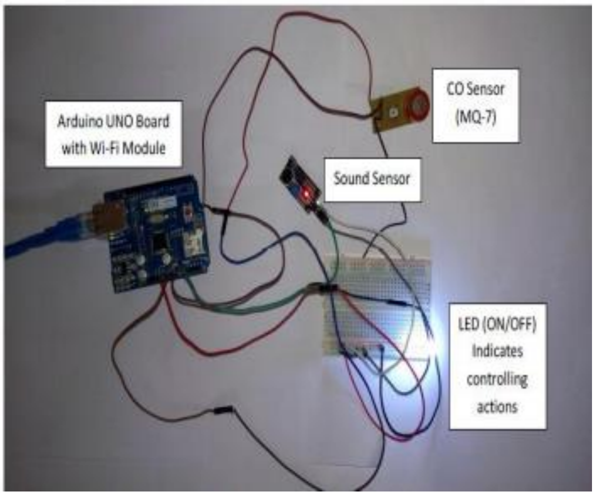


Figure 2.31: Noise And Air Pollution Monitoring Embedded System With Its Components

The embedded system and its components for reading and storing pollutant parameters in the cloud. After sensing is finished effectively, the information will be dissected and saved in a data set for future reference. Following the finishing of the information examination, edge values will be laid out for the purpose of controlling.

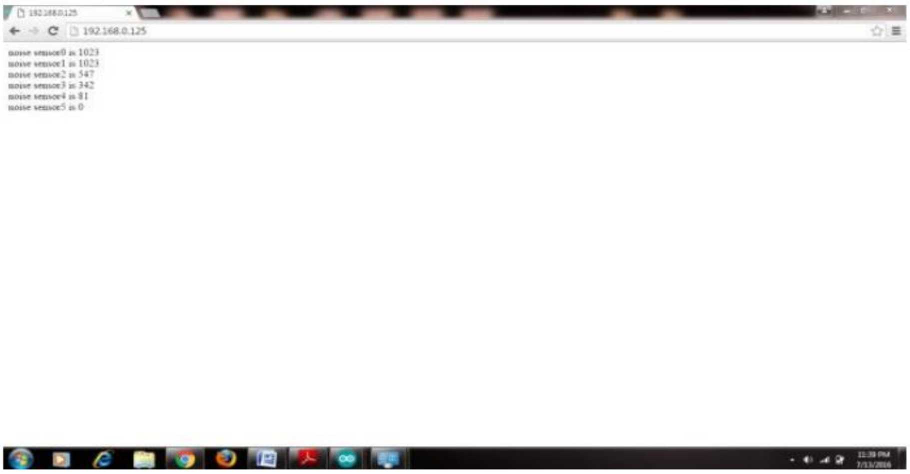
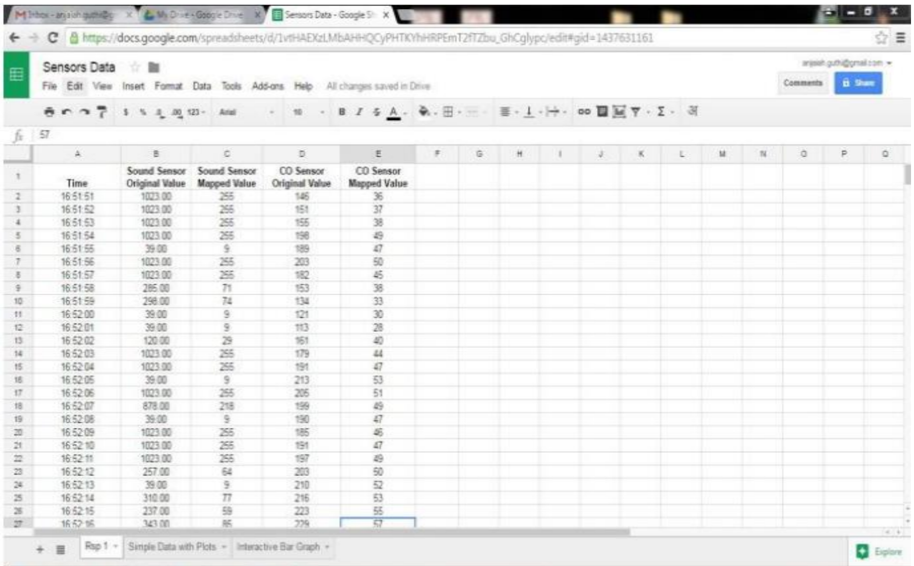


Figure 2.32: Page of Web Server



	Time	Sound Sensor Original Value	Sound Sensor Mapped Value	CO Sensor Original Value	CO Sensor Mapped Value
2	16:51:51	1023.00	255	146	36
3	16:51:52	1023.00	255	151	37
4	16:51:53	1023.00	255	155	38
5	16:51:54	1023.00	255	198	49
6	16:51:55	39.00	9	189	47
7	16:51:56	1023.00	255	203	50
8	16:51:57	1023.00	255	182	45
9	16:51:58	285.00	71	153	38
10	16:51:59	298.00	74	134	33
11	16:52:00	39.00	9	121	30
12	16:52:01	39.00	9	113	28
13	16:52:02	120.00	29	161	40
14	16:52:03	1023.00	255	179	44
15	16:52:04	1023.00	255	191	47
16	16:52:05	39.00	9	213	53
17	16:52:06	1023.00	255	205	51
18	16:52:07	878.00	218	199	49
19	16:52:08	39.00	9	190	47
20	16:52:09	1023.00	255	185	46
21	16:52:10	1023.00	255	191	47
22	16:52:11	1023.00	255	197	49
23	16:52:12	257.00	64	203	50
24	16:52:13	39.00	9	210	52
25	16:52:14	310.00	77	216	53
26	16:52:15	237.00	59	223	55
27	16:52:16	343.00	85	229	57

Figure 2.33: Cloud Storage (Google Spread Sheets) For Sensors Data

The sensed data will be recorded in the cloud (Google Sheets). Data stored in the cloud can be used to analyse parameters and monitor them continuously. The diagram above illustrates the noise intensity and CO levels in the air at regular intervals. All of the above data will be stored in the cloud, allowing us to provide trending of noise intensity and CO levels at a specific place at any time.

The functions of several modules were discussed in the proposed architecture. The noise and air pollution monitoring system using the Internet of Things (IoT) concept was tested experimentally for two parameters. It also transferred sensor data to the cloud (Google Spread Sheets). This information will be beneficial for future research and may be easily shared with other end users.

This model can be expanded to monitor pollution levels in emerging cities and industrial zones. To protect public health from pollution, this model provides an efficient and low-cost option for continuous environmental monitoring.

2.19 Weather Monitoring System

This document by (Mahajan & Supriya Konar, 2023)Weather is the state of the atmosphere and is critical to human survival. Scientific research and critical decisions such as where to live are influenced by the weather in that region, state, or space. People will require ambient temperature and humidity measurements for this, which weather stations can provide. The major function of this weather station is to measure the local atmospheric weather observations. Temperature and humidity sensors will record these readings. Every activity of this station is controlled by an Arduino UNO, and all readings are displayed on an LCD display.

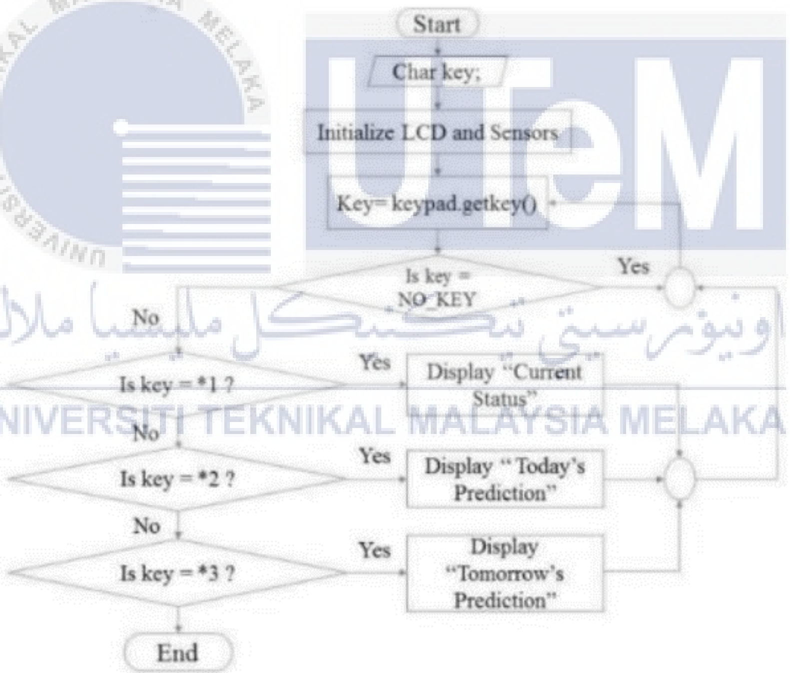


Figure 2.34: Flowchart of Weather Monitoring System

When the Arduino is linked to the power supply, it will power the entire station. DHT22 Temperature and Humidity sensors are used to monitor the data LCD. The Arduino board is linked to the LCD and sensors. To retrieve the output, the code is executed. At first, the readings with those will be recorded and ready to display. The system constantly monitors temperature and humidity utilizing temperature and humidity sensors.

Weather monitoring systems detect and collect numerous weather characteristics at various locations and show them on LCD screens so that they can be studied or used for weather forecasting. As we modify the sensor's settings, the reading will vary accordingly. The sensor's data is uploaded to an open-source web server. The open server can be used to obtain real-time data as well as data for the prediction process.

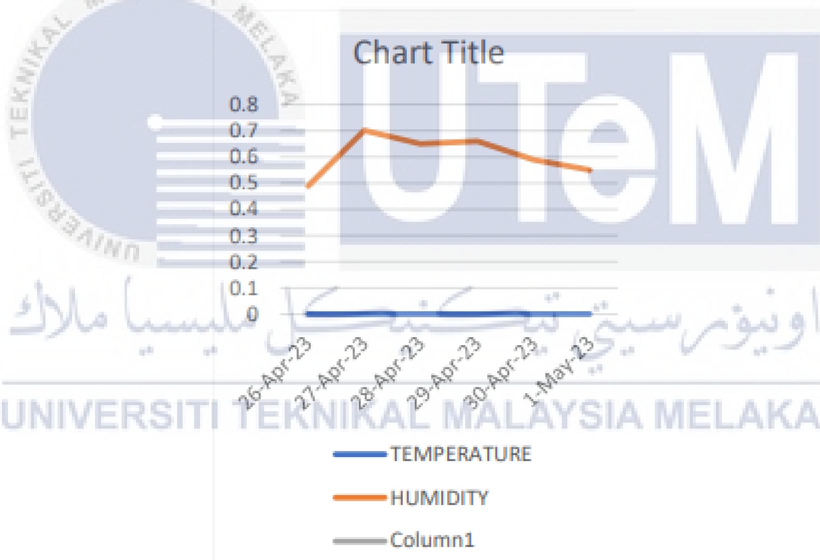


Figure 2.35: Graph of Weather Monitoring System

The graph depicts the relationship between temperature and humidity. This meteorological station provides temperature and humidity measurements in real time. These temperature and humidity statistics can be used to forecast the weather. A weather reporting system's primary function is to continuously track weather and climatic conditions over controlled areas such as buildings, enterprises, farms, and so on. We built a weather station using temperature and humidity sensors in this project.

This project provides a clear instance of a system that employs sensors to track weather readings. The sensors will interact with the local weather and display data on an LCD screen. Because of the ever-changing climate, weather forecasts are erratic and untrustworthy these days. The meteorological Reporting System is thus widely employed in highly regulated domains such as residences, industry, agriculture, and so on to track the constantly changing climatic and meteorological conditions.

2.20 Micro-controller Based Mobile Weather Monitor System

The goal of this study by Ibrahim Musa (2018) is to investigate the significance of stored data and records of weather parameters such as temperature, atmospheric pressure, relative humidity, and wind speed, which can be measured remotely using appropriate sensors and are useful for environmental monitoring, weather forecasting, and many industrial processes. This gadget measures, processes, and records air temperature, atmospheric pressure, relative humidity, sun radiation, and wind speed, which it then displays.

This project intends to create and install a mobile weather monitoring system that can be accessed remotely. We used items such as an anemometer (wind speed sensor), DHT11 (temperature sensor), MPX5100AP (pressure sensor), LDR (Light Dependent Resistor) for light intensity, a microcontroller, and a Memory Card (MMC). Finally, a weather monitoring system was created, which monitored five weather parameters.

The design consists of two major components: hardware and software implementation. The system overview includes information about the power supply, input and output components, and the controller. At the modular level of this project implementation, the microcontroller receives input data from several measurement devices, saves it, and displays it on the LCD.

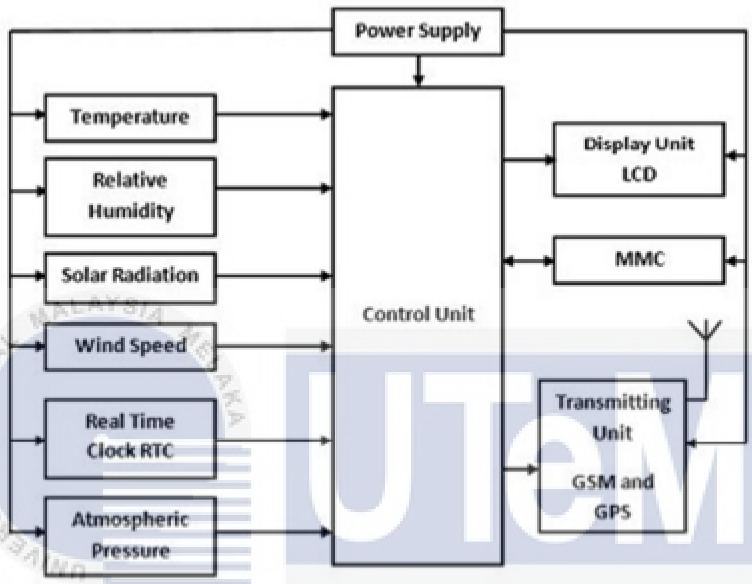


Figure 2.36: Block Diagram of System

Outside memory was associated with the microcontroller to give sufficient extra room to the got information after it had been handled. The GSM module, which was coupled to the microcontroller, got order signals for transferring the deliberate climate factors to remote location(s) by means of the web. Moreover, the microcontroller can be interacted with a PC (PC) to show the got information.

The built gadget was tested for wind speed, temperature and humidity. A selection of the recorded parameters has been summarized in the tables below. The % inaccuracy in measurement was calculated for each table. As a result, the minimal %

error obtained served as a clear indicator of the newly developed device's great accuracy.

Time (hrs)	NIMET value (m/s)	Constructed device (m/s)	Difference (m/s)
7:00am	2.75	2.60	0.03
8:00am	1.54	1.60	0.06
9:00am	2.06	2.10	0.04
10:00am	2.06	2.10	0.04
11:00am	3.09	3.10	0.01
12:00am	2.06	2.10	0.04
1:00pm	0.00	0.00	0.00
2:00pm	0.00	0.00	0.00
3:00pm	3.60	3.65	0.05
4:00pm	1.54	1.54	0.00
5:00pm	1.54	1.55	0.01
6:00pm	2.06	2.10	0.04
7:00pm	1.54	1.60	0.06
TOTAL	23.66m/s	24.04m/s	0.38m/s

Table 2.6: Recorded of Wind Speed Data

	NIMET Temperature Readings (°C)		New Device Temperature Readings (°C)		Difference (°C)		Percentage Error	
Time	Day 1	Day 2	Day 1	Day 2	Day 1	Day 2	Day 1	Day 2
7:00am	21.6	22.3	22.0	23.5	0.40	1.2	1.85%	5.38%
8:00am	24.5	24.1	25.0	25	0.50	0.9	2.04%	3.73%
9:00am	25.2	25.3	26.5	26.5	1.30	1.2	5.16%	4.74%
10:00am	26.4	26.5	27.5	27.5	1.10	1	4.17%	3.77%
11:00am	27.3	27.6	28.0	28.8	0.70	1.2	2.56%	4.35%
12:00am	28.2	28.1	29.0	29.5	0.80	1.4	2.84%	4.98%
1:00pm	27.7	28.8	28.0	29.9	0.30	1.1	1.08%	3.82%
2:00pm	26.7	29.8	27.7	29.9	1.00	0.1	3.75%	0.34%
3:00pm	28.6	30	29.5	31	0.90	1	3.15%	3.33%
4:00pm	28.4	28.5	29.5	29.5	1.10	1	3.87%	3.51%
5:00pm	28.5	21.5	29.5	22.5	1.00	1	3.51%	4.65%
6:00pm	22.8	26.2	23.5	27.0	0.70	0.8	3.07%	3.05%
7:00pm	21.9	24.5	23.0	25.0	1.10	0.5	5.02%	2.04%
				Average Percentage Error			3.24%	3.67%

Table 2.7: Recorded of Relative Temperature

Time	NIMET Readings %		New Device Readings %		Difference %		Percentage Error	
	Day 1	Day 2	Day 1	Day 2	Day 1	Day 2	Day 1	Day 2
7:00am	95	94	96	95	1	1	1.05%	1.06%
8:00am	95	88	95	88	0	0	0.00%	0.00%
9:00am	96	87	96	88	0	1	0.00%	1.15%
10:00am	96	83	97	84	1	1	1.04%	1.20%
11:00am	95	77	96	79	1	2	1.05%	2.60%
12:00am	89	71	89	71	0	0	0.00%	0.00%
1:00pm	94	76	95	76	1	0	1.06%	0.00%
2:00pm	88	65	89	66	1	1	1.14%	1.54%
3:00pm	74	65	75	65	1	0	1.35%	0.00%
4:00pm	73	62	74	62	1	0	1.37%	0.00%
5:00pm	68	62	69	63	1	1	1.47%	1.61%

Table 2.8: Recorded of Relative Humidity

The goal of this work was to provide a simple and cost-effective weather station, considering the importance of weather-related data availability. Five meteorological parameters should be measured: air temperature, relative humidity, wind speed, sun radiation, and atmospheric pressure. It can also save the data for a specific amount of time. Finally, the system has a peripheral device that allows it to send data to end users over the internet. The results of the tests conducted, as shown in tables 1-5 above, demonstrate that the project's goal and objectives were met.

This framework can be updated in the future to give prescient capacities in view of recently assembled and put away information. Satellite pictures, which incorporate Cloud Movement Vector (CMV), Water Fume Wind (WVW), Quantitative Precipitation Gauge (QPE), Ocean Surface Temperature (SST), and different information, could likewise be used to further develop weather conditions anticipating exactness. The framework might be altered to assist ranchers with deciding the best time for each yield and exact weather patterns for bug control.

2.21 Weather Monitoring System Using Wi-Fi

This paper(Parashar et al., 2015) describes a weather monitoring system that measures temperature, pressure, humidity, rainfall, wind speed, and direction. Various sensors sense all the parameters that must be monitored. Temperature, pressure, and humidity were monitored using widely accessible sensors, whereas wind speed, wind direction, and rainfall were measured using a rotary encoder, opto-coupler, and tipping bucket approach, respectively. The measured data is processed using a microcontroller-based system and wirelessly transmitted to the server for continuous storage and access. Because the system is completely automated, human error is minimized. This type of monitoring system can be utilized for data collecting and analysis in industries, agriculture, and metrological departments, among other places.

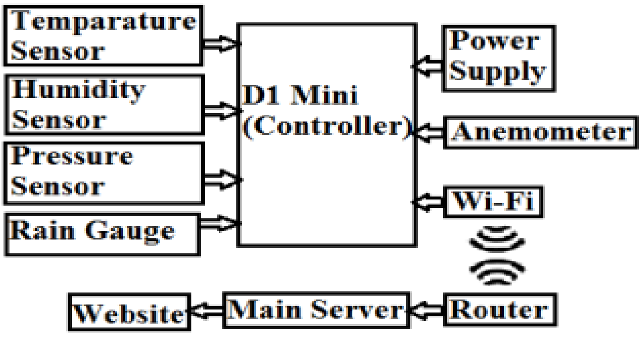


Figure 2.37: Block Diagram of The Proposed System

The controller receives all sensor outputs and processes them in the appropriate format. These are then wirelessly transferred to the main server. All of these sensed parameters' databases are kept on the main server and are routed and presented continually on the website at a refresh rate of 10 seconds. MySQL is used to create the database. To show the data on the website, a PHP script is built.

Table displays the measured calibrated readings of several sensors. Table also shows the visible parameters on the page.

ID 1	log	Temperature	Pressure	Humidity	Wind_Direction	Wind_Speed	Rainfall
1	2016-05-08 23:50:56	28	948.8	18	North Direction	0	0
2	2016-05-08 23:51:34	28	948.8	18	North Direction	0	0
3	2016-05-08 23:52:59	28	948.8	18	North East Direction	1.2	0.02
4	2016-05-08 23:53:09	28	948.8	18	North East Direction	1.2	0.02
5	2016-05-08 23:53:33	28	948.8	18	North East Direction	3.2	0.04
6	2016-05-08 23:54:24	28	948.8	18	East Direction	3.2	0.04
7	2016-05-08 23:54:54	28	948.8	18	East Direction	4.6	0.06
8	2016-05-08 23:55:44	28	948.8	18	South East Direction	5	0.08
9	2016-05-08 23:56:05	28	948.8	18	South East Direction	5	0.08
10	2016-05-08 23:56:23	28	948.8	18	South East Direction	6.7	0.08
11	2016-05-08 23:57:09	28	948.8	18	South Direction	3.2	0.1
12	2016-05-08 23:58:03	28	948.8	18	South West Direction	3.2	0.1
13	2016-05-08 23:58:22	28	948.8	18	West Direction	3.2	0.1
14	2016-05-08 23:59:06	28	948.8	18	North West Direction	2.2	0.12
15	2016-05-08 23:59:57	28	948.8	18	North Direction	5.6	0.14

Figure 2.38: Snapshot Of Measured Database Parameters

Temperature	28 degrees
Humidity	18%
Pressure	948.8mb
Wind direction	North Direction
Wind speed	5.6m/s
Rainfall	0.14mm

Table 2.9: Displayed Parameters On Website

The proposed approach can be used to accurately monitor weather parameters over a large area without using satellite photos. This system provides a low-cost, simple, and real-time option for monitoring meteorological conditions. The system aids in the reduction of human mistake. Because all of the parameters are displayed online, smart cities can be made available to the entire public. An IoT application can benefit from a Wi-Fi interface with embedded application.

The method can be changed to assist farmers in determining the ideal time for each crop and precise weather conditions for pest control.

2.22 Development of IoT Weather Monitoring System Based on Arduino and ESP8266 Wi-Fi Module

The study was done by (Holovatyy, 2021) is the equipment and programming of the IoT weather conditions observing framework have been made in this article. The IoT weather conditions observing framework's equipment comprises of a microcontroller gadget based on the Arduino Mega2560 board, a computerized strain, temperature, and moistness sensor BME280, a Wi-Fi module ESP-01 based on the ESP8266 chip, and a 162 alphanumeric LCD in view of the Hitachi HD44780 regulator. Proteus VSM was utilized to plan the electronic circuit and model of the microcontroller weather conditions observing device. The IoT weather conditions checking framework's activity calculation has been planned. The Internet of Things device measures meteorological factors such as atmospheric pressure, temperature, and relative humidity.

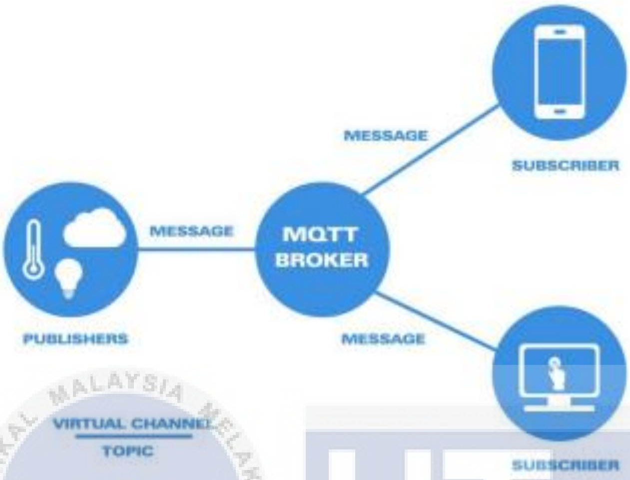


Figure 2.39: MQTT Publish-Subscribe Model

Weather factors such as temperature, relative humidity, and air pressure must be monitored by an IoT system. The constructed block diagram of the hardware of the IoT weather monitoring system is illustrated in Fig. 2.39. The IoT weather monitoring system's hardware is constructed on the Arduino Mega2560 board, which is based on the ATmega2560 microcontroller. The Arduino Mega2560 board is coupled to the BME280 digital pressure, temperature, and humidity sensor, ESP8266 Wi-Fi module, alphanumeric LCD module HD44780, and four buttons. The weather data from the BME280 sensor is read and processed by the Arduino ATmega2560 microcontroller. On the LCD, the device displays the current values of the meteorological parameters.

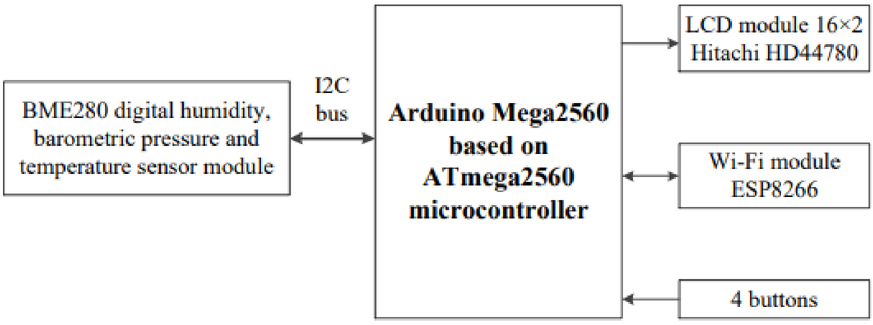


Figure 2.40: Block Diagram Of The Iot Weather Monitoring System

The equipment and programming of IoT weather conditions checking frameworks were made as a component of the work.The IoT weather conditions observing framework's equipment contains a microcontroller gadget that actions climate information like temperature, relative mugginess, and gaseous tension and sends them to the IoT stage for extra handling and representation.

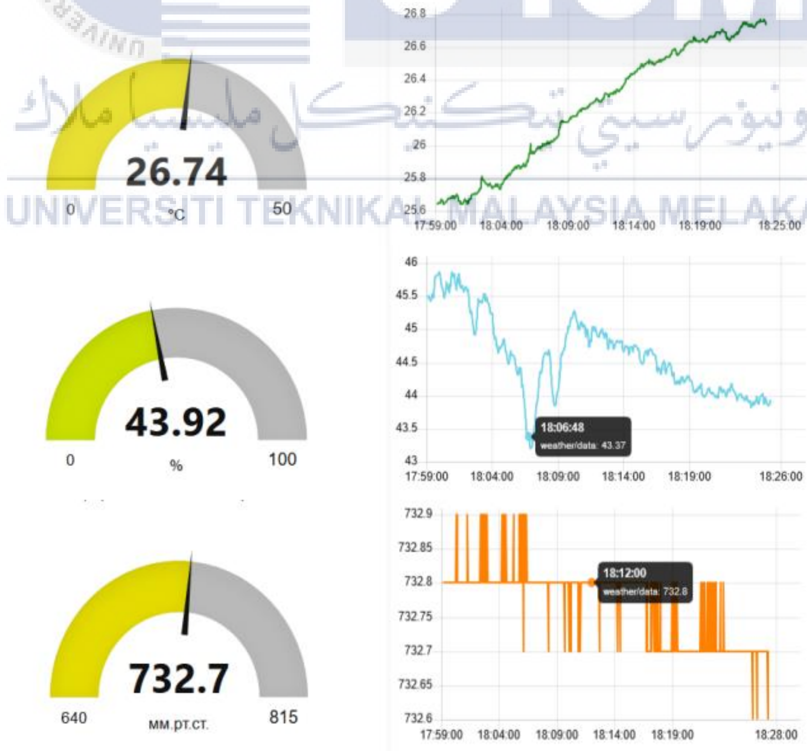


Figure 2.41: Iot Weather Data Visualization On The Node-RED Dashboard

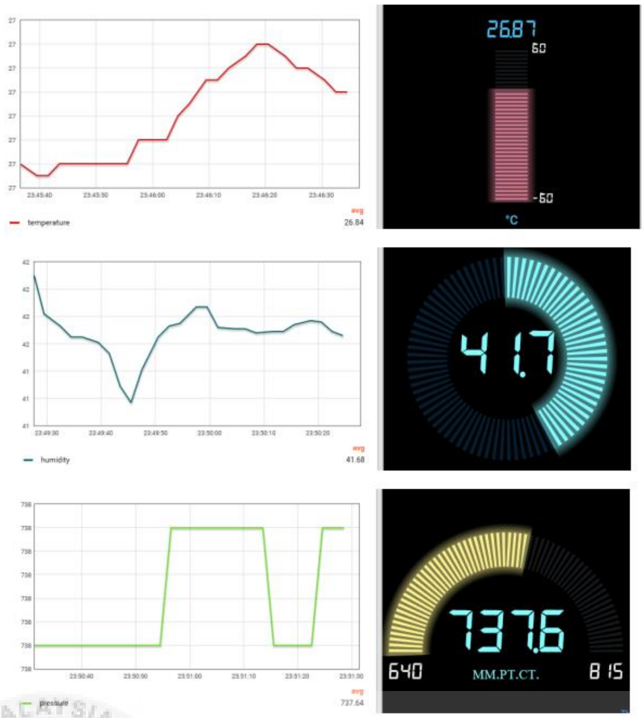


Figure 2.42: Iot Weather Data Visualization On Thingsboard Dashboard

The test circuit for the IoT weather monitoring gadget has been created and tested on the solderless breadboard. The modelling and testing results of the IoT weather monitoring system demonstrated that the designed hardware and software worked well.

2.23 Design and Analysis Performance of IoT-Based Water Quality Monitoring System using LoRa Technology

(Promput et al., 2023)is design about Water quality is one of the most critical factors influencing human life. In most cases, water quality measurements must be taken on-site. If the area to be explored is large, numerous test sites will be needed. Repeated water quality assessments will be complicated and time-consuming.

As a result, an ongoing water quality checking framework is expected to save and screen the water to execute proactive enemy of pollution activities. This task tends to the worries by creating ecological sensors that screen and show water quality utilizing LoRa innovation and the Hub RED application. According to the region requiring analysis, it is the measurement and collecting of data on water quality characteristics such as temperature, electric conductivity, pH, air quality, and turbidity.

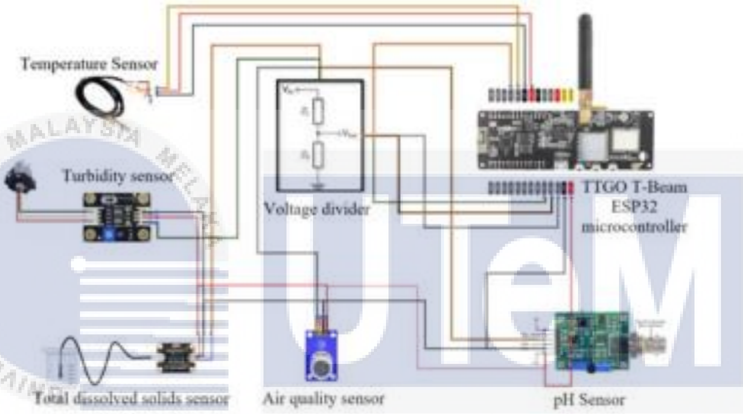


Figure 2.43: Diagram Of Water Quality Monitoring System

The essential concept underlying all of this research is the use of sensors to monitor water quality metrics and the transfer of that data via LoRa wireless communication. It consists of two components: a transmitter and a receiver.

Figure 2.44: Water Quality Monitoring Dashboard

Figure depicts the water quality parameters displayed on the Node-RED dashboard. Furthermore, Figure 8 depicts the mobile collector's precise location using Node-RED's world map library.

The examination goals were to plan and break down the presentation of a water quality checking framework in view of IoT and LoRa innovation involving the Hub RED application for plan and part engineering, as well as to assess how the water quality observing framework was carried out.

Usability testing was discovered to be more advantageous and time productive. This framework can be coordinated with state of the art innovations like brilliant urban areas. As a result, there will be a greater need for a real-time monitoring system. The BOD/COD sensor will be added to the system in the future, as well as a new antenna for LoRa technology to increase signal transmission.

2.24 Weather Monitoring System using Blynk Application

(Bin Zohari & Bin Johari, 2019) was done about this project, the objective of this project is to create a simple microcontroller, Arduino UNO, with a wireless weather monitoring system, a WiFi-WeMos ESP8266, that monitors weather conditions using three sensors: temperature, rain, and carbon monoxide. The data is then displayed in the Blynk programmed. The Arduino UNO Microcontroller, WiFi-WeMos ESP8266, DHT 11 temperature and mugginess sensor, downpour sensor, and Carbon MQ-7 monoxide sensor were utilized to make this task. It is perfect for weather conditions checking in any area and whenever.

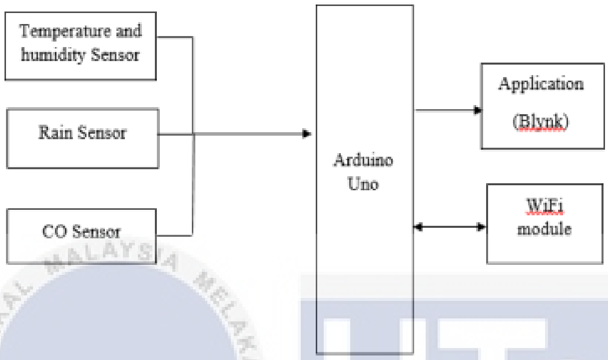


Figure 2.45: Block Diagram Of Weather Monitoring System Using Blynk

The major component in the system is the Arduino microcontroller, which is connected to all the other components as illustrated in Figure 1. The framework's sensors are connected to the Arduino microcontroller's simple info. The WiFi-WeMos ESP8266 is likewise associated with the Arduino. All data collected will be sent to the Blynk application.

Figure 2.46: Prototype Of Weather Monitoring System

Figure 2.47: Interface of Blynk Application

This section will describe briefly and demonstrate the outcome of the implemented circuit. All of the hardware equipment is being assembled, and various types of data in this project are being examined to guarantee that the system performance is stable and in the intended condition. The project's outcomes are examined in the analysis section. It can assist users in analysing system performance and checking system function based on the system's results and data analysis. As a result, it can monitor the weather in the outdoor area.

The gadget gives a low-power choice to climate and ecological checking. The checking framework has been tried in an external climate and has effectively refreshed sensor information. The data will be utilized for different kinds of investigations and might be imparted to others or clients.

The technology has the potential to be used for monitoring developing cities and industrial zones, particularly for monitoring pollution. To protect public health from pollution, the system must also be capable of providing an efficient and low-cost solution to the authorities. It is also suited for continuous environmental monitoring in the future.

2.25 A Low-power Real-time Pollution Monitoring System Using ESP LoRa

The purpose of this research by (Atiwanwong & Hongprasit, n.d.)is to look into the air quality in the Thai city of Khon Kaen. We designed and built four sensors (PM2.5-PM10, pressure, altitude, temperature, humidity, and carbon dioxide) that will function on a Wireless Sensor Network (WSN) using LoRa technology. The experimental data for air quality have been displayed on a web application. This paper's proposal can enable consumers learn about historical and real-time air quality data using a web browser.

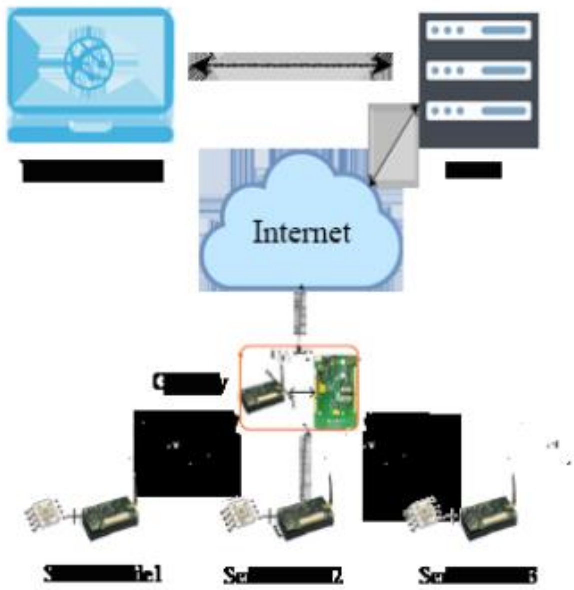


Figure 2.48: The System Architecture

The suggested system is divided into two parts: software and hardware. The first is software, which includes a software platform based on the browser/server framework (see Fig. 1). The second is hardware, which includes sensor nodes and microcontrollers. The microcontroller will receive and transform all data from sensors into digital signals, which will subsequently be transferred to the gateway via the LoRaWAN protocol. The gateway will then use Wi-Fi connectivity to deliver this data to the database.

Figure 2.49: Component of Air Quality Sensors

The sensors picked in light of execution rules like exactness and low power utilization. Plan tower PMS3003 for location of PM10 and PM2.5, MQ-135 for identification of carbon dioxide (CO2), DHT22 for recognition of temperature and moistness, and BMP180 for discovery of strain and elevation are the four brilliant sensors we use. Figure 2.49 Air quality sensors devices depicts smart sensor air pollution monitoring.

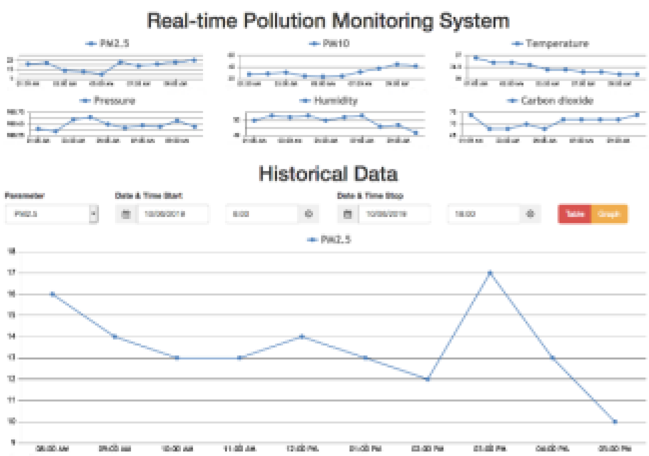


Figure 2.50: Real-Time Pollution Monitoring System

To start, when the sensor hub gives information to the server, the information is gotten and saved in the MySQL data set. Second, the site page was created and constructed utilizing HTML and JavaScript. This site page's responsibility is to inquiry and show continuous contamination information as charts and information tables on the page.

This examination proposed a low-power, continuous contamination observing framework in view of ESP LoRa for researching air poisons and giving information by means of a web application. LoRa can ship information over huge spans with low power use; the power supply is filled by a battery that is charged by sunlight-based cells.

As a result, air pollution data will be stored for a long time in the future, and data analysis methods will be used to determine the air pollution ratio and predict the future.

2.26 Development of air temperature and soil moisture monitoring systems with LoRa technology

In this study create by (Purnomo et al., 2021)a machine-to-machine method using LoRa technology was used to transmit air and soil temperature humidity data. This method is useful because communication can be done at with a frequency of 433 MHz, so it does not depend on mobile phone frequencies. Look Internet can be made by connecting the LoRa to the ESP32 microcontroller that is connected access point for sending data online through the Ceyene application. The test environment is built giving changes in air temperature and the depth of the implanted sensor in the ground. Research shows that soil temperature and moisture can be monitored With LoRa technology at an affordable price.

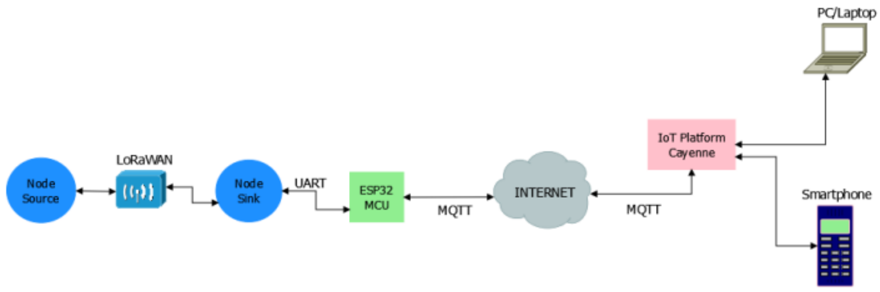


Figure 2.51: Diagram of System Design

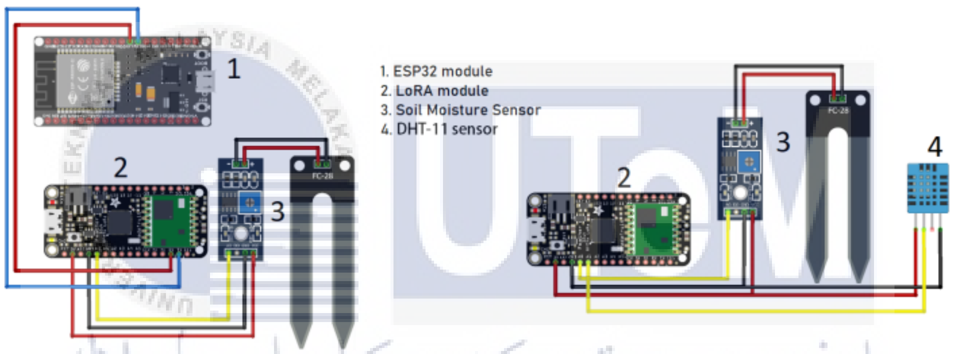


Figure 2.52: Configuration Setup For DHT-11 And Soil Moisture Sensor

The figure shows the block diagram of the system. Soil moisture, temperature and humidity is measured by sensors. The sensor is utilized to quantify temperature and stickiness is DHT-11 and soil dampness sensor is FC-28 soil dampness sensor. The design of the DHT-11 sensor and the dirt dampness sensor are displayed in the figure. Subtleties then sent by LoRa. Sink Hub gets information from source hub. Every one of the information is there handled by the ESP32 microcontroller and shipped off the Cayenne IoT stage for recording and perception.

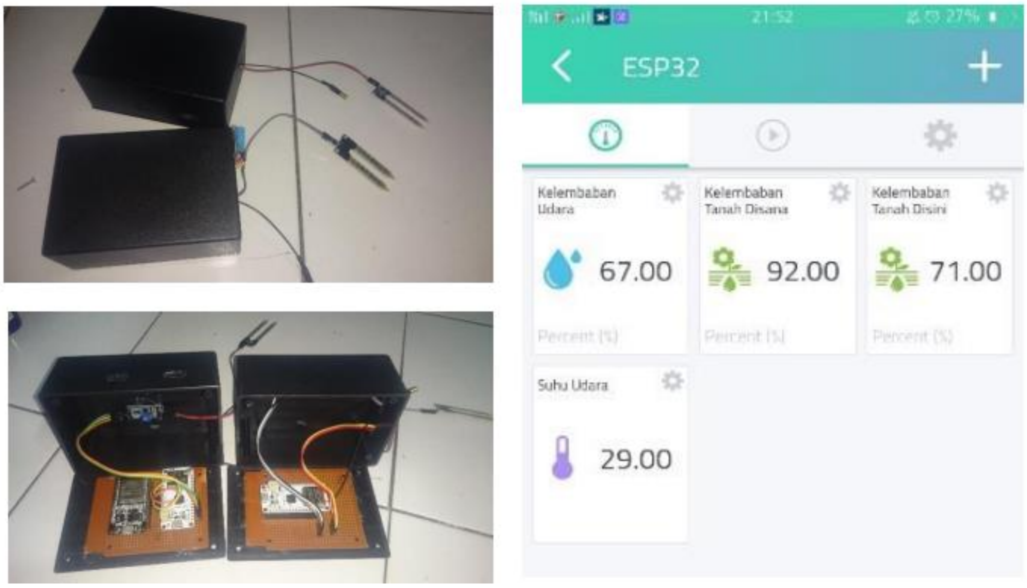


Figure 2.53: Setting And Display Interface In Mobile Application

To evaluate the performance of the proposed system, the system was operated and is used in a setting describing a typical agricultural field. The DHT-11 sensor measures air temperature in degrees Celsius (°C) and humidity in percent (%). Used soil moisture sensor measure soil moisture in percent (%).

Monitoring over the Internet can be done by connecting LoRa to an ESP32 microcontroller base station to send data online with the Ceyene application. The test equipment is made provides variations in air temperature and variations in the depth of the sensor planted in the ground. Research shows that soil temperature and moisture can be monitored using LoRa technology at an affordable price.

2.27 Hardware Specification and Comparison

2.27.1 Microcontroller

The Long-Range Alliance, or LoRa, is making waves these days. New wireless technology that can work at a long range while consuming a low amount of power. A battery may last for a considerable amount of time in some cases. LoRa, or Low-power Radio Access Technology (LoRa), is based on the Chirp Spread Spectrum (CSS) technology. Transmissions modulated with LoRa can be received over a very wide range of distances and are robust against disturbances. As the de facto wireless platform for the Internet of Things, LoRa is the de facto Internet of Things platform. Sensor data can be transmitted in real-time via Semtech's LoRa chipset, enabling data analytics and real-time collaboration to enhance productivity and efficiency.

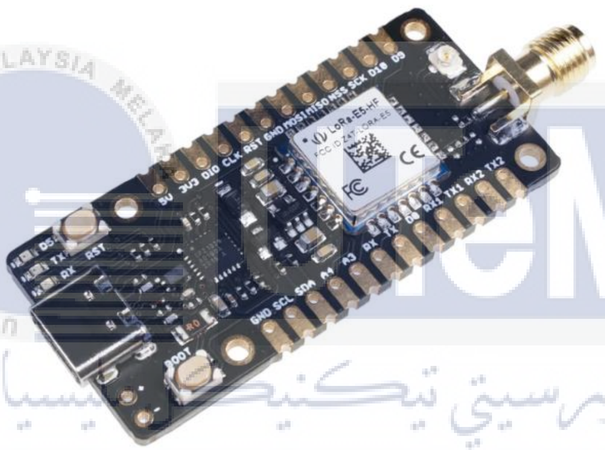


Figure 2.54: LoRa E5 Mini

In just one board size, LoRa-E5 mini provides a compact, lightweight solution for rapid prototyping, testing, and building your ideal IoT device with long-range, wireless Lora WAN capability. LoRa-E5 Development Board provides more advanced interfaces and features to unlock the more powerful performance of the LoRa-E5 module; it is an alternative to the LoRa-E5 mini. It can provide a wider range of access protocols and a variety of interface types that are beyond compare.

With the Grove interfaces and rich GPIOs, you can test and prototype the module quickly and efficiently, so that you can quickly bring the module to market.

Figure 2.55: Cytron LoRa- RFM

With the RFM LoRa Shield, you can integrate the RFM95W LoRa module into any Arduino project using the Open-Source Library. The Arduino Mega2560 is compatible with Arduino Uno, CT Uno, Maker Uno, and perhaps other pin-compatible boards. Data can be sent using RFM Lora Shield over very long distances at a low data rate (< 50kbps). A professional wireless sensor network application such as irrigation systems, smart metering, smart cities, smartphone detection, building automation, and smartphone detection can be achieved with this device, which offers ultra-long range, high interference immunity, and minimal current consumption.

Figure 2.56: Atilze LoRa Shield

The Atilze LoRa Shield integrates with any Arduino project using Gemtek GL6509 Low-Power Long Range LoRa Technology Transceiver Module and Open-Source Library is used for it. Possibly other boards with pins compatible with CT-UNO, Arduino Uno, Arduino Due, Arduino Mega2560, Arduino Leonardo are also compatible with the circuit board. LoRa shield has stackable headers, allowing it to be stacked with more Arduino-compatible shields.

Digital pins on this shield can be used to communicate with the GL6509 via software and hardware UARTs, or even to connect the PC to the board. Wireless data transmission is easy with low-power LoRa shields onboard the GL6509 module. A shortcut to market can be achieved by using AT commands (advanced command interface). According to LoRaWAN Class A protocol specifications, the GL6509 module conforms. Long-range solution integrated with RF, baseband controller, and application programming interface (API) processor.

Board / Specification	LoRa E5 mini	Cytron LoRa	Atilze LoRa
Modulation	LoRa, (G)FSK, (G)MSK, BPSK	FSK, (G)FSK, MSK, (G)MSK, LoRa, OOK	FSK, (G)FSK, LoRa.
Frequency	868MHz/ 915MHz	915MHz	919MHz– 923MH
Sensitivity	-116.5dBm ~ -136dBm	Down to -146dBm	As low as -137.5dBm
Cost	± RM 97.46	± RM 99.00	± RM 58.80

Table 2.10: Comparison LoRa

Cytron Lora Shield will be chosen because Even though Atilze LoRa Shiels is cheaper than Cytron Lora, Cytron Lora is a very good board. Atilze lora shield and LoRa E5 mini provide less modulation than Cytron LoRa, although it has good features. WMBus and IEEE802 also support a variety of high-performance (G)FSK modes. It consumes significantly less power than its competitors while offering superior phase noise, selectivity, receiver linearity, and IIP3 measurements.

2.27.2 Temperature and Humidity

Figure 2.57: Temperature And Humidity Sensor-DHT11

Voltage	3.5V to 5.5V
Current	0.3mA(measuring) 60uA(standby)
Output	Serial Data
Temperature Range	0 ˚C to 50˚C
Humidity Range	20% to 90%
Resolution	Temperature & Humidity both are 16-bit
Accuracy	±1 ˚C & ±1%

Table 2.11: Specification of DHT11 Sensor

With the DHT11 Temperature and Humidity Sensor, a digital signal output is calibrated. A unique digital-signal-acquisition technology and temperature and humidity sensing technology make this system extremely reliable and stable over an extended period. Besides offering excellent quality, it has fast response times, anti-interference capabilities, and is cost-effective, by combining a NTC temperature measurement component with a resistive-type humidity measurement component.

2.27.3 Rain Sensor

Figure 2.58: Raindrop Sensor

Using the rain sensor module is simple and allows rain drops to be detected when they fall through the raining board and the intensity of the rainfall can be measured. In addition to a power indicator LED and a potentiometer that can be adjusted to adjust sensitivity, it is equipped with a rain board and a control board that are separated for convenience.

Analog output can be used to detect a drop in rainfall amount when there is a drop in rainfall amount. An LED will turn on when connected to a 5V power supply whenever the water droplet does not land on the induction board and the DO output is high. A switch indicator will illuminate when a very small amount of water is dropped, when the DO output is low, and when the DO output is low. The device produces high levels of output after removing the water droplets and restoring it to its original state.

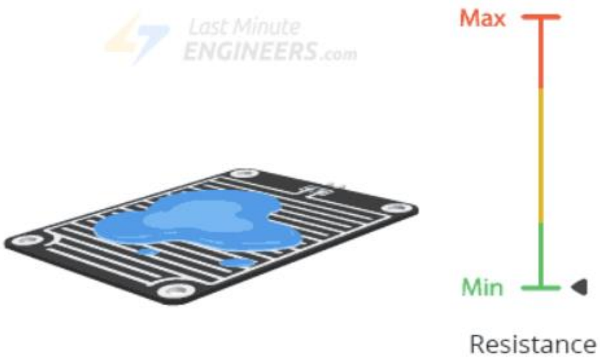


Figure 2.59: Raindrop Pad Sensor Detect The Rain

This sensor is a semiconductor device that contains an exposed copper trace which contains a sensor pad that is installed out in the open, possibly over the roof or out in the open where it may encounter rainwater. There is a tendency for these traces to not be connected but instead they are bridged by water.

2.27.4 Wind Sensor

Figure 2.60: Wind Sensor

Output	0.4 to 2V
Testing Range	0.5 to 50 m/s
Start wind speed	0.2 m/s
Resolution	0.1 m/s
Accuracy	Worst case 1 meter/s
Max wind speed	70 m/s
Connect details	Pin 1 – Power (brow wire), Pin 2 – Ground (black wire), Pin 3 – (Blue wire), Pin 4 connected

Table 2.12: Specification of Anemometer Sensor

An instrument that measures wind speed is known as a wind speed sensor. In order to calculate wind speed, the sensor elements on the top three wind cups are rotated by the wind generated by the airflow, and the central axis provides the motion for the internal sensing element to produce an output signal, which can be used to calculate wind speed. There is a physical device that is used to measure and transmit the directions of the wind, known as the wind direction sensor. By rotating a wind vane arrow, information about the direction of the wind flying is transmitted to the coaxial encoder board and the result is output along with the measurement information.

2.27.5 Noise Sensor

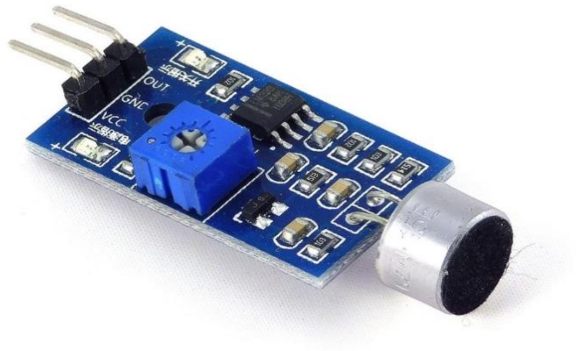


Figure 2.61: Noise Sensor

Operating Voltage	3.3 to 5 V
Operating Current	4~5mA
Voltage Gain	V=6V, f=1kHz
Sensitivity of Microphone	1kHz = 52 to 48 dB
Impedance Of Microphone	2.2kΩ
Frequency of Microphone	16 to 20kHz
Signal to Noise Ratio	54dB

Table 2.13: Specification of Sound Sensor

Decibels (dB) are used to measure the level of noise. There is a direct correlation between the loudness of the noise and the decibel level. A decibel level can be adjusted so that it corresponds to a human's ear. It is therefore understood that the level of noise can be described in decibels A (dBA). When a person is exposed to noise, the effects of the noise will vary, depending on the level of noise.

Hearing loss can occur when long-term exposure to loud noises (75 dBA for more than eight hours a day for years) leads to hearing loss as we age. There is also a possibility that the body reacts to lower noise levels. For example, an outdoor noise level of 40 decibels can cause sleep disturbances.

2.27.6 Pressure Sensor

A barometric sensor is helpful for determining weather conditions.This sensor is perfect for precision pressure measurements of up to 1 hPa and temperature readings of up to 1.0 ° C due to its high accuracy and low cost. You may use this sensor and the altimeter with 1 metre accuracy because pressure fluctuates with altitude and pressure readings are quite accurate.

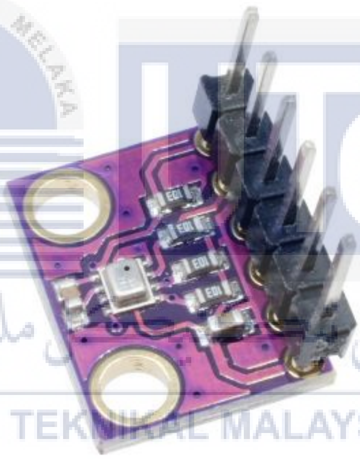


Figure 2.62: Pressure Sensor(BMP180)

Pin	Detailed
VCC	+3.3V Connected
Ground	Ground Connected
SDA	The Serial Data Pin is used for I2C and SPI communication.

SCL	I2C and SPI communication require a serial clock pin.
CSB	Choose a chip. SPI communication is also supported.
SDO	Used to set the device address as well as for SPI communication.

Table 2.14: Pin Detail of BMP280

Biometric or atmospheric pressure is measured by the BMP280 sensor. Based on air weight, the air pressure sensor works very simply. There is a specific pressure for each weight in the air around us.

Voltage Operation	1.7V-3.6V
Pin Pitch	2.54mm
Pressure operate	300hPa – 1100hPa
Termpareture Operation	-40˚C - +85˚C

Table 2.15: Specification of Pressure Sensor(BMP280

2.28 Software Comparison

2.28.1 Thingspeak, Ubidots and Blynk Applications

Blynk, Thingspeak, Ubidots, and Ubidots are among platforms that may be used in this project. The ThingSpeak app could collect, visualize, and analyse live data streams through the cloud. Data can be transferred from your device to ThingSpeakTM using online services such as Twitter® and Twilio®, and visualisations of live data can be built immediately.

The Ubidots' device is a virtual representation of a data source or simply a sensor asset that enables data to be sent from the sensor to the cloud via a connection protocol. Data can be sent to the cloud from any Internet-enabled device, and actions and alerts can be triggered based on that data.

Blynk is a low-code IoT software platform that lets you manage thousands of users and devices, connect them to the cloud, and build mobile apps for remote monitoring and control. With Blynk, you can control Arduino, Raspberry Pi, and NodeMCU over the Internet with iOS or Android smartphones. By compiling and providing the appropriate address on the available widgets, this application creates a graphical interface or human machine interface (HMI).

Platform / Features	Thingspeak	Ubidots	Blynk
Period of Store data	1 year	2 years	No time
Platform supported	Web	Web	Web Android Iphone/ipad
Processing speed	As fast as every second	1 second per token	1 min per 60 values
Limit data	cannot exceed 3000 bytes	10,000 bytes	10,000 data per day
Is platform free to use?	Yes. Its free service	No. This apps have price every plan and have limit 3 devices for free.	No. This apps have price every plan and have limit for free.

Table 2.16: Comparison of Platform

The platform comparison has been shown in the Table 2.2 below. Blynk is the chosen platform for this project. This is because Blynk is one of the platforms that can be monitor from mobile phone and easy used.

2.29 Summary

Throughout chapter 2, the journal reviews that have already been done are reviewed, as well as a couple of connections to the title to be created. This chapter has described several of the devices that will be used in this project, as well as conducted a shield comparison in order to perform the actual shield main calculation needed to determine which shield main is required for this project.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology of this project is discussed in this chapter. It is divided into several parts to ensure that the objective is achieved. The proposed system consists of several main components, namely Cytron Lora Shield, Arduino, Barometric pressure sensor, anemometer, rain sensor pad, temperature and humidity sensor, and noise sensor. Flow charts, block diagrams, hardware and weather monitoring systems are presented in this chapter.

3.2 Project Workflow

The workflow project shown in the flowchart is the same as the one in the diagram below. It is suggested that you start by selecting or discussing the title of the project, decide on that title and then find some journals related to the title to write about and compare in your literature review. Finish the problem statement, the project objectives, the scope of the project, and the expected results according to the time schedule given in chapter 1. It is necessary to explain the methodology in relation to flow charts, block diagrams, and system design in the next chapter.

Software and hardware are designed according to the system design, then tested. The data collection can be continued if it proves successful, and comparisons can be made with other data collections. It is possible that the connection, software, or both need to be redone or corrected until the problem is resolved. Once this project is complete, it will be time to write conclusions and recommendations for this project, and that will be the final step in the process.

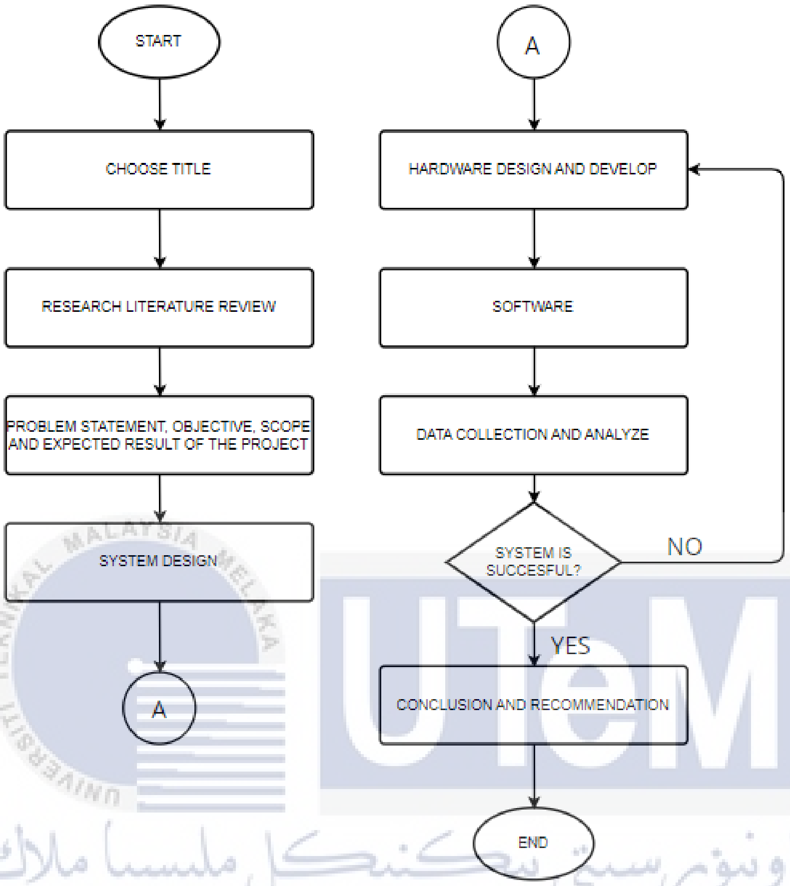


Figure 3.1:Flowchart of Project Workflow

3.3 Project System Architecture

The transmitter is Dht11 (temperature and humidity sensor), anemometer (wind speed sensor), noise sensor, rain volume meter, barometric pressure sensor and raindrop sensor that needs to be connected to the Cytron LoRa shield to detect all the parameters. Receiver is the recipient of data that will be sent to the user through the Blynk application.

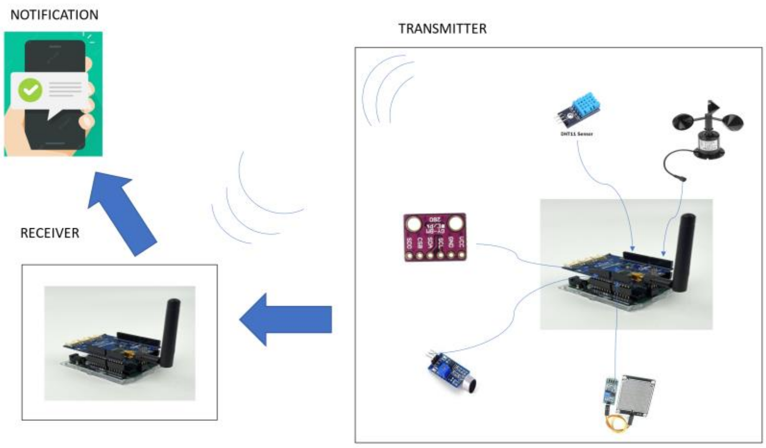


Figure 3.2: System Architecture

The operation flow of the proposed system is shown using a flow chart as shown in the figure. Inter Cytron LoRa with several sensors ready to detect the required parameters. The LoRa transmitter will transmit to the LoRa receiver after all sensor parameters are detected. Notifications will be sent to users through the Blynk application. Users will get notifications from Blynk to monitor the weather they want to know.

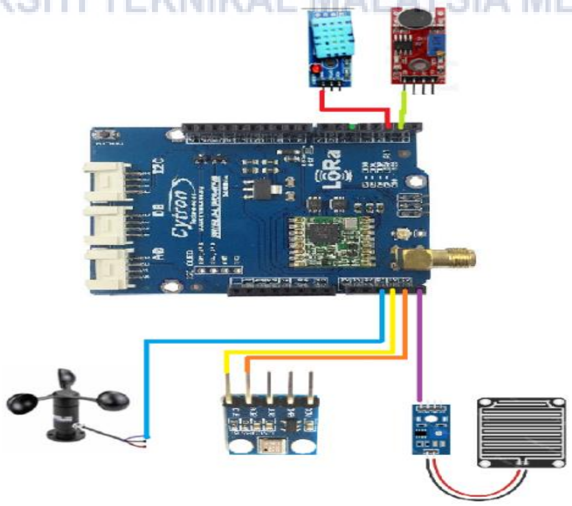


Figure 3.3: Circuit Diagram

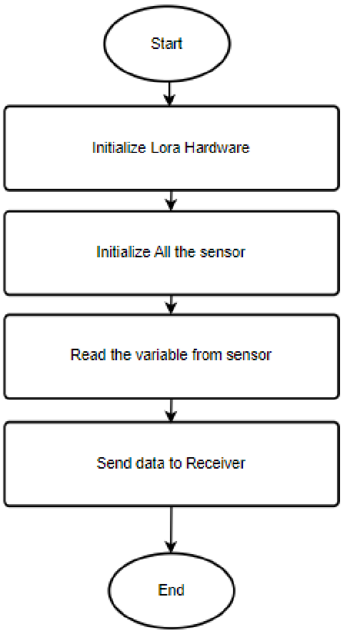


Figure 3.4: Flowchart of Transmitter

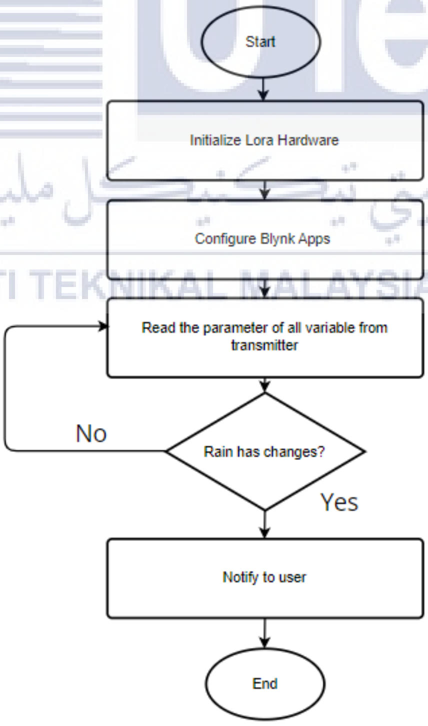


Figure 3.5: Flowchart of Receiver

3.4 Hardware Methodology

3.4.1 DHT 11

This sensor has a dedicated NTC that is used to measure the temperature and humidity in it, and is a common temperature and humidity sensor that is used often. 8-bit microcontroller to output temperature and humidity values as serial data.

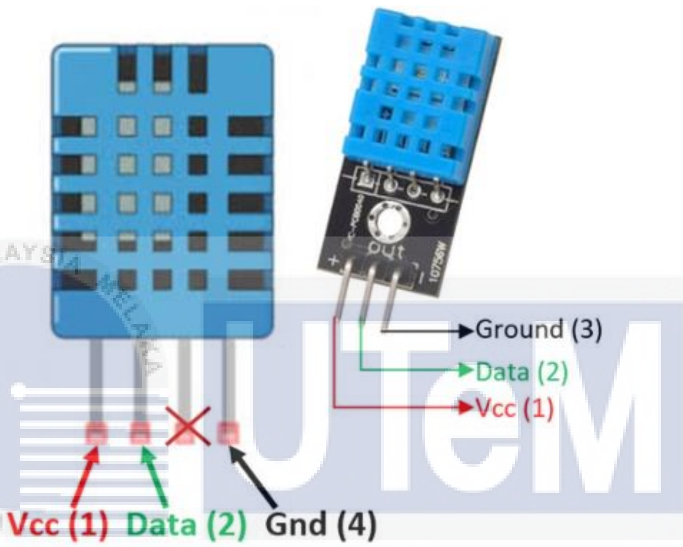


Figure 3.6 : DHT11 Sensor

The DHT11 sensor consists of a capacitive humidity measurement element and a temperature sensing thermistor. A moisture-retaining substrate acts as a dielectric between the humidity sensor capacitor's two electrodes. As the humidity level fluctuates, so does the capacitance value. The integrated circuit measures, processes, and transforms resistance measurements into digital form.

Using a thermistor with a Negative Temperature coefficient, this sensor detects temperature by decreasing its resistance value as the temperature rises. This sensor is frequently made of semiconductor ceramics or polymers in order to achieve better resistance values despite modest temperature changes.

DHT11 communicates over a single cable. This pin is logic one or logic zero depending on the voltage levels with certain time values. There are three steps to the communication procedure: first, a request is sent to the DHT11 sensor, followed by a response pulse from the sensor, and finally, a four-bit transmission is made to the microcontroller.

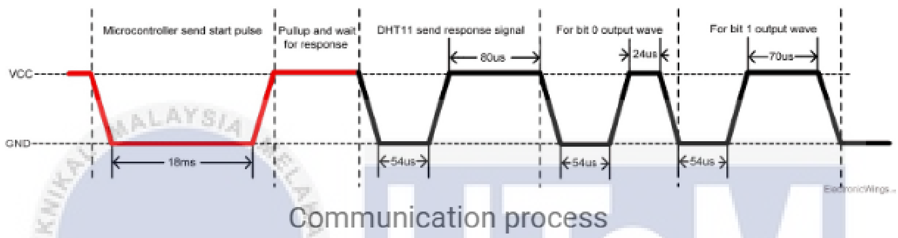


Figure 3.7: Communication Process of DHT11

To begin communication with DHT11, we must first transmit the start pulse to the DHT11 sensor. To give a start pulse, pull down (low) the data pin for at least 18ms and then pull up, as illustrated in the diagram.

In response to a start pulse, the DHT11 sensor emits a return pulse to indicate that it has received a start pulse. 54us of reaction pulse are low, followed by 80us of high pulse.

Following the response pulse, the DHT11 sensor sends data that comprises humidity, temperature, and a checksum. It consists of 5 segments (bytes) of 8 bits each, totaling 40 bits in total. Humidity readings in decimal integer form are included in the first two segments of these five parts. Relative Percentage Humidity is represented by this quantity. There are eight integers in the first eight bits, followed by eight fractions in the next eight bits. In the following sections, the temperature is expressed as a decimal integer. Temperature in degrees Celsius is indicated by this number.

The checksum consists of the checksums of the four previous segments. When humidity and temperature are equal, the checksum byte is a direct sum. By looking at the checksum value, we can check whether the value matches. Data received with an error contains a difference between them. Data received by the DHT11 pin is converted to low power until the next start pulse is received. DHT11 sensor delivers a 54us low level after delivering 40-bit data and then goes high. DHT11 enters sleep mode after this.

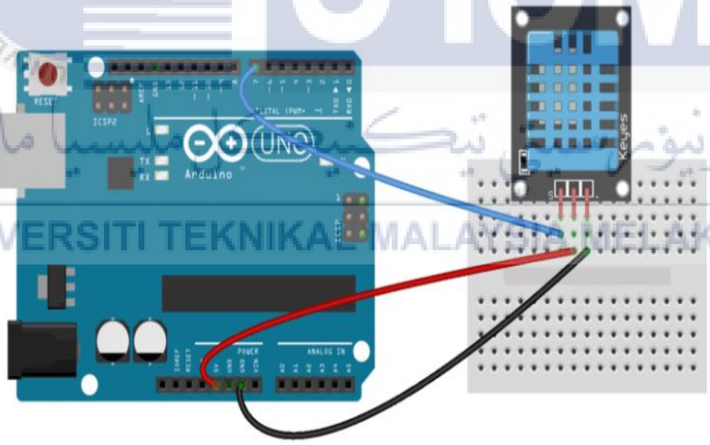


Figure 3.8: DHT11 Connection With Arduino

3.4.2 Bmp280 Barometric Pressure or Atmospheric Pressure

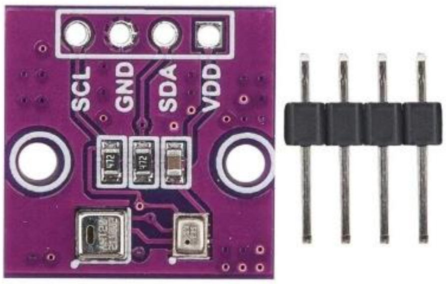


Figure 3.9: BMP280 Board

The BMP280 is an outright barometric strain sensor that is great for portable applications. As a result of its little size and low power utilization, it tends to be utilized in battery-fueled gadgets like as cell phones, GPS modules, and watches. The BMP280 is based on Bosch's laid out piezo-resistive strain sensor innovation, which offers high exactness and linearity, as well as long haul security and EMC strength. Various gadget activity choices guarantee greatest flexibility.

The device's power consumption, resolution, and filter performance have all been optimized. The BMP280 is a two-in-one temperature and pressure sensor module that interfaces readily with microcontrollers. It is an improvement over Bosch Sensortech's BMP180. Because the module comes in a breakout board style, investigations on a breadboard can be completed more quickly. The BMP280's pinout is shown in the image below:

Figure 3.10: BMP280 Pinout

In the BMP280, the CSB pin configures the communication interface to I2C or SPI. When the CSB pin is pulled high, the module uses I2C to communicate. When the pin is low, SPI is used to communicate. By default, a pull-up resistor in the BMP280 module pushes the CSB pin high. If we wish to use I2C for connectivity, we can leave this pin unconnected. In both communication interfaces, the BMP280 exclusively serves as a slave device.

The BMP280 has a 7-bit device address that may be set using the SDO pin. The device has an address of 1110110 (0x76) when the pin is connected to ground. This is the default mode of the BMP280 module, which is controlled by an inbuilt resistor. The BMP280 can quantify temperature and pneumatic stress, but the BME280 can gauge moistness as well as temperature and air pressure.

3.4.3 Anemometer Sensor

A wind anemometer allows a person to measure the velocity or speed of gases flowing in a confined space, such as airflow in a duct, or freely flowing in an open space, such as atmospheric wind, by using the results from a sensor. An anemometer detects changes in some physical attribute of the fluid or the influence of the fluid on a mechanical device put into the flow to calculate velocity.

Figure 3.11: Anemometer Sensor

Anemometers typically have three or four cups attached to horizontal arms. There is a vertical rod linking the arms together, which serves as a connection between them. It is during windy conditions that the cups rotate, pushing the rod to spin when the wind blows. A rod spins faster if the wind is stronger, and it spins faster the harder the wind is. Wind speed is calculated using the rotations or revolutions registered by the anemometer. The reason why wind speed is averaged over a short period of time is because wind speed varies from day to day, there are gusts and lulls, and so on.

By removing the knot, the sensor will appear as in the diagram. The internal circuit includes a photovoltaic module, a current generator, and an industrial processor.

Figure 3.12: Internal Circuitry of The Anemometer Sensor

The circuit PCB is made of military-grade A material. This assures the parameters' stability and the electrical characteristics' quality. The interior electronic components are all industrial chips with excellent electromagnetic interference resistance. The internal system can operate normally in temperatures ranging from -20°C to +50°C, with humidity levels ranging from 35% to 85%.

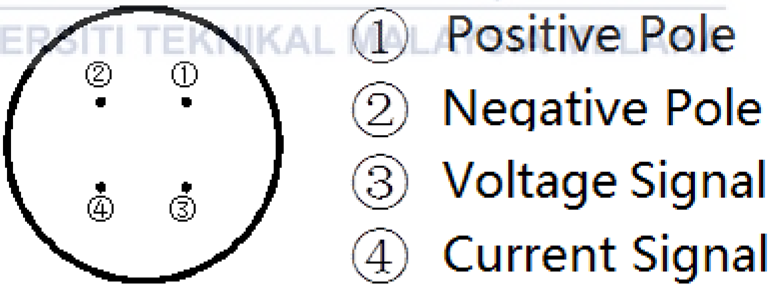


Figure 3.13: Military Plug

The cable's plug is a military plug. It has a good anticorrosive and anti-erosive performance, so the instrument may be used for a long period.

3.4.4 Raindrop Sensor

Raindrop Sensor is a gadget that distinguishes downpour. It comprises of two modules: a downpour board that distinguishes downpour and a control module that looks at and changes simple qualities over completely to computerized values. Raindrop sensors are utilized in the car area to consequently change wipers, farming to distinguish downpour, and home robotization frameworks to recognize downpour.

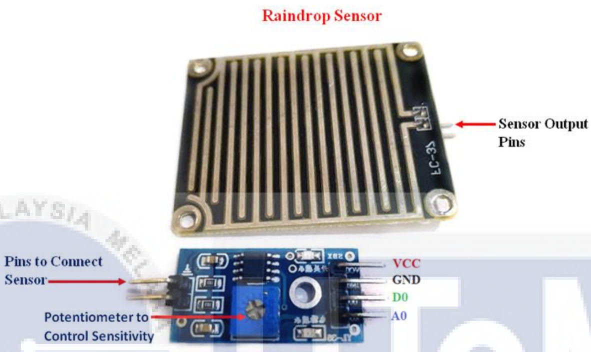


Figure 3.14: Raindrop Sensor

It is straightforward to connect the raindrop sensor to a microcontroller such as an 8051, Arduino, or PIC. The rainboard module and the raindrop sensor's control module are connected by a waterproof cable that can be seen in the figure below.

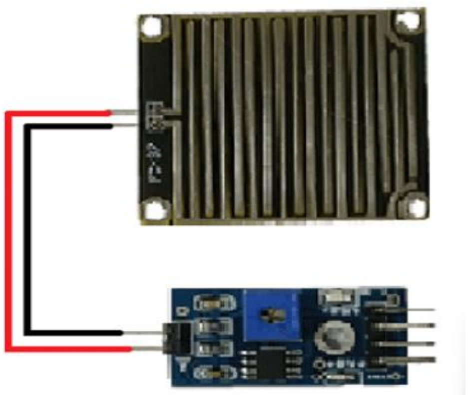


Figure 3.15: Raindrop Pad Connection Board

A raindrop sensor is controlled by a control module that has four outputs. To supply power to the VCC, a 5V power source is required. As you can see in the picture above, it is connected to the ground through its ground pin. Analogue output can be achieved by connecting D0 pin to the microcontroller's analogue pin, or digital output can be achieved by connecting D0 pin to the microcontroller's digital pin. Using an analog output of a microcontroller requires connecting the A0 pin to its ADC pin. An ADC converter isn't necessary with six ADC pins on the Arduino.

Sensor module components include LEDs, capacitors, resistors, and a potentiometer. Above is a pinout diagram showing the control module's components. During the configuration of the rain board module, copper rails serve as variable resistors. Having a lot of moisture on a rain board makes it more resistant to moisture. A circuit schematic of a raindrop sensing module is shown below.

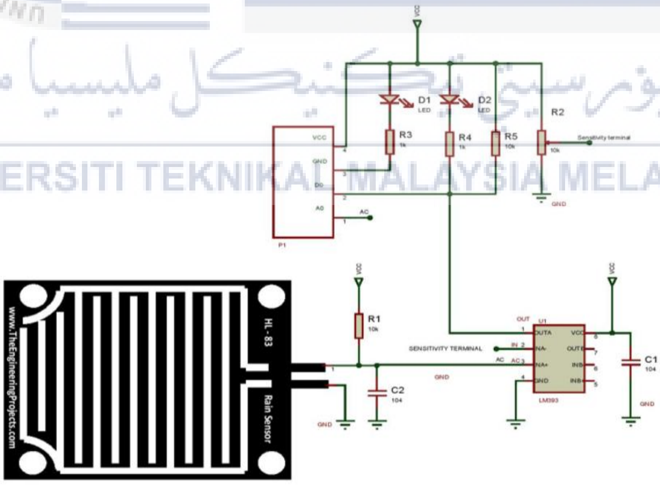
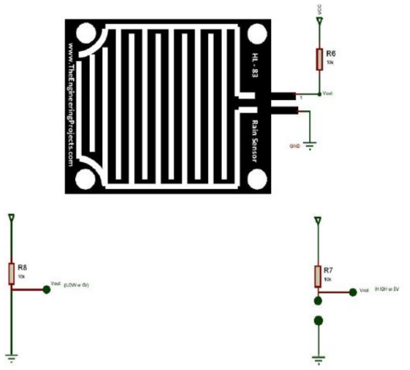


Figure 3.16: Raindrop Sensor Circuit Diagram

A voltage divider will be formed by R1 and the rain board module, as shown above. During biasing, capacitors C2 and C1 are used. Rain board module and R1 connect to the non-inverting terminal. The control module's A0 terminal is linked to another point derived from this connection. A potentiometer (R2) provides an inverting terminal on the LM393. A voltage divider, R2, can be used to change the sensitivity of the control module. R3 and R4 function as current limiters as well as pull-up resistors when the bus is idle.



Condition 1: During Wet   Condition 2: During Dry

Figure 3.17: Circuit of Two Conditions

There is a difference between the inputs of the inverting and non-inverting terminals, which is known as condition 1. In Condition 2, the inverting terminal's input must be less than the non-inverting terminal's input.

By altering the potentiometer, the input to the inverting terminal is set to a certain value, and the sensitivity is set. Rain board modules become moist if exposed to precipitation and give little resistance to supply voltage. The LM393 Op-Amp's non-inverting terminal therefore appears at the minimum voltage. By comparing terminal voltages that are inverting and non-inverting, the comparator determines whether they are inverting or not. An Op-Amp will display digital LOW if the condition is Condition (1). Digital HIGH will be the output if the scenario falls under Condition (2). Above is a graphic showing the comparable circuit for both scenarios.

3.4.5 Sound Sensor Module

Sound sensors are one type of module used to detect sound. The sound intensity is often measured using this module. Switching, security, and monitoring are the primary uses of this module. It is possible to adjust the accuracy of this sensor for ease of use. Sensors use microphones to provide input to buffers, peak detectors, and amplifiers. Microcontrollers receive voltage signals from this sensor when sound is detected. Then it goes on to perform the necessary processing as a result of that.

Figure 3.18: Sound Sensor

Pin 1 (VCC): 3.3 to 5V DC

Pin 2 (GND): Ground Pin

Pin 3 (DO): Output Pin

The operation of this sensor is similar to that of a human ear. Having a diaphragm in the human eye is a very important feature, as it is responsible for converting vibrations into signals that can be read by the brain. A microphone is used in this sensor, on the other hand, in order for vibrations to be converted into a current or voltage, which is the primary function of a sensor. This diaphragm consists of magnets that are twisted with metal wire to make a diaphragm that is encased inside a shell. There is a possibility that when sound signals strike the diaphragm of the sensor, magnets within the sensor vibrate, causing current to flow from the coils at the same time.

Using a sound detection sensor module for Arduino, it is possible to determine if a threshold value has been exceeded by the device. There is a microphone that detects sound and feeds the signal into an op-amp called the LM393. To set the volume of the sound, a potentiometer is built into the device and can be used to adjust it. Whenever the sound level of the module reaches a certain threshold, an LED on the module will illuminate and the output will be turned off.

3.5 Software

3.5.1 Blynk Application

Blynk is an IoT platform for iOS and Android smartphones that uses the Internet to control Arduino, Raspberry Pi, and NodeMCU. This application is used to compile and provide the required address on the available widgets to construct a graphical interface or human machine interface (HMI). An IoT platform supports a wide range of device kinds and protocols out of the box. An IoT platform is especially useful for devices that do not support conventional IoT protocols since it provides a software development kit (SDK) to integrate devices with the rest of your ecosystem.

Open the Library Manager in the Arduino IDE and look for the Blynk library. Look for the Blynk library.

```
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";
```

Figure 3.19: Wi-Fi Credential Code

fter entering your data (ssid, password, and token), submit the code to the board. Then, put the detail needed like TEMPLATE_ID, TEMPLATE_NAME and AUTH_TOKEN given.

```
#define BLYNK_TEMPLATE_ID          "MyTemplateID"
#define BLYNK_TEMPLATE_NAME        "MyTemplateName"
#define BLYNK_AUTH_TOKEN           "MyAuthToken"
```

Figure 3.20: Main Parameters to Define

Follow each step in order to connect and receive data. The figure below illustrates the user interface applications on a mobile phone. All configuration is done through the Blynk Console via the website.

Figure 3.21: Interface of Blynk

3.6 Summary

This chapter presents the proposed methodology for developing this project. This method focuses on simple, less rigorous and effective estimation without significantly affecting results' accuracy. With this method, the focus is not on obtaining maximum accuracy, but rather on monitoring and analyzing each parameter on a large-scale distribution network.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter demonstrated the overall model of the weather monitoring system and briefly reviewed its development. The examination of the results of each variable sensor discovered after testing with various distances between the item and sensors was described.

4.2 The Development of Weather Monitoring System Using LoRa

The plan of the weather conditions observing framework had been talked about in this segment. Figure 4.1 and Figure 4.2 show the improvement of the weather conditions checking framework utilizing LoRa. For the weather conditions observing framework, it gave transmitter and recipient. The RFM95 LoRa module serves as the weather monitor's communication devices, and the Arduino Uno serves as the system's microcontroller for the transmitter. for the sensor utilizing on transmitter is DHT-11 for temperature and dampness, anemometer for wind speed, Bmp280 to see the strain barometric, commotion sensor for clamor and raindrop module for the downpour identifier. RFM95 LoRa module at the transmitter part is utilized to communicate information to the LoRa module at the collector, while the Cytron Safeguard Esp-Wi-Fi Fire up 2.0 is utilized to get information from transmitter and ship off the Blynk applications. Figure 4.1 shows the equipment of transmitter for the weather conditions checking framework. For the beneficiary part, the equipment was arrangement by involving the Arduino UNO as the microcontroller, RFM95 LoRa module as the specialized gadget that used to get information from the transmitter and the Cytron-Esp-Wi-Fi 2.0 as the handset module. Figure 4.2 shows the equipment of beneficiary for the weather conditions checking.

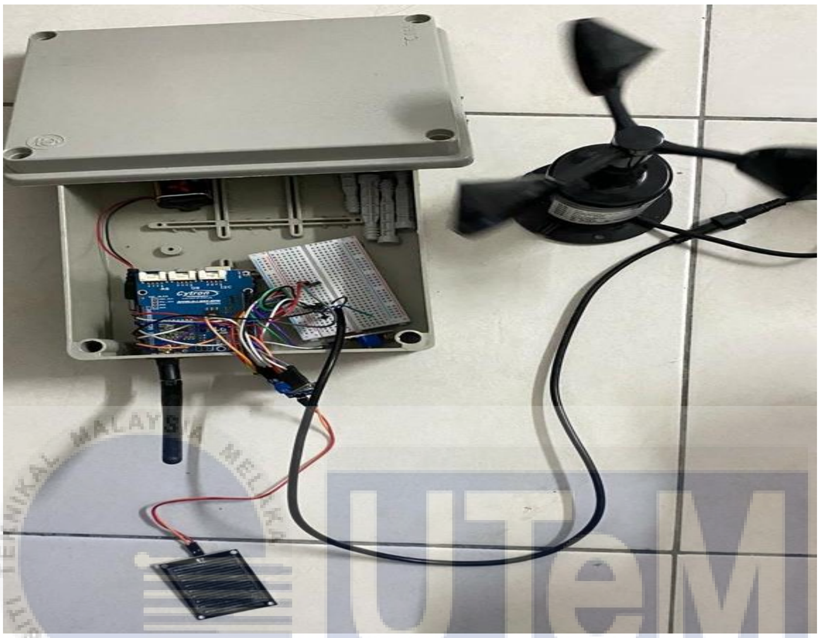


Figure 4.1: Tranmsitter Of Weather Monitoring System

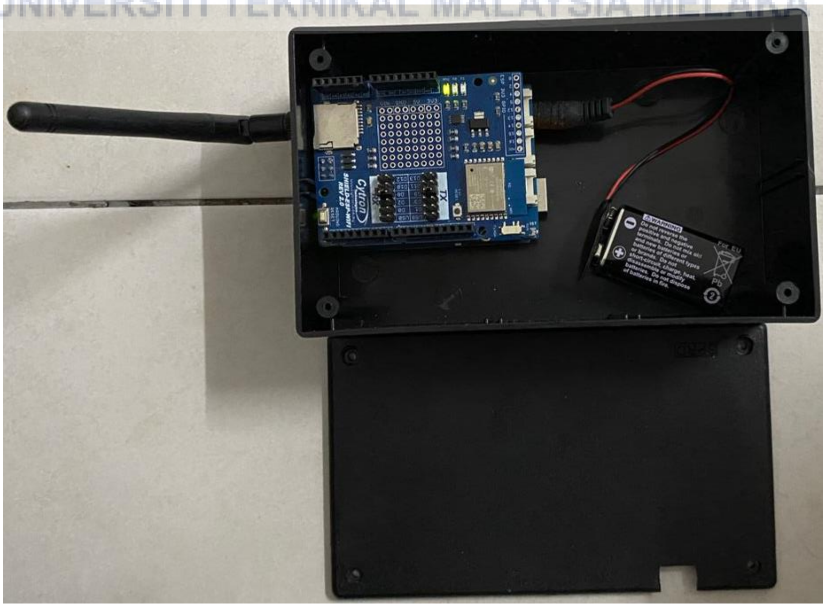


Figure 4.2: Receiver of Weather Monitoring System

for the application it is necessary to set in the blynk console first for the link in the application on the mobile phone. In the blynk console that needs to be set are the main datastream and dashboard. to notify the user needs to be set to automation.

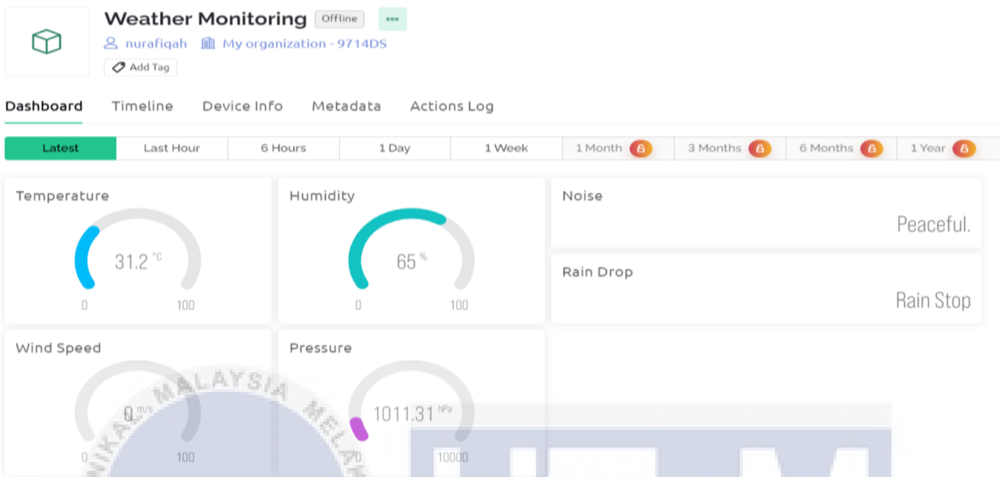


Figure 4.3: Blynk Console To Setting And Connect With Devices

Figure 4.4: Interface of Weather Monitoring System

4.3 Result and Analysis

For Day 1, data was taken from 3 pm to 3 am to see the difference in each variable such as wind speed, temperature, humidity, and pressure in the evening until late at night. On day 2, data was taken from 8 am to 8 pm.

Here you can also see the difference between 3 p.m and 8 p.m for both days. on day 1 the temperature slightly decreased and day 2 slightly increased.

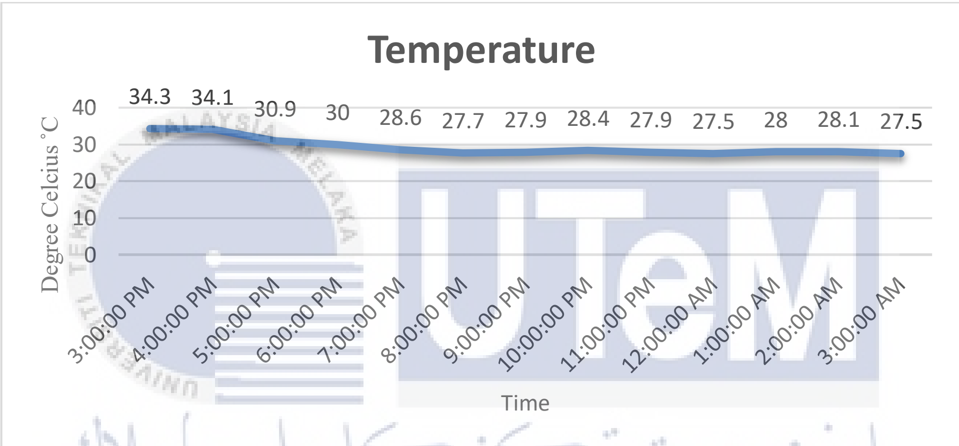


Figure 4.5: Graph of Temperature Day 1

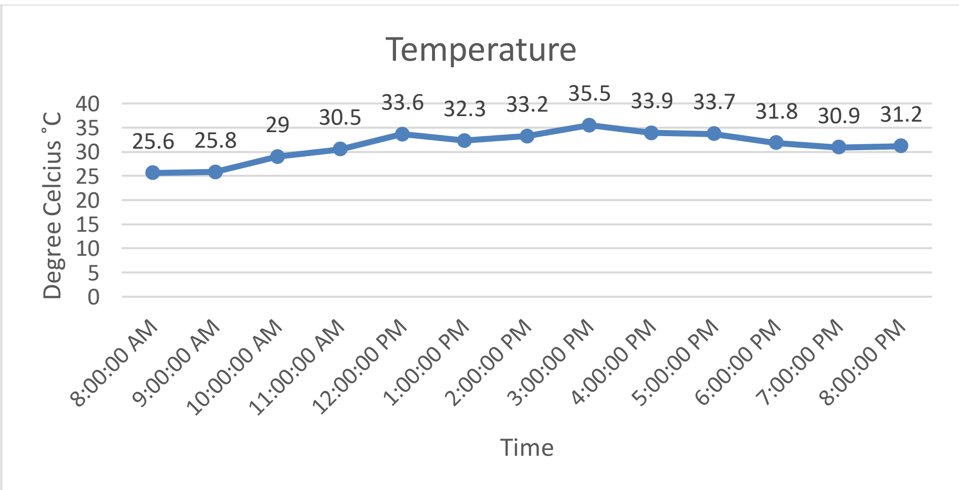


Figure 4.6: Graph of Temperature Day 2

For the humidity on days 1 and 2 increased, showing the weather for good humidity.

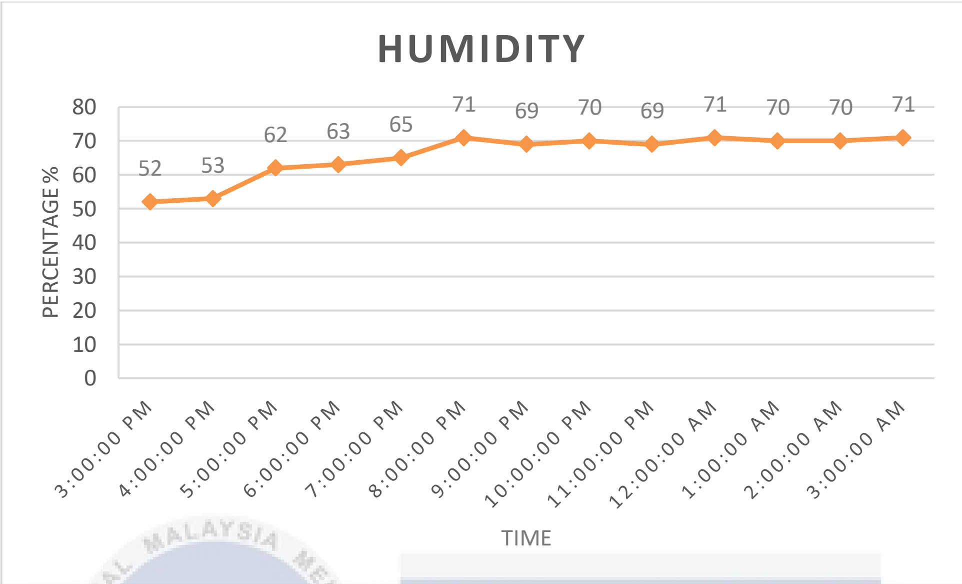


Figure 4.7: Graph of Himidity Day 1

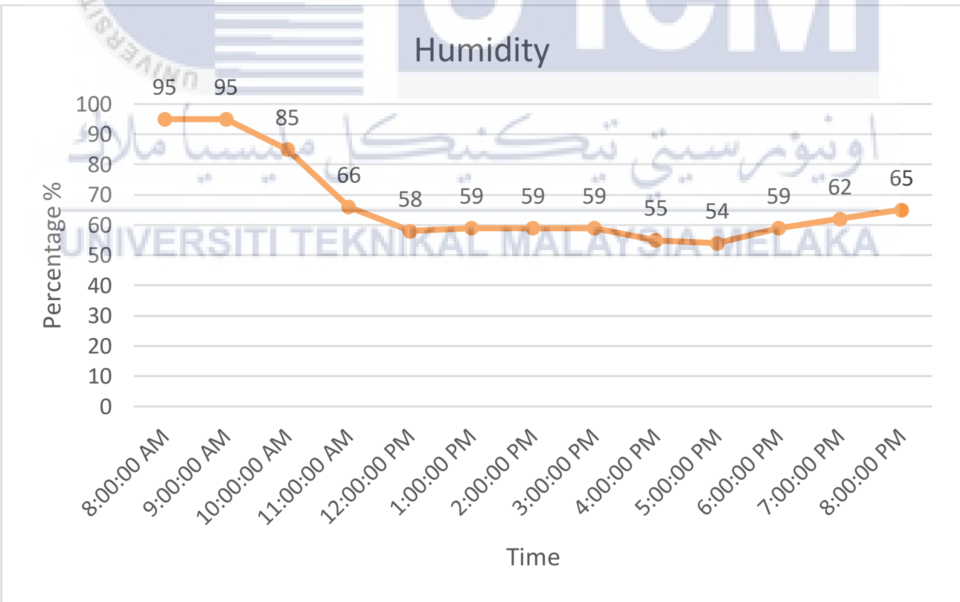


Figure 4.8: Graph of Himidity Day 2

Then tell about the wind speed on the 1st day showing an increase in the wind speed at 4 p.m until 10 p.m and also on day 2 from 8 a.m to 10 a.m and occur at 7 p.m as well.

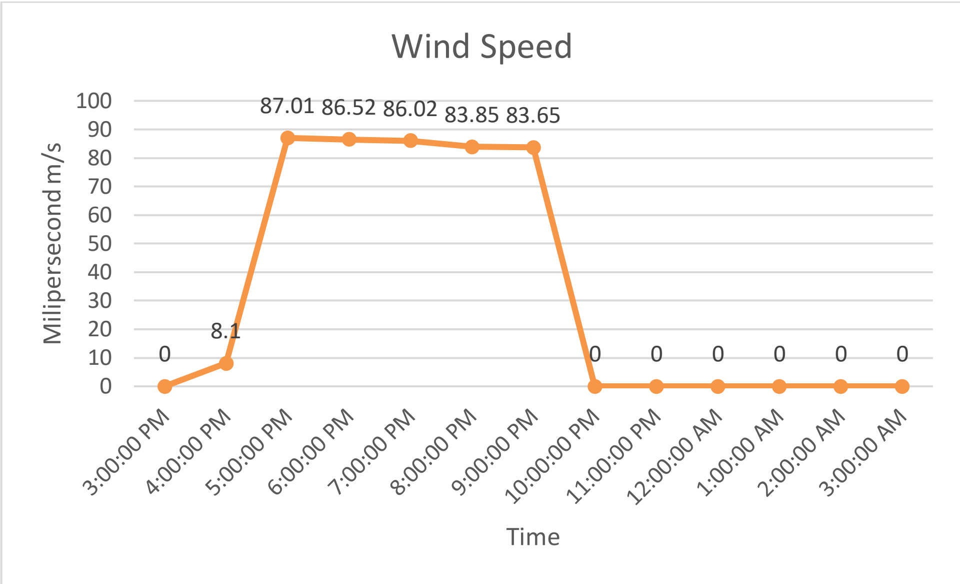


Figure 4.9: Graph of Wind Speed Day 1

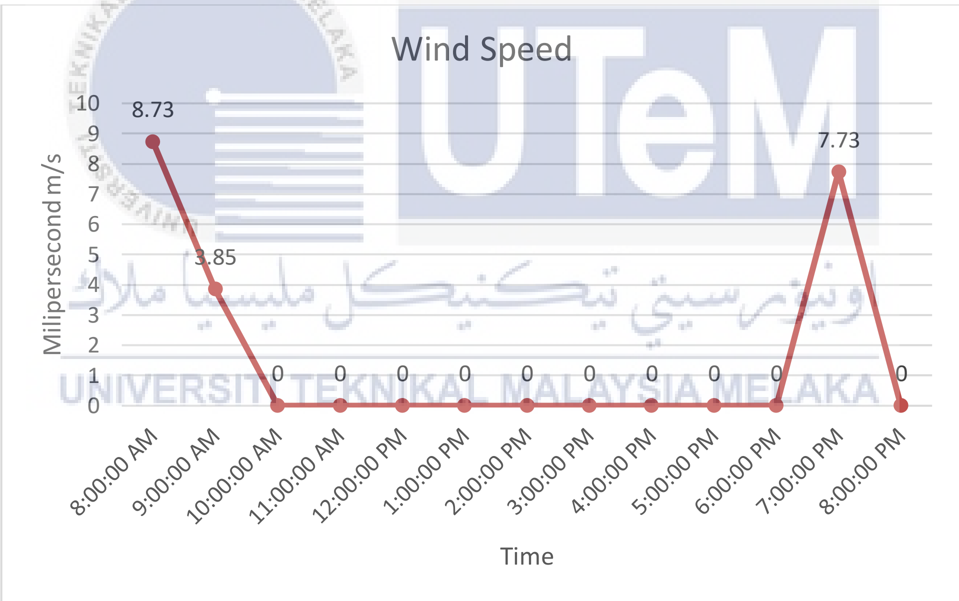


Figure 4.10: Graph of Wind Speed Day 2

The difference in pressure is on day 1 from 7 p.m to 3 a.m there is high pressure while on day 2 there is high pressure from 8 am to 1 p.m but at 12 noon there is a temporary drop in pressure. Then after 1 p.m the pressure starts to drop until 4 p.m.

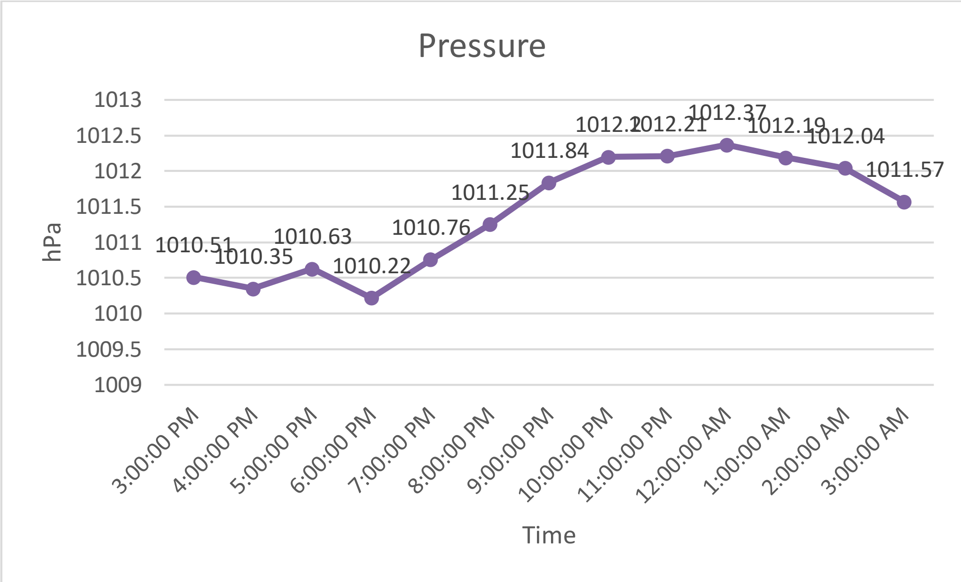


Figure 4.11: Graph of Pressure Day 1

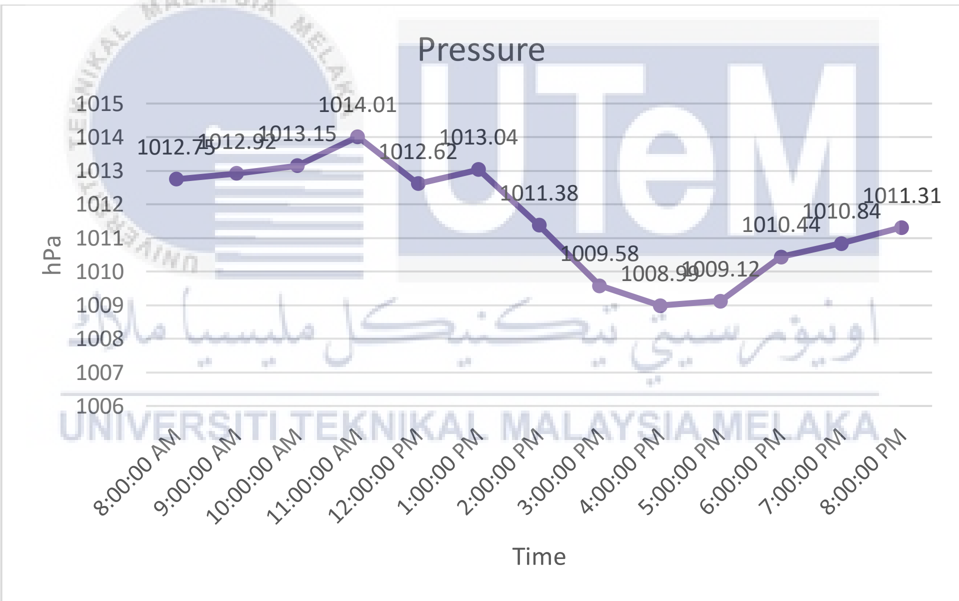


Figure 4.12: Graph of Pressure Day 2

4.4 Display of Results

4.4.1 Arduino Ide on Transmitter

The loop function collects data from a variety of sensors, including DHT, sound, rain, wind, and BMP280. And generates a string named dataString, which concatenates the sensor readings separated by commas. The section of the code starts a new LoRa packet, prints the dataString (sensor readings), and then closes the packet to transmitted wirelessly via the LoRa module.

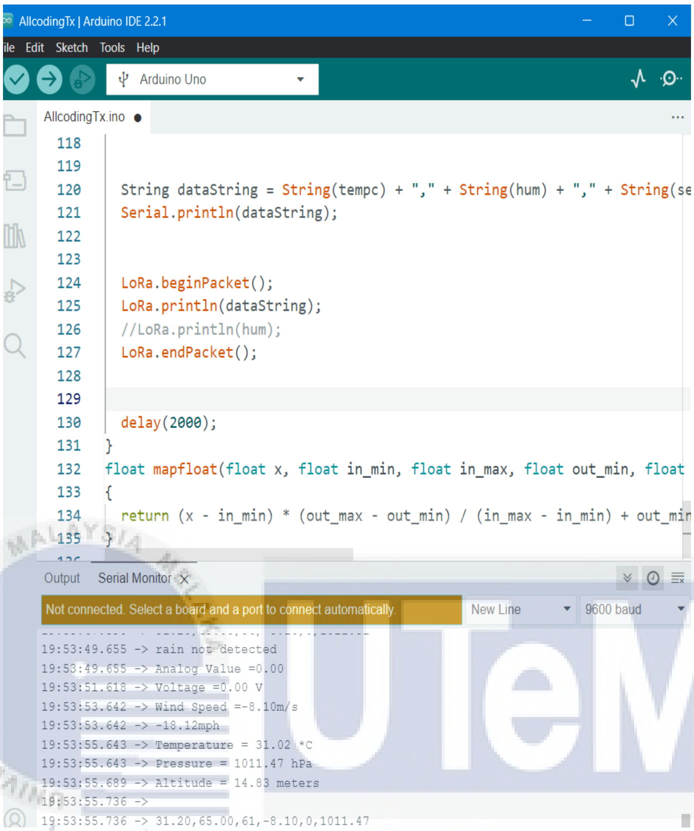


Figure 4.13: The Serial Monitor Tool On Transmitter of Arduino IDE

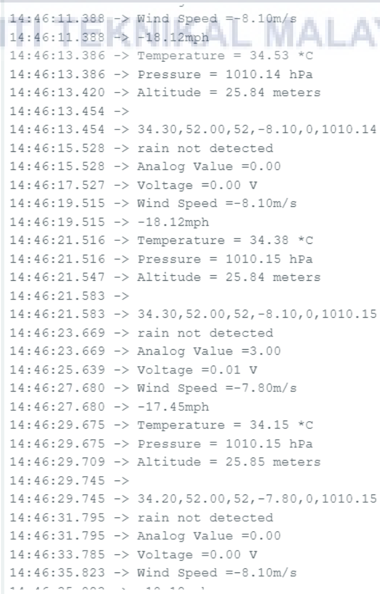


Figure 4.14: Data On Transmmitter In Serial Monitor of Arduino IDE

4.4.2 Arduino Ide on Receiver

This part detects incoming LoRa packets, reads the contents, and then parses the received string with commas as delimiters. The lines essentially find the commas in the received data string, allowing distinct portions of the string to be extracted between them. The following code extracts substrings from the original string using these indexes, allowing for the separation of separate sensor values. Each commaIndex variable serves to establish the boundaries for extracting various pieces of information from the received data string. Each substring is then converted to the proper data type (double, integer, or float), and the results are displayed on the Serial Monitor. Additionally, the data is delivered to the Blynk app, which enables real-time monitoring on a mobile device.

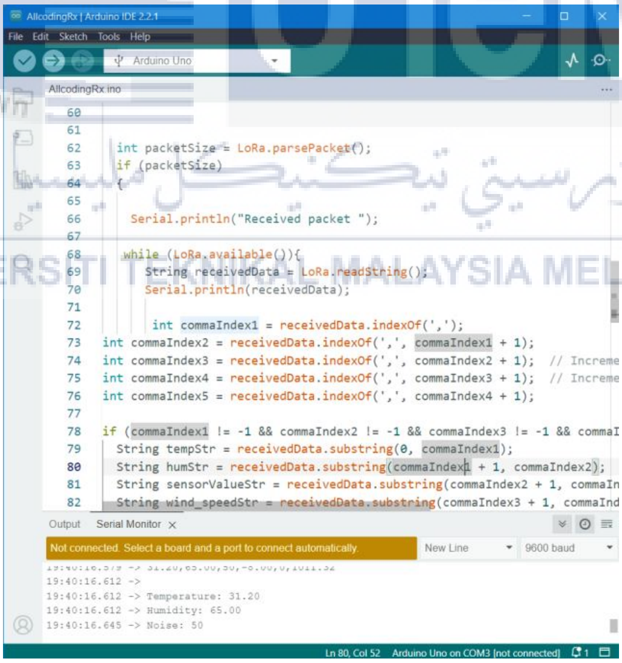


Figure 4.15: The Serial Monitor Tool On Receiver of Arduino IDE

```
20:17:25.809 -> Received packet
20:17:25.809 -> 27.80,70.00,49,83.85,1023,1011.44
20:17:25.854 ->
20:17:25.854 -> Temperature: 27.80
20:17:25.875 -> Humidity: 70.00
20:17:25.917 -> Noise: 49
20:17:25.917 -> Pressure: 1011.44
20:17:26.745 -> Rainy
20:17:32.156 -> Received packet
20:17:32.156 -> 27.70,70.00,49,83.85,1023,1011.43
20:17:32.220 ->
20:17:32.220 -> Temperature: 27.70
20:17:32.233 -> Humidity: 70.00
20:17:32.263 -> Noise: 49
20:17:32.263 -> Pressure: 1011.43
20:17:33.092 -> Rainy
20:17:38.496 -> Received packet
20:17:38.529 -> 27.70,71.00,49,83.85,1023,1011.43
20:17:38.562 ->
20:17:38.562 -> Temperature: 27.70
20:17:38.594 -> Humidity: 71.00
20:17:38.594 -> Noise: 49
20:17:38.630 -> Pressure: 1011.43
20:17:39.434 -> Rainy
20:17:44.854 -> Received packet
20:17:44.887 -> 27.80,70.00,49,83.85,1023,1011.42
20:17:44.920 ->
20:17:44.920 -> Temperature: 27.80
20:17:44.920 -> Humidity: 70.00
20:17:44.952 -> Noise: 49
20:17:44.952 -> Pressure: 1011.42
20:17:45.803 -> Rainy
```

Figure 4.16: Data On Receiver In Serial Monitor of Arduino IDE

```
    ___  __          __
   / _ )/ /_ _____  / /__
  / _  / / // / _ \/  '_/
 /____/_/\_, /_//_/_/\_\
        /___/ v1.3.2 on Arduino Uno

 #StandWithUkraine    https://bit.ly/swua


[2639] Connecting to MuhdNur
[15728] AT version:2.2.0.0(b097cdf - ESP8266 - Jun 17 2021 12:57:45)
SDK version:v3.4-22-g967752e2
compile time(6800286):Aug  4 2021 17:34:06
Bin version:2.2.0(Cytron_ESP-12F_WROM2O
[18334] +CIFSR:STAIP,"192.168.68.138"
+CIFSR:STAMAC,"c8:c9:a3:02:56:d5"
[18345] Connected to WiFi
[28703] Ready (ping: 25ms).
LoRa Receiver
```

Figure 4.17: Blynk Connecting With Wi-Fi

4.5 Blynk Apps for Monitoring

The Figure 4.18 below shows the data that has been read by each sensor. Each sensor has a role in detecting each weather change. For this section, every variable detected by the transmitter, such as temperature, humidity, wind speed, pressure, sound, and raindrop, is relayed to the receiver via Cytron LoRa and then to Blynk via Shiel-Esp-Wifi Rev 2.0. As depicted in the Figure 4.18 below, every data received can be examined using the Blynk app.

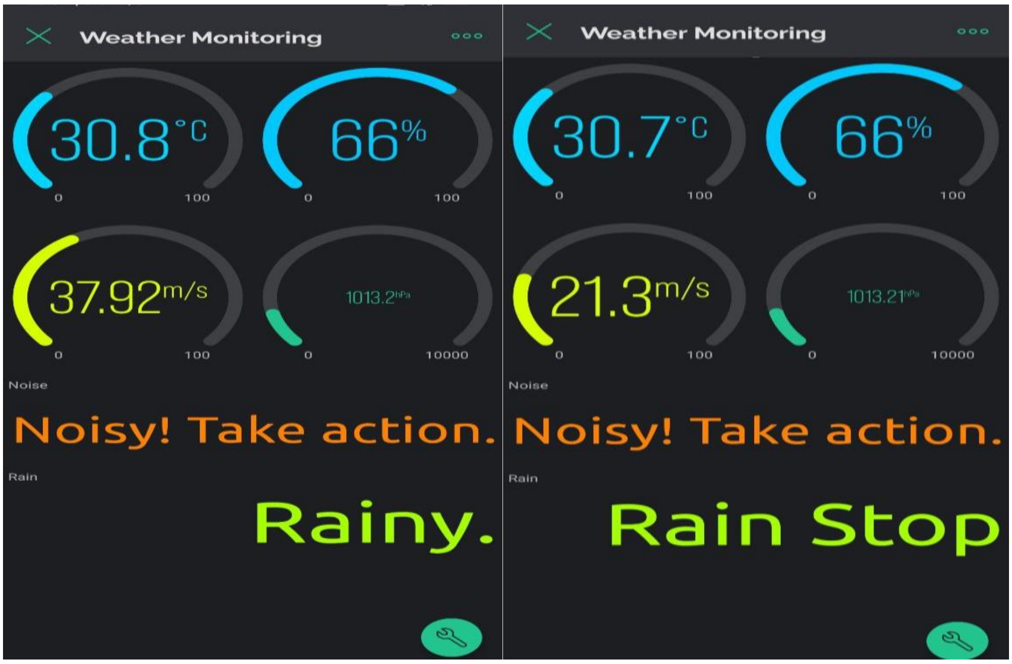


Figure 4.18:Weather Monitoring System With Different Value And Conditions

For the raindrop sensor module will detect rain if a rain situation occurs and will send a notification "Raininggg!!" to the user as shown in the Figure 4.19 below.

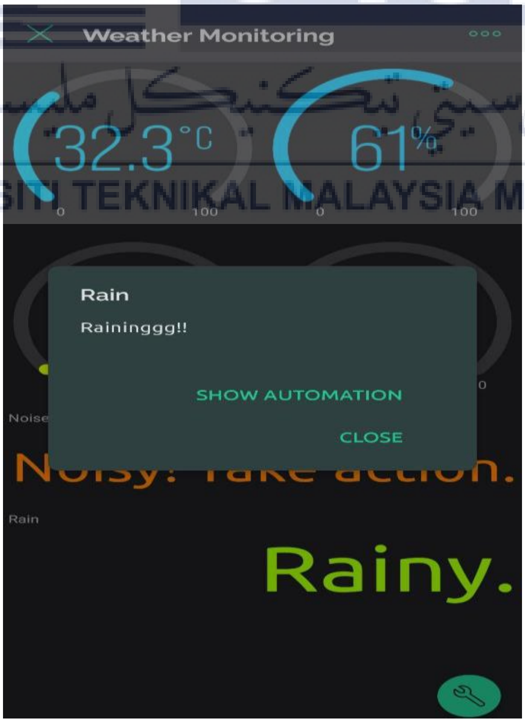


Figure 4.19: Notify To User

4.6 Summary

Results and data analysis have been presented in this chapter. Explanation of the difference in data analysis on each variable such as temperature, humidity, wind speed and pressure that can indicate significant weather changes. For rain, show a statement on the mobile application if it rains or stops. Then for the sound sensor is the addition of a sensor to detect noise. This chapter also shows the use of Arduino IDE for programming into Arduino Uno and seeing the results through a serial monitor.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter will discuss the project's conclusion and make recommendations for future improvements.

5.2 Conclusion

In conclusion, this project includes both hardware and software development. The transmitter and receiver are part of the hardware. The Arduino Uno microcontroller is used for both the transmitter and receiver, and the Cytron lora RFM radio frequency board is used for the radio frequency board. However, for the transmitter, just use an Arduino Uno and a Cytron lora RFM. The Arduino Uno, Cytron Shielf-wifi Rev 2.0, and lora module are used to build the receiver. The language utilised in this project for the software portion is C++ programming. The weather monitoring system was built using lora technology, which has a long range and low power usage. The lora module functions as a communication device, receiving and transmitting data. The Blynk platform is used to remotely monitor the weather. Rain will be reported to users via Blynk Apps. Furthermore, it takes some time for lora receivers to receive data from the transmitter. As the distance between the transmitter and the receiver grows, the signal weakens. As a result, the data takes longer to reach the recipient. This project's objectives have been met with success.

5.3 Future Works

5.3.1 Microcontroller

For the microcontroller can be replaced from Arduino Uno to one that has a Wi-Fi module to facilitate programming and wiring connections between the board and the wifi module that has the same frequency between the transmitter and the receiver.

5.3.2 Solar Power Supply

Solar power generates electricity by converting solar energy. For our use, the sun generates two types of energy: electricity and heat. Both are produced by the use of solar panels, which range in size from residential rooftops to "solar farms" spanning acres of rural terrain. If your weather station will be put in an open area with plenty of sunlight, consider employing solar power. Select a solar panel with enough capacity to charge a rechargeable battery or directly power the system during daytime hours. Include a charge controller to manage battery charging and prevent overcharging.

5.4 Project Potential

This lora weather monitoring system can be used to detect variables other than temperature, humidity, pressure, raindrop, wind speed and sound. It can also detect other variables such as soil moisture sensor that can be monitored for agriculture that is Soil water potential, it also known as soil moisture tension, is a measure of how tightly water clings to the soil and is expressed in pressure units known as bars. Then farmers can monitor their soil moisture from looking in blynk.

REFERENCES

Ariffin Noordin, K., Chee Onn, C., & Faizal Ismail, M. (2006). A Low-Cost Microcontroller-based Weather Monitoring System. In Journal (Vol. 5, Issue 1).

Atiwanwong, T., & Hongprasit, S. (n.d.). A Low-power Real-time Pollution Monitoring System Using ESP LoRa. In MAHASARAKHAM INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY (Vol. 6, Issue 1).

Balakrishnan Sivakumar, & Chikkamadaiah Nanjundaswamy. (2021). Weather monitoring and forecasting system using IoT. Global Journal of Engineering and Technology Advances, 8(2), 008–016. https://doi.org/10.30574/gjeta.2021.8.2.0109

Bestari, D. N., & Wibowo, A. (2023). An IoT-Based Real-Time Weather Monitoring System Using Telegram Bot and Thingsboard Platform. International Journal of Interactive Mobile Technologies, 17(6), 4–19. https://doi.org/10.3991/ijim.v17i06.34129

Bharadwaj, A. (2021). Raspberry Pi Based Weather Monitoring System.

Bin Zohari, M. H., & Bin Johari, M. F. (2019). Weather monitoring system using blynk application. International Journal of Innovative Technology and Exploring Engineering, 9(1), 1315–1319. https://doi.org/10.35940/ijitee.L3666.119119

Fadhli, M., Asriyadi, A., Lindawati, L., Salamah, I., Affrylia, G., Valerie, M., & Ramadhan, A. (2022). Low Cost Air Quality Monitoring System Using LoRa Communication Technology.

Girija C. (n.d.). Internet of Things (IOT) based Weather Monitoring System. www.ijert.org

Holovatyy, A. (2021). Development of IoT Weather Monitoring System Based on Arduino and ESP8266 Wi-Fi Module. IOP Conference Series: Materials Science and Engineering, 1016(1), 012014. https://doi.org/10.1088/1757-899x/1016/1/012014

Ibrahim Musa, A. (2018). Micro-controller Based Mobile Weather Monitor System. American Journal of Embedded Systems and Applications, 6(1), 23. https://doi.org/10.11648/j.ajesa.20180601.14

Jayasree, G., Anjali, G., Lakshmi Saraswathi, D., Shamshujeya, S., Kranthi Kumar, M., & Tech, M. (2023). LoRa BASED SMART CITY AIR QUALITY MONITORING SYSTEM Under the Guidance of Mr. In International Journal of Creative Research Thoughts (Vol. 11). www.ijcrt.org

Kamarudin, M. A., Yunus, N. H. M., Razak, M. R. A., Nadzir, M. S. M., & Alhasa, K. M. (2022). Development of Blynk IoT Platform Weather Information Monitoring System. In Advanced Structured Materials (Vol. 162, pp. 295–305). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-92964-0_29

Krishnamurthi, K., Thapa, S., Kothari, L., & Prakash, A. (2015). Arduino Based Weather Monitoring System. International Journal of Engineering Research and General Science, 3(2). www.ijergs.org

Mahajan, N., & Supriya Konar, I. (2023). Issue 6 www.jetir.org (ISSN-2349-5162). In JETIR2306617 Journal of Emerging Technologies and Innovative Research (Vol. 10). www.jetir.org

Marwa, C., Othman, S. Ben, & Sakli, H. (2020). IoT Based Low-cost Weather Station and Monitoring System for Smart Agriculture. Proceedings - STA 2020: 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, 349–354. https://doi.org/10.1109/STA50679.2020.9329292

Muhammad Farid, A. T. (2021). Pembangunan pesat punca banjir kilat di bandar. Kosmo.

Murdyantoro, E., Setiawan, R., Rosyadi, I., Nugraha, A. W. W., Susilawati, H., & Ramadhani, Y. (2019). Prototype weather station uses LoRa wireless connectivity infrastructure. Journal of Physics: Conference Series, 1367(1). https://doi.org/10.1088/1742-6596/1367/1/012089

Nerella, O. (n.d.). Internet of Things (IOT) Based Weather Monitoring system. https://doi.org/10.17148/IJARCCE.2016.5966

Parashar, T., Gahlot, S., Godbole, A., & Thakare, Y. B. (2015). Licensed Under Creative Commons Attribution CC BY Weather Monitoring System Using Wi-Fi. In International Journal of Science and Research (IJSR) ISSN. www.ijsr.net

Promput, S., Maithomklang, S., & Panya-isara, C. (2023). Design and Analysis Performance of IoT-Based Water Quality Monitoring System using LoRa Technology. TEM Journal, 12(1), 29–35. https://doi.org/10.18421/TEM121-04

Purnomo, F. A., Yoeseph, N. M., Bawono, S. A. T., & Hartono, R. (2021). Development of air temperature and soil moisture monitoring systems with LoRA technology. Journal of Physics: Conference Series, 1825(1). https://doi.org/10.1088/1742-6596/1825/1/012029

Purnomo, F. A., Yoeseph, N. M., Yulianto, A., Royan, Y. I., & Safi'ie, M. A. (2022). Development of wind monitoring systems with LoRA technology. IOP Conference Series: Earth and Environmental Science, 989(1). https://doi.org/10.1088/1755-1315/989/1/012011

Rusyaidi, M., Ridzwan, M., Uddin, S. M., Izhar, M., & Bakar, A. (2022). Development of IoT Weather Monitoring System. Journal of Engineering Technology, 10(1), 70–75.

Sai Manasa, R., Aarthi Reddy, P., Nandini Reddy, T., Ramakrishna Goud, G., & Bhulakshmi, K. (2023). LoRa based Wireless Weather Station with WebServer. INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH.

Sharma, P., & Prakash, S. (2021). Real Time Weather Monitoring System Using Iot. ITM Web of Conferences, 40, 01006. https://doi.org/10.1051/itmconf/20214001006

SOY, H., & DILAY, Y. (2021). A Conceptual Design of LoRa based Weather Monitoring System for Smart Farming. European Journal of Science and Technology. https://doi.org/10.31590/ejosat.1011947

APPENDICES

Appendix 1: Gant Chart of The Project

PROJECT ACTIVITY	WEEK													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Debug the board trasnmitter and receiver														
Configure The Wi-Fi Shield														
connect with Blynk														
Temperature & Humidity Sensor														
Log Book Writing														
Wind Speed Sensor														
Barometric Pressure Sensor														
Raindrop Sensor														
Noise Sensor														
Complete the hardware														
Chapter 4&5 Report Writing														
Slide Presentation														
Submit Report														
BDP presentation														

Appendix 2: Coding of Transmitter

```
#include <SPI.h>
#include <LoRa.h>
#include <DHT.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Wire.h>
#include <Adafruit_BMP280.h>


int sensorValue = 0;


#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

int sensorPin = A2;     // input for LDR and rain sensor
int enable2 = 13;       // enable reading Rain sensor
int sensorValue2 = 0;


const int soundSensorPin = A0;


#define BMP_SCK  (13)
#define BMP_MISO (12)
#define BMP_MOSI (11)
#define BMP_CS   (10)

Adafruit_BMP280 bmp;


void setup()
{
  Serial.begin(9600);
  Serial.println("LoRa Sender");
  if (!LoRa.begin(915E6))
  {
    Serial.println("Starting LoRa failed!");
```

```
    while(1);
  }

  dht.begin();
  unsigned status;
//  status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
  status = bmp.begin(0x76);
  bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,     /* Operating
Mode. */
                  Adafruit_BMP280::SAMPLING_X2,     /* Temp.
oversampling */
                  Adafruit_BMP280::SAMPLING_X16,    /* Pressure
oversampling */
                  Adafruit_BMP280::FILTER_X16,      /* Filtering.
*/
                  Adafruit_BMP280::STANDBY_MS_500); /* Standby
time. */
}
void loop()
{

  float tempc = dht.readTemperature();
  float hum = dht.readHumidity();
 int sensorValue = analogRead(soundSensorPin);
  sensorValue2 = analogRead(sensorPin);
sensorValue2 = constrain(sensorValue2, 150, 440);
sensorValue2 = map(sensorValue2, 150, 440, 1023, 0);
float pressure = bmp.readPressure() / 100.0F;

  if (sensorValue2>= 20)
{
  Serial.println("rain is detected");
  Serial.print(sensorValue2);

  }
  else

{
  Serial.println("rain not detected");

  }
```

```
  if (isnan(tempc), isnan(hum)) {
    Serial.println("Failed to read temperature & humidity from
DHT sensor");
    }


     float sensorVal = analogRead(A1);
  Serial.print("Analog Value =");
  Serial.println(sensorVal);
 delay(2000);
  float voltage = (sensorVal / 1023) * 5;
  Serial.print("Voltage =");
  Serial.print(voltage);
  Serial.println(" V");
 delay(2000);
  float wind_speed = mapfloat(voltage, 0.4, 2, 0, 32.4);
  float speed_mph = ((wind_speed *3600)/1609.344);
  Serial.print("Wind Speed =");
  Serial.print(wind_speed);
  Serial.println("m/s");
  Serial.print(speed_mph);
  Serial.println("mph");
  delay(2000);


  Serial.print("Temperature = ");
  Serial.print(bmp.readTemperature());
  Serial.println(" *C");

  Serial.print("Pressure = ");
  Serial.print(pressure);
  Serial.println(" hPa");

  Serial.print("Altitude = ");
  Serial.print(bmp.readAltitude());
  Serial.println(" meters");

  Serial.println();
```

```
  String dataString = String(tempc) + "," + String(hum) + "," +
String(sensorValue) + "," + String(wind_speed) + "," +
String(sensorValue2) + "," + String(pressure);
  Serial.println(dataString);


  LoRa.beginPacket();
  LoRa.println(dataString);
  //LoRa.println(hum);
  LoRa.endPacket();


  delay(2000);
}
float mapfloat(float x, float in_min, float in_max, float
out_min, float out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min)
+ out_min;
}
```

Appendix 3: Coding of Receiver

```
#define BLYNK_PRINT Serial

/* Fill in information from Blynk Device Info here */
#define BLYNK_TEMPLATE_ID "TMPL6FykyyW_Q"
#define BLYNK_TEMPLATE_NAME "Weather Monitoring"
#define BLYNK_AUTH_TOKEN "4SSGSIDceg1y_oOyWpswQ2nb47YOdKyE"


#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
#include <SPI.h>
#include <LoRa.h>
#include <ArduinoJson.h>

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "MuhdNur";
char pass[] = "mnr0137089943";
char auth[] = "4SSGSIDceg1y_oOyWpswQ2nb47YOdKyE";
// Hardware Serial on Mega, Leonardo, Micro...
#define EspSerial Serial1

// or Software Serial on Uno, Nano...
#include <SoftwareSerial.h>
SoftwareSerial EspSerial(8, 9); // RX, TX

// Your ESP8266 baud rate:
#define ESP8266_BAUD 9600

ESP8266 wifi(&EspSerial);



void setup()
{
  // Debug console
  Serial.begin(9600);

  delay(1000);
```

```
  // Set ESP8266 baud rate
  EspSerial.begin(ESP8266_BAUD);

  delay(1000);

  Blynk.begin(auth, wifi, ssid, pass);

  Serial.println("LoRa Receiver");
  if (!LoRa.begin(915E6))
  {
    Serial.println("Starting LoRa failed!");
    while (1);
  }

}

void loop()
{
  Blynk.run();


  int packetSize = LoRa.parsePacket();
  if (packetSize)
  {

    Serial.println("Received packet ");

   while (LoRa.available()){
      String receivedData = LoRa.readString();
      Serial.println(receivedData);

       int commaIndex1 = receivedData.indexOf(',');
int commaIndex2 = receivedData.indexOf(',', commaIndex1 + 1);
int commaIndex3 = receivedData.indexOf(',', commaIndex2 +
1);  // Increment based on commaIndex2
int commaIndex4 = receivedData.indexOf(',', commaIndex3 +
1);  // Increment based on commaIndex3
int commaIndex5 = receivedData.indexOf(',', commaIndex4 + 1);

if (commaIndex1 != -1 && commaIndex2 != -1 && commaIndex3 != -1
&& commaIndex4 != -1 && commaIndex5 != -1) {
```

```
  String tempStr = receivedData.substring(0, commaIndex1);
  String humStr = receivedData.substring(commaIndex1 + 1,
commaIndex2);
  String sensorValueStr = receivedData.substring(commaIndex2 +
1, commaIndex3);
  String wind_speedStr = receivedData.substring(commaIndex3 + 1,
commaIndex4);
  String sensorValue2Str = receivedData.substring(commaIndex4 +
1, commaIndex5 );
  String mcgStr = receivedData.substring(commaIndex5 + 1);



      // Convert strings to doubles
      double tempc = tempStr.toDouble();
      double hum = humStr.toDouble();
      int sensorValue = sensorValueStr.toInt();
      double windSpeed = wind_speedStr.toDouble();
      float sensorValue2 = sensorValue2Str.toFloat();
      double mcg = mcgStr.toDouble();


      // Print the received values
      Serial.print("Temperature: ");
      Serial.println(tempc);
      Serial.print("Humidity: ");
      Serial.println(hum);
      Serial.print("Noise: ");
      Serial.println(sensorValue);
      Serial.print("Pressure: ");
      Serial.println(mcg);



      Blynk.virtualWrite(V0,tempc);
      Blynk.virtualWrite(V1,hum);
      Blynk.virtualWrite(V3,windSpeed);
      Blynk.virtualWrite(V5,mcg);


      if (sensorValue > 50){
```

```
        Serial.println("Noisy! Take action.");

        Blynk.virtualWrite(V2,"Noisy! Take action.");

      }else
      {
        Blynk.virtualWrite(V2,"Peaceful.");
        }

      if (sensorValue2 > 50){

        Serial.println("Rainy");

        Blynk.virtualWrite(V4,"Rainy.");

      }else
      {
        Serial.println("Rain Stop");
        Blynk.virtualWrite(V4,"Rain Stop");
        }



    } else {
      Serial.println("Invalid message format");
    }



    }


  }

}
```

Appendix 4: Turnitin Report

psm 2

ORIGINALITY REPORT

23%	17%	14%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	1%
2	www.atlantis-press.com Internet Source	1%
3	article.sciencepublishinggroup.com Internet Source	1%
4	docplayer.net Internet Source	1%
5	F.A. Purnomo, N.M. Yoeseph, A. Yulianto, Y.I. Royan, M.A. Safi'ie, "Development of wind	1%