DEVELOPMENT OF WEATHER MONITORING SYSTEM USING ESP32 WITH MOBILE APPLICATION INTERFACE

KUSHALHINI A/P DEVENDRA KUMAR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF WEATHER MONITORING SYSTEM USING ESP32 WITH MOBILE APPLICATION INTERFACE

KUSHALHINI A/P DEVENDRA KUMAR

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

Faculty of Electronics and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF WEATHER MONITORING SYSTEM USING ESP32 WITH MOBILE APPLICATION INTERFACE

Sesi Pengajian : 2023/2024

Saya KUSHALHINI A/P DEVENDRA KUMAR 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) (COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: ....

TS. ELIYANA BINTI RUSLAN
Pensyarah Kanan
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKE
Universiti Teknikal Malaysia Melaka (UTeM)

Tarikh : 13 JANUARI 2025 Tarikh : 13 JANUARI 2025

*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 WEATHER MONITORING SYSTEM USING ESP32 WITH MOBILE APPLICATION INTERFACE**" 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 : KUSHALHINI DEVENDRA KUMAR

Date : 13 JANUARI 2025

APPROVAL

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

Signature : ....

Supervisor Name : Eliyana binti Ruslan ....

Date : 20 Januari 2025 ....

ABSTRACT

Due to their high cost and tedious data collecting, traditional weather monitoring systems are frequently ineffective, prone to human error, and inaccessible. For sectors that rely on real-time environmental data to make educated decisions, like agriculture, urban planning, and disaster management, these restrictions provide difficulties. The necessity for a cost-effective, automated, and precise weather monitoring system that can deliver trustworthy data in real time is further highlighted by the speed at which climate change is occurring. This project suggests creating an ESP32-based Internet of Things weather monitoring system that is integrated with sensors like the FC-37 (rainfall intensity), BMP280 (atmospheric pressure), and DHT22 (temperature and humidity) in order to address these issues. The system records environmental data, sends it in real time to the Blynk IoT platform for remote access, and shows it locally on an LCD. By offering a scalable, affordable, and user-friendly substitute, this creative solution seeks to enable both individuals and businesses to efficiently monitor environmental conditions. The system's accuracy and dependability were validated by extensive testing, which makes it a powerful and long-lasting instrument for encouraging data-driven decision-making and climate resilience.

ABSTRAK

Disebabkan kos tinggi dan pengumpulan data yang membosankan, sistem pemantauan cuaca tradisional selalunya tidak berkesan, terdedah kepada kesilapan manusia dan tidak boleh diakses. Bagi sektor yang bergantung pada data alam sekitar masa nyata untuk membuat keputusan terdidik, seperti pertanian, perancangan bandar dan pengurusan bencana, sekatan ini memberikan kesukaran. Keperluan untuk sistem pemantauan cuaca yang kos efektif, automatik dan tepat yang boleh menyampaikan data yang boleh dipercayai dalam masa nyata diserlahkan lagi oleh kelajuan perubahan iklim berlaku. Projek ini mencadangkan mewujudkan sistem pemantauan cuaca Internet of Things berasaskan ESP32 yang disepadukan dengan penderia seperti FC-37 (intensiti hujan), BMP280 (tekanan atmosfera) dan DHT22 (suhu dan kelembapan) untuk menangani isu ini. Sistem ini merekodkan data alam sekitar, menghantarnya dalam masa nyata ke platform Blynk IoT untuk akses jauh dan menunjukkannya secara tempatan pada LCD. Dengan menawarkan pengganti yang berskala, mampu milik dan mesra pengguna, penyelesaian kreatif ini bertujuan untuk membolehkan kedua-dua individu dan perniagaan memantau keadaan persekitaran dengan cekap. Ketepatan dan kebolehpercayaan sistem telah disahkan oleh ujian yang meluas, yang menjadikannya instrumen yang berkuasa dan tahan lama untuk menggalakkan pembuatan keputusan dan daya tahan iklim yang dipacu data.

ACKNOWLEDGEMENTS

I would like to begin by expressing my heartfelt gratitude to my supervisor, TS. ELIYANA BINTI RUSLAN for their invaluable guidance, wisdom, and unwavering support throughout this project. Their patience and expertise have been instrumental in shaping the success of my studies.

I am also deeply thankful to my parents, parents-in-law, and beloved family members for their constant love, encouragement, and prayers during my entire period of study. Their unwavering support has been a constant source of motivation for me, and I am truly grateful for their presence in my life.

Furthermore, I would like to extend a special thank you to my dear friends who have stood by me with unwavering encouragement and cooperation throughout this journey. Their support has been indispensable, and I am truly grateful for the strong bond we share.

I am also indebted to all those who have provided me with valuable suggestions and comments during my research. Your input has greatly contributed to the development of this study, and I sincerely appreciate your insights.

Finally, I would like to express my heartfelt appreciation to all individuals who have played a significant role in the completion of this FYP report. Although I regret not being able to personally mention each one of you, please know that your contributions have not gone unnoticed. Thank you all for your support, and I am sincerely grateful for your presence on my academic journey.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

A tool that measures environmental factors without human intervention is the weather monitoring system. The weather monitoring system has recently demonstrated its worth and the community's need to stay informed on weather conditions till industrial. Weather monitoring will assist in gathering data on weather factors including temperature, humidity, altitude, and pressure through various climatic behavior. This weather monitoring system is portable and uses wireless communication to capture data and transmit it to a phone. By wireless communication, users can connect more easily, negating the need for physical presence information that is needed everywhere.

Wireless communication is the transmission of data across a distance whether great or small without the use of a cable. Convenient wireless connections for phones and portable weather monitoring systems are provided by Wi-Fi technology. Hiking and jungle trekking are two examples of outdoor activities when the device might be used. The sensors help the user by enabling them to keep an eye on how the weather is changing. Additionally, it uses Wi-Fi to record data from the LCD and upload it to the internet.

1.2 Project Relation to Sustainable Development Goal

The system that is being used by the is biggest weather monitoring system using Arduino along with a functional mobile application interface for monitoring weather and climate is well aligned specifically to SDG 13 because it makes way for climate resilience and therefore the improvement of adaptative capacity. This project contributes to the social aspect because it makes people more informed about the weather and thus helps to prevent different climate disastrous events and others. The system plays a major role in providing early and accurate alerts on destructive weather conditions that may cause harm to several communities, hence reduce risk and eliminate vulnerability on such types of conditions in various communities. Moreover, by providing climatic information and making it readily accessible to all for use, the system improves climate awareness as well as improves positive climate decisions, measures and practices.

The project is complete consonance with the pledge to ensure the achievement of SDG 11 and the improvement of cities and human settlements for mascot sustainable populations. Monitoring the current weather conditions proved to be one of the significant components of city planning and development because it helped maintain control over issues that stem from changing weather conditions or consecutive reverses. Some of them are as follows but not limited to the following it entails precision in the system when informing the weather forecast or when there is a need for informing the societies that natural disasters are eminent hence boosting the safety and susceptibility of the urban societies to the disaster. Therefore, the project serves as a formative intervention for raising awareness of the populations and the cities on the quality of the air they are breathing and is an architectural improvement of cities that lowers the impact of urbanization on the environment.

Therefore, the project contributes to the advancement of technology and improvement of qualitative and innovative infrastructure together with them, thus responding to its call of achievement of SDG 9. This can be observed in the way technological improvement and development was incorporated in the monitoring of weather such as adopting an Arduino-based application and using mobile applications. All in all, it is a good method which can be readily used in improving the technological advancement of communities particularly in the developing world and is cost effective. Further, it creates employment and improvement of facilities through encouraging renewable and consistent reinvestment in the project; thereby, realizing the finding that innovation drives sustainable development.

Thus, the weather monitoring system provides information that may help avoid or lessen the impact of negative weather conditions and pollutants on our health and well-being, hence satisfying SGD 3. The risks posed by high temperatures, storms, and pollutants to the health of everyone including humans, crops, and livestock can be greatly reduced due to accurate early warnings of the bad weather situation through the meteorological information accessible to various communities. Therefore, the same system enhances the viability of swift actions and, at the same time, offers the data required for the polices that strive to enhance the quality of the environment besides contributing to advancement of the public health initiatives and improvement of wellbeing of the people in general.

1.3 Problem Statement

Environmental parameter monitoring is crucial for several applications. Weather monitoring systems in the past typically required human intervention, which might lead to durability issues and parallax mistakes. This necessitates the creation of an embedded weather monitoring system based on microcontrollers that can monitor and supply data for analysis. For instance, in the field of agriculture, there is a belief that extreme weather events can be caused by climate change. Heat waves, droughts, powerful winds, and torrential rains are a few examples of these events. Plants are severely harmed by droughts because they will eventually run out of water to survive. Moreover, it impacts global poverty as well.

Heat waves can cause crops to experience intense heat, which may reduce plant productivity. The plant may sustain significant damage from exceptionally strong winds, often known as "sandblasting." Since most plants need air to survive, they cannot thrive in prolonged wet conditions; heavy rains that result in flooding can also harm crops and soil structure. Furthermore, a powerful wind and rain combination can topple enormous trees and destroy homes.

In addition, the community may feel uncomfortable given the current circumstances, which include extremely high temperatures and humidity levels like those in Malaysia. In addition, weather patterns might shift suddenly, endangering people's comfort and safety.A person who constantly faces obstacles needs to be aware of any changes in the surrounding environment immediately so they can be ready for the worst.

To address this issue, there is a need for cost-effective, scalable, and easy-to-use climate monitoring solutions that can provide real-time information to multiple audiences regarding climate monitoring systems using Arduino with mobile application interface The aim is to fill this gap. Utilizing affordable and accessible technology, the project seeks to deliver accurate weather information directly to users' smartphones, empowering them to make informed decisions and emergency measures in response to the weather have increased. This solution not only democratizes access to critical climate information but also promotes resilience and sustainable development in underserved communities.

1.4 Project Objective

The objectives of this project are:

a) To design and implement an ESP32 based weather monitoring system that measures air pressure, temperature, and humidity.

b) To develop an interface for mobile applications that can communicate with the ESP32 based system to receive and show real-time weather data.

c) To validate the ability to track weather conditions over time by providing real time updates and historical data storage.

1.5 Scope of Project

The scope of this project are as follows:

a) Design and assembly of components such as ESP32, DHT22 (temperature and humidity sensor), FC-37 (rain sensor), and BMP-180 (barometric pressure sensor).

b) Development of code for NodeMCU (ESP32) for sensor integration and data collection.

c) Integration of sensors to gather and transmit environmental data to a mobile application for real-time monitoring.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

To provide a better understanding of this project, several related studies have been done. The journals, conferences, and articles regarding weather monitoring are cited directly from reliable internet sources and are taken as references for this project.

It is impossible to conduct weather research or weather forecasting without an awareness of the conditions that are frequently present in the atmosphere. This is why people have always been interested in and worked towards creating means to measure different characteristics of the weather. Technology has led to the development of numerous situations and tools for measuring, obtaining, and tracking meteorological data. Meteorology is the academic discipline dedicated to studying weather.

2.2 Weather Monitoring System

Weather science is a branch of science dealing with the atmosphere. Its jurisdiction is the atmosphere of the whole universe, and its requirement involves collaboration with the whole cosmic system of action. Weather data is crucial for both meteorological observation and forecasting, which have practical application in our daily activities. Tracking different climatic variables such as temperature, humidity, air pressure, light intensity, altitude, dew point, and precipitation would be easier if the weather was monitored on an ongoing basis[1].

The term weather is concerned with the current atmospheric condition of a specific place as it is associated with the science of meteorology. The atmosphere is defined as the state of the air at a place at a specific time. We describe it as weather because it is the state of the atmosphere for a short period of time. By observing the normal weather over a long time-period, we can determine the climate of a specific region. Climate is the word that we use to describe the longer-lasting weather patterns.

To accurately describe the atmospheric conditions at a specific location and time, it is necessary to understand, have access to, and measure specific meteorological components or variables. Temperature, relative humidity, air pressure, wind (direction and speed), precipitation, and the intensity of solar radiation or light are some of the most important aspects of weather that need to be examined carefully. Temperature is the most common and essential of those elements because it is responsible for controlling and influencing many of the other weather elements [1].

Monitoring, forecasting, and measuring weather is beneficial in different scenarios and aids in observing various climatic patterns. In the past, people primarily used human observations of wind patterns and cloud cover to track and predict the weather. Farmers were among those who were most adversely affected by the weather. Over time, a range of electronic devices have been created with the ability to measure, track, forecast, and predict different aspects of the weather, thanks to significant scientific and technological advancements. Progress in theory, technology, and observation throughout the centuries have contributed to our comprehension of the atmosphere. At the same time, people in various areas started to observe, document, and take note of the atmosphere.[1]

In developed countries, several satellites are currently orbiting the earth's atmosphere to track the present weather. Additionally, a single purpose may be served by the usage of multiple types of radars. It's crucial to remember that these devices are highly durable, pricey, and frequently associated with advanced technology. Therefore, a system that can be installed anywhere in homes, small enterprises, and institutions must not only properly monitor weather conditions but also be durable, affordable, and able to function in real-time.

2.3 Reflection for the Weather Monitoring System

2.3.1 Arduino powered smart weather monitoring system

The project titled "Arduino powered smart weather monitoring system" was developed at the Sri Krishna College of Engineering and Technology [1]. This study develops a database device using symbols, a two-dimensional control system, and information gathering methods to generate the supplied data. The sensors are the main components of the real-time weather monitoring device. Different types of sensors can be utilized for the purpose of monitoring and regulating temperature and humidity. As the information is shown on the LCD screen,the collected data can also be presented virtually. The system regularly modifies the weather in the specified area to manage and supervise the space within the prescribed location. This enables accurate tracking and display of temperature and humidity in every region, with the outcome visible on the LCD screen.

The authors used the DH11 sensor, Arduino Uno, LCD display, Wi-Fi module (ESP8266), and think speak (IoT) in their project. The proposed method is an effective way to achieve success as it provides a practical way to record real-time weather data and supports subsistence farmers dependent on the country's climate. Through constant monitoring of embedded devices, the environment can protect itself (smart environment). To collect and analyze information, it is necessary to place sensors in various locations within the environment. By placing sensor devices throughout the environment and allowing them to communicate with other objects via the network, we could animate the environment. Wi-Fi is utilized to provide data analysis findings and collect information for the user.

Table 1:Advantages and disadvantages of weather monitoring system

Advantages	Disadvantages
• Accurate Readings • Ease of Use. • Enhances Home Maintenance • Makes Your House Smarter • Provides Accurate Forecasting.	• This program is only intended for small spaces; and • The cost of the computers needed to execute millions of computations is very high.

The weather is predicted by current weather systems, which create abrupt changes in the prediction with a small delay. The Arduino Uno board is connected to the CPU via a data cable. The temperature sensor and DHT 11 are on the circuit board. The sensor is also connected to the Arduino Uno board via a serial cable connection. Then in the Arduino software (IDE), download the required program and read the temperature and humidity displayed on the LCD screen as follows. The environmental data sensor records temperature and humidity. The hardware display readings and temperature and humidity readings are shown below in Figure 2.1[1].

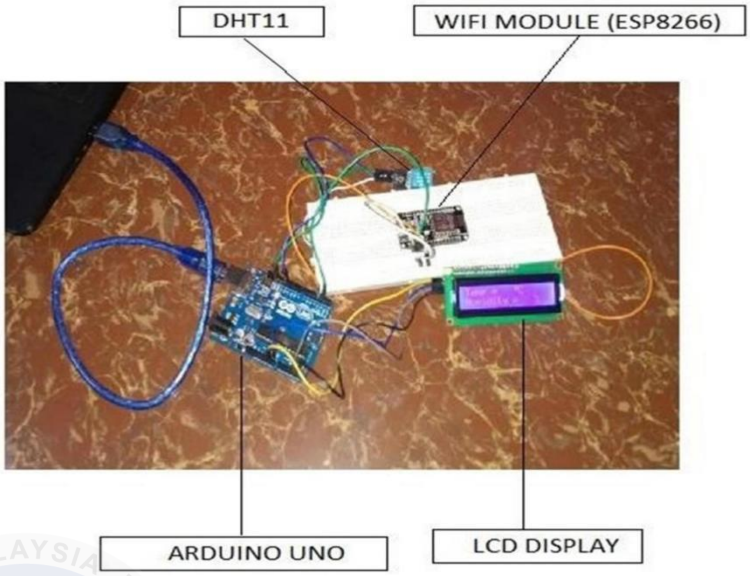

Figure 2.1 Prototype of DHT11 and Arduino connection

2.3.2 Design of IoT based Weather Monitoring System[2]

An IoT weather monitoring system design is outlined in the research project "Design of IoT based Weather Monitoring System" at the Department of Electrical and Computer Engineering, North South University Dhaka, Bangladesh in 2022[2]. Thanks to the advanced method of monitoring weather conditions and the helpful results provided by the proposed system, we are able to determine the weather at any given time and location. Through the Internet of Things (IoT), designers have the ability to connect to any section of a network and attain the best outcomes. We keep track of temperature, humidity, barometric pressure, and rainfall.

The concept of project work explains how the components are put into action and their restrictive usefulness. At first, all components are powered up with the necessary +5 volts. To obtain an accurate temperature reading, we utilize the DHT11 temperature sensor. a small detail regarding the weather. Temperature and humidity can be used to predict climate without additional data. The collected data is serially supplied into a computer, which connects to the

Arduino framework via the com port. A content record contains the data. At that point, the imported data has been sorted and arranged before being plotted on the tables. The data is presented in a realistic way on the maps. The visual patterns show the climate activities of the major region. This serves as the primary goal of the display work.

The DHT11 sensor transmits humidity measurements for indoor temperature. The DHT11, with analogue output, is connected to the Arduino micro-controller through the A0.0 analogue input. The DHT11 sensor is composed of three pins. The dht11 sensor observes or collects additional data, such as the dew point, heat index, as well as temperature and humidity. The focus is on the temperature where cement transitions to water droplets, while the heat index is the warmth felt by human skin from the environment, particularly important in highly humid areas. Furthermore, even with the lower temperature, the body still feels damp. The topic of conversation is the wet material that is stickiness. The usual situation involves the dialogue being high and humid, causing us to perspire.

Moreover, a Bmp 180 sensor was utilized. This device determines the weight of the discussion. The amount of weight felt within the surrounding area is dependent on the atmospheric pressure. Overall, the device is highly useful in high altitude situations as it provides a close estimate of the region's climate tilt by combining recorded elevation data with additional natural information, allowing us to monitor elevation changes as altitude increases or decreases. The BMP180 is composed of an E2PROM control unit, a piezoresistive sensor, an analogue to digital converter, and a serial I2C interface. BMP180 offers raw pressure and temperature readings. The microcontroller transmits an initial sequence to commence a temperature or weight examination.

We utilized a FC-37 rain sensor that is divided. The rain sensor module is a simple tool used for predicting rain. When the raindrop breaks through the turbulent clouds, it can function as a lever to regulate the speed of rainfall. The component features a separate control panel, rain board, control pointer driven (D0), and an integrated potentiometer for enhanced output sensitivity adjustment to assist with comfort. The analogue yield was employed to identify decreases in the total amount of precipitation. The control-driven light on the rain sensor illuminates when activated, along with a better performance LED. The Driven, powered by the 5V control supply, will be activated if the DO output is high and there are no raindrops on the receiving board. Once the water level decreases and the dissolved oxygen production reaches a certain threshold, the indicator switch will activate. Remove the water droplets and discard the excess amounts until the initial state is restored.

Sensor readings are displayed in a web browser when a smartphone or computer is linked directly to the NodeMCU microcontroller, typically yielding precise results regardless of whether the tablet is used indoors or outdoors. Using the climate system, we can collect temperature and humidity information, and our results align with historical and present data. The outcomes can also be shared on websites that enable users to access the Internet from a secure range using a PC or smartphone. Our objective was to create a device capable of detecting weather conditions in real-time and making climate predictions independently of human error by monitoring the components of the weather-making center. Yet, even if the average error size decreases in a specific system, significant mistakes can still happen during any presentation with any specific guidance.

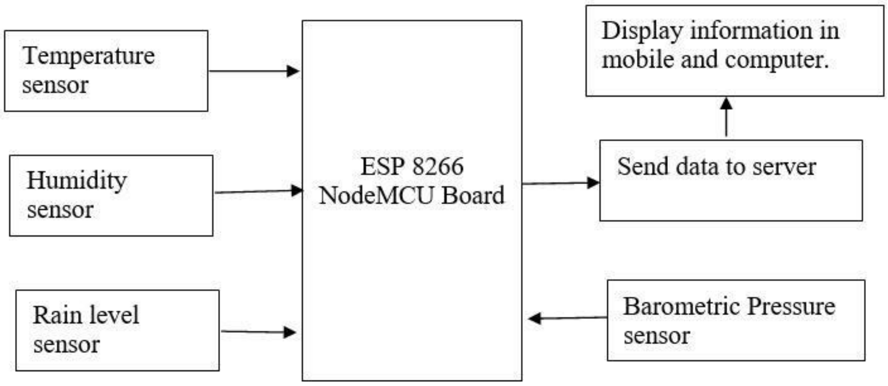

Figure 2.2: Block Diagram

Figure 2.2 displays a pie chart that outlines the whole system. The system includes various sensors (temperature, humidity, rain, barometric pressure) that collect real-world data and transmit it to our NodeMCU ESP8266 microcontroller [2]. After that, the data will be transferred to a server by our board through the internet, allowing us to access it on our computer and mobile devices using browsers.

2.3.3 IoT Based Weather Monitoring System Using Arduino-UNO[3]

Amity University's "IoT Based Weather Monitoring System Using Arduino-UNO" project [3] . In order to establish a massive information base framework that depends on the credits used to create the provided information, the proposed system offers two-study fields- based control frameworks and an information collection technique. As shown in Figure 2.3 the essential components chosen here are based on the sensors that are heavily used to build the foundation for organising a successful climate monitoring project. Here, the temperature and humidity data are measured and assembled using the recommended sensors. Weather Monitoring suggests a system that uses a portable application to gradually screen the climate. The Arduino UNO provides this simple or less expensive platform for connecting all these electronic devices and different kinds of sensors via the internet. The main goal of the project is to record climate, which can be efficiently observed remotely with the use of an Arduino

UNO and the Internet of Things. Customers will benefit from a more direct, reliable, and efficient method of screening climate and its many boundaries.

The main goal of this project is to turn on an electronic device or association that can detect, measure, and restore temperature and humidity. Data will then be sent to a cloud or other location for analysis. Here, the Arduino Uno can serve as the basic brains of the construction by acting as a microcontroller. To set up your Internet affiliation, we specifically require a Wi-Fi module while using the Arduino as a microcontroller. Furthermore, the DHT sensor, the best sogginess sensor in its class, needs to be included into the structure to detect variations in temperature, suddenness, and wetness at a specific location. The sensor continuously monitors temperature variations and transmits information to the microcontroller. The microcontroller transfers the data to the cloud for storage and analysis.

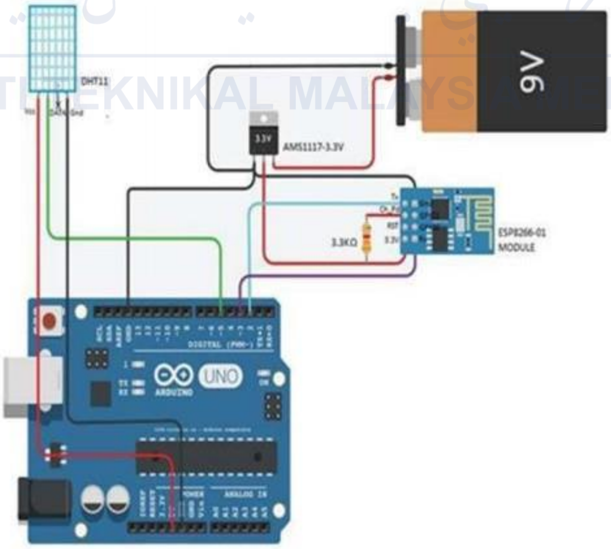

Figure 2.3: Arduino UNO Circuit Diagram

This system architecture is designed to monitor air temperature, humidity, light intensity, and spot conditions to create an intelligent environment through remote management. The planned framework exhibits greater versatility and distributive qualities while examining ecological boundaries. A four-level model is used to assess the engineering plan. the operation of modules designed for climate verification. This model has four levels. Level 1 consists of the overall climate, Level 2 includes all sensors, Level 3 includes dynamic information and recovery, and Level 4 includes an extreme environment. Additionally, engineering is depicted in the figure.

Level 1 provides information regarding the limits and particular location that must be monitored for climate boundaries. Level 2 arrangements use sensor devices with varying attributes, highlights, and control over each sensor's device based on its affectability and detecting range. Between level 2 and level 3, detecting and controlling based on the circumstances, such as edge esteem that we set, periodicity, alert or LED messages, and so on the analysis of the boundary limit values in normal working settings is done considering the two-layer information investigation and previous experiences. Level 3 describes the acquisition of data from operational sensing devices and is also dynamic [3].

2.3.4 Internet of Things (IoT) Based Cost Effective Weather Monitoring Station[4]

The Rajiv Gandhi Institute of Petroleum Technology in Jais, India, has proposed a project titled "Internet of Things (IoT) Based Cost Effective Weather Monitoring Station"[4]. This affordable weather station presents a novel and useful method for tracking various locations' weather and disseminating meteorological information globally. The primary characteristic of the suggested work is the addition of monitoring the Air Quality Index and sending the data to a cloud server. The Internet of Things, or IoT, is a system that collects data from a network of different devices and uses it for different purposes.

Every sensor in the system is connected to the Arduino Uno microcontroller, which serves as the primary processing unit for the overall setup. All the data from the sensors is sent to the microcontroller, which then transfers it to the NodeMcu (ESP8266). The data is uploaded by the Wi-Fi module to the Things Speak API, where it is viewable in a graphical representation.

The various sensors that are linked to the Arduino Uno provided the data that was gathered. The application programming interface (API) of Things Speak will receive the gathered data. All the data is shown on the web server pages as graphs, which make it easier for us to monitor and comprehend the data. The temperature, humidity, CO2 level, rain detection and intensity, and air quality index (AQI) are all shown by the server. The information is kept on cloud servers and can be retrieved and viewed for analysis and forecasting purposes. temperature, humidity, air quality index, CO2 concentration, and the presence of rain at regular intervals.

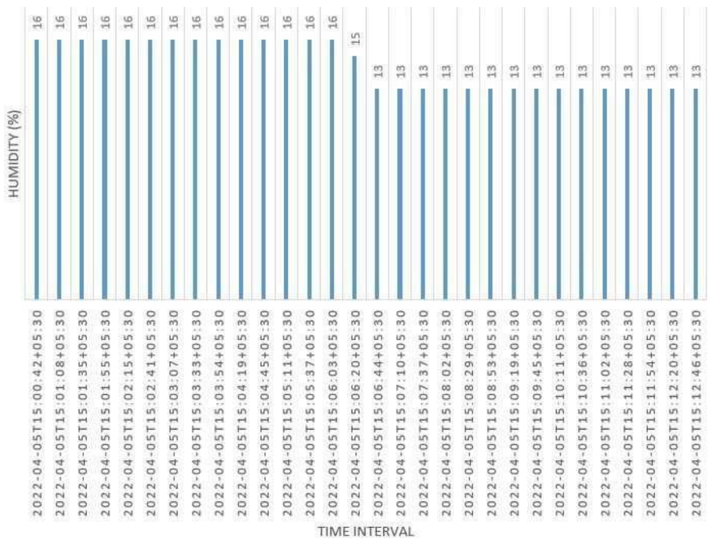

Figure 2.4: Humidity in % vs Time

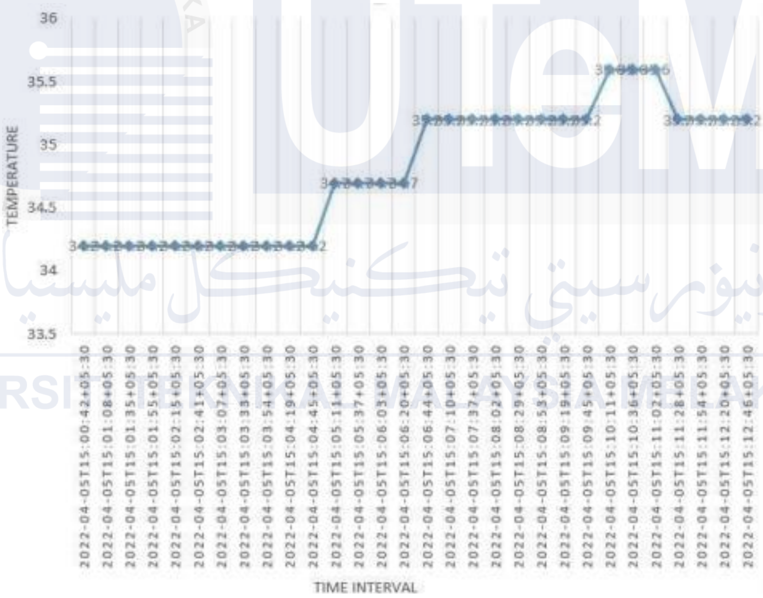

Figure 2.5: Temperature in °C vs Time

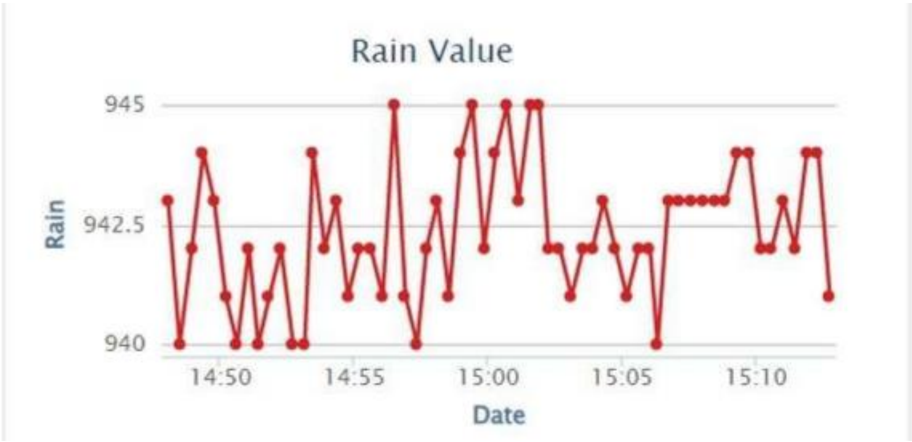

Figure 2.6: Rainfall Data Analogue Value vs Time

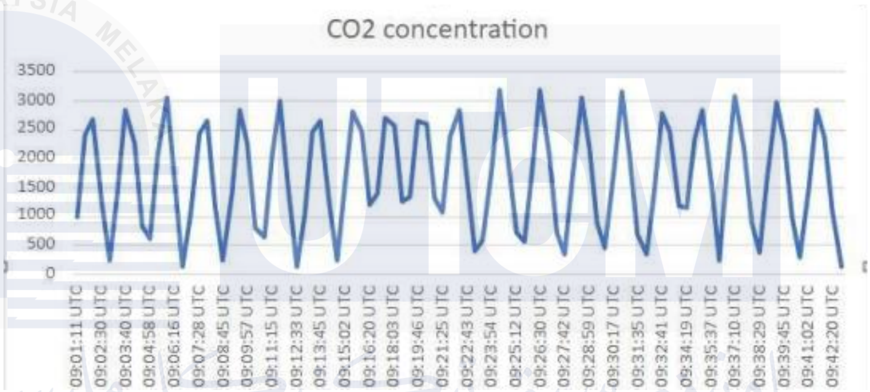

Figure 2.7: CO2 concentration in ppm vs Time

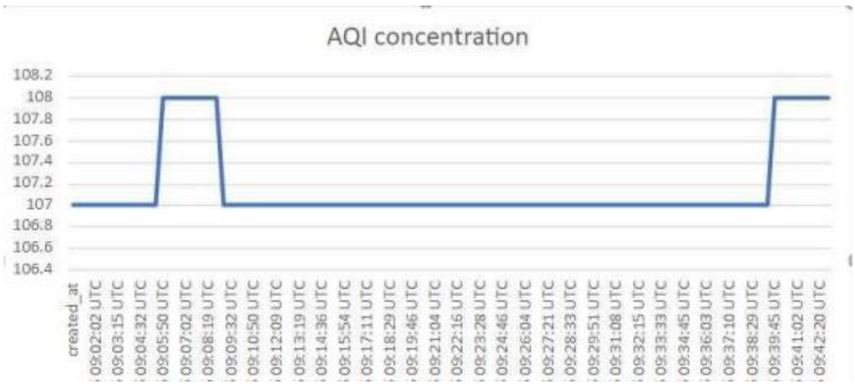

Figure 2.8: Air Quality Index (AQI) in ppm vs Time

Figure 2.4 to 2.8 graph displays the data easily by examining the temperature and humidity data, we can quickly determine the value. The graph's rain value can be interpreted as follows: if the number drops below 525, rain has begun, as indicated by the red light that will turn on. By taking 525 as the baseline between dry and rainy conditions, we may also determine the intensity of the rainfall.

A portion of the data from the lengthy 12-hour measurement of the CO2 concentration is displayed below. Finding the average value of the data displayed by the MQ-135 sensor will be necessary to determine the ppm concentration of CO2. According to the chart, the average figure we receive is 1668.747 ppm, which indicates poor air quality. The smoke sensor's average air quality index (AQI), which indicates moderate to poor air quality, is 107.1667. It is evident that the weather station generates accurate and dependable results. The internet provided the following meteorological statistics for the Jais, Amethi Region: 36°C temperature, 15% humidity, and a poor air quality index No rain. This Internet of Things (IoT)-based weather station displays the meteorological data: 35.2°C temperature, 15% humidity, and a poor air quality index No rain[4].

2.3.5 An Implementation of Internet of Things-based Live Temperature and Humidity Monitoring System [5]

Mahaveer University in Moradabad (2022) reported a study that was completed by the researchers at College of Computing Sciences & Information Technology to develop an embedded system to track various weather parameters for academic research in a laboratory environment. This embedded system includes two sensors, one to measure humidity, and another to measure temperature. The microcontroller captures the data from the sensors and uses serial communication to send it to Google Cloud app (Google Sheets). This module will carry data to the cloud in case the values surpass the set level. If required, through the GSM module, we can also access the data on a mobile device. To understand how the proposed system is realistic, the values from the device are compared to values from other resources and the data from the National Centre for Metrology is compared [1].

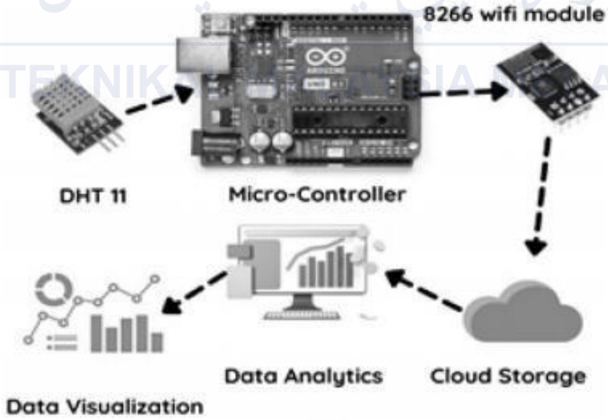

Figure 2.9: Proposed System

The author suggests a wireless method for temperature and humidity monitoring and controlling utilizing DHT11 sensor and GSM networks. It is an economical sensor. The DHT11 sensor functions at -20 to 60 degrees Celsius. The relative humidity measuring range is 5 to 95 percent with an accuracy of 5 percent. Moreover, the temperature ranges between - 20 and 60 degrees Celsius with an accuracy of 2 degrees Celsius. The DHT11 sensor has attracted the interest of consumer electronics and industrial automation creators due to many appealing features. One of the significant tasks of a moderated weather station is to calculate the temperature and humidity.

In the field of biomedicine, it is important that the production of pharmaceuticals and the cell culture processes like that of plants and cells are monitored for temperature and humidity, the patients will have to have predetermined housing conditions so they will benefit from the HVAC system, the Sensor module for measuring both temperature and humidity by using a sensor combination. The module has calibrated a white digital signal output, that has two devices, the DHT 11 which is a popular sensor that is used by the Arduino for measuring both temperature and humidity by the DHT 11, this combination made an excellent reading of the temperature and humidity. With the data, the Arduino is able to process and display it for the user. This module has been able to give an output to the application with actual temperature and humidity readings for balanced living and horticulture from which agriculture may provide more junior readings and a humid environment for the growth of more junior plants and cells. The sector is punctuated for its need for sensitive humidity, on two main fronts - clinical, with the potential for mold outbreaks, and virus proliferation. Therefore, this is going to be a substantial topic to investigate in healthcare settings for hospices and residential homes for the infirmed. With the rapid increase in changing climate, it is necessary to keep current on the current hazardous weather and to be prepared for a rapid response to the environmental disasters that result from the changing climate in the world at present.

The study focused on the introduction and evaluation of a low-cost weather monitoring system with operation capabilities, on-device display, affordability, small size, effectiveness, and portability. The weather monitoring system used an ATmega328 microcontroller, sensors, a monochrome 1.3-inch 128x64 OLED graphic display, real-time data logger on Google Sheets, and short message service (SMS). The system developed in this study presented a beneficial weather monitoring system in terms of its operation, on-device display, cost-effective system, small form factor, versatile operation, and portable system. The developed weather monitoring system had a variety of weather phenomena sensors. The study's results were consistent with several sources. The potential customer sectors for this device would be farmers, travelers, airline personnel, scientists, institutions, and it would have relevance to other publics to monitor, analyze, or predict the weather to reduce weather-related disasters in our world [1].

2.3.6 Weather Monitoring System using Blynk Application[6]

The paper "Weather Monitoring System using Blynk Application" by Mohd Hakimi Bin Zohari and Mohamad Farid Bin Johari (2019) introduces a weather monitoring system using the Blynk app. It centres around an Arduino UNO microcontroller linked with a WiFi- WeMos ESP8266 wireless weather monitoring system. Three sensors are utilized by the system to observe weather conditions: temperature, precipitation, and carbon monoxide. The Blynk application then displays all the information. The project utilized the Arduino UNO microcontroller, WiFi-WeMos ESP8266, DHT 11 temperature and humidity sensor, MQ-7 carbon monoxide sensor, and rain sensor in its development. It functions effectively for observing the weather at any place, at any moment.

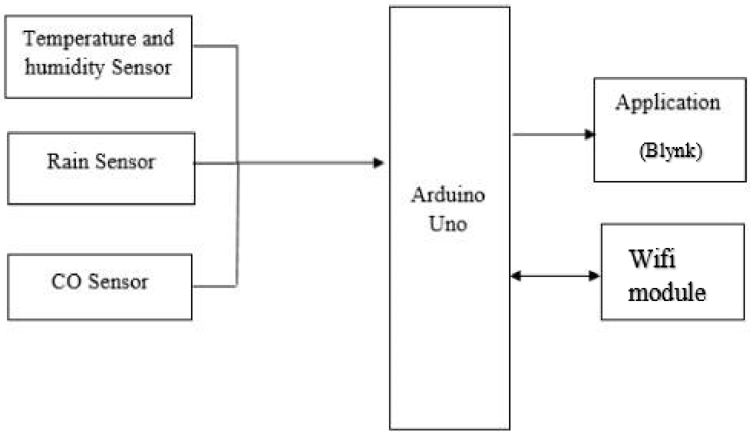

Figure 2.10: Block diagram of Weather Monitoring System using Blynk

The central element of the system, as depicted in Figure 10, is the Arduino microcontroller, which is connected to every other element. The Arduino microcontroller's analogue input is linked to the system's sensors. The WiFi-WeMos ESP8266 is connected to the Arduino as well. Every piece of information gathered will be sent to the Blynk app [6].

Arduino and WemosD1 are the main hardware components. The Arduino Uno is a microcontroller board based on the ATmega328P. To ensure the system's stability and proper functioning, all hardware components are being assembled, and different types of data are being analyzed for this project. The analysis section reviews the outcomes of the project. It can help the user evaluate the system's performance and confirm its functionality by examining the system's results and analyzing data. Therefore, it might monitor the external weather conditions. The project prototype in its final state is shown in Figure 2.11, along with the side and overall views of the system.

Figure 2.11: Weather monitoring system prototype

Blynk was used to gather all the data. To establish an account, the individual is required to register initially. The Auth Token will be sent to the email address on record by Blynk.Once the Auth Token is received, the user can use the application to generate their own data including graphs, display values, buttons, tables, and more. The Blynk application's UI is shown in Figure 2.12 below.

Figure 2.12:Blynk application interface

By constructing the prototype model, this project must be completed with the Weather Monitoring System implemented. The system must be turned on in its basic state. The weather at Melaka is tracked using this weather monitoring system. The temperature, the level of carbon monoxide, and the presence of rain will all be detected by the system. The system will upload the data to the Blynk application after it has detected all the data. Anyone can use the application to search for and find the current weather conditions. People can go there without worrying about the quality of the air. The "Weather Monitoring System" project is being developed in two sections: hardware development and software development. Many studies have been conducted on software development [6].

The most challenging aspect of this project is the programming done with the Arduino compiler. The problems that were encountered were numerous. Every time the program is run, an error message always appears. All issues are resolved, though, after the mistakes are identified. There are also a lot of issues with hardware development. The sensor may not always detect the condition. This issue arises when the cable is slack or not completely connected. It must ensure that all the wires are tight and free of any connections before powering on the device.

2.3.7 IoT based Smart Weather Monitoring System[7]

At the Uttarakhand Technical University Pithoragarh Campus (NPSEI), the research study titled "IOT based Smart Weather Monitoring System" states an emphasis on using IoT technology, as it's a combination of electronics and communication technologies. Even though the study is beginning shortly, it is already shaping up, with specialized attention given to examining the common elements of temperature, humidity, light, noise, and air quality. Systems of embedded wireless networking are being created in order to oversee the pollution levels in target areas, or in certain urban activities. The system is being put into practice using a design that consists of the sensors. The sensors are connected with a built-in Wi-Fi microprocessor, to record any changes in the parameter to the range of the penultimate. The main goal is to create a structure framework for the environmental physiological conditions being evaluated by the framework[7].

This study solely investigates the physical IOT devices for visual surveillance and atmospheric sensing by collecting the data of the parameters. The created system utilizes an advanced microcontroller approach in order to convert the analog data detected from multichannel sources of sensors into digital values that are then sent online. The goal of the system is to simplify remote climate monitoring. The text details how the gathered data can be transferred to the internet to remotely observe the environmental conditions and parameters such as humidity, temperature, dewpoint, airpressure, sound, water level detection, light, gases, and soil wetness.

In this paper, we also cover the basic concept of the Internet of Things (IoT) and the applications of IoT, especially in weather conditions monitoring. In our project demonstration, we used an Arduino UNO board to monitor weather statistics and their respective sensors. The sensors are: LDR, LCD, buzzer, Soil Moisture Sensor, rain detection sensor, sound sensor, and MQ2 sensor. The Blynk Apps display the data and send the alert messages and mobile notifications. You can log in to your Blynk Apps using your email and password. By logging in, you can see the different weather parameters on the Blynk Apps. There are different weather parameters displayed on the Blynk Apps. With the embedded system, which is on the internet, the updated data can be accessed from anywhere in the world.

Therefore, machine learning techniques performed better in learning compared to traditional statistical methods without any explicit modifications. The information gathered by the prototype can be utilized for different purposes, like monitoring weather for improving agricultural output and tracking air quality to lessen pollutants in the air. The assignment utilizes the buzzer of the weather station to update the individual at home about the outdoor conditions, along with using Blynk packages to alert the person during extreme weather situations. The Blynk Apps are capable of activating and deactivating sensor devices as well. Furthermore, we can control the submersible pump that is connected to the Blynk Apps to switch it on or off when the soil becomes dry. The system architectural standards incorporate the hardware and software requirements.[7]

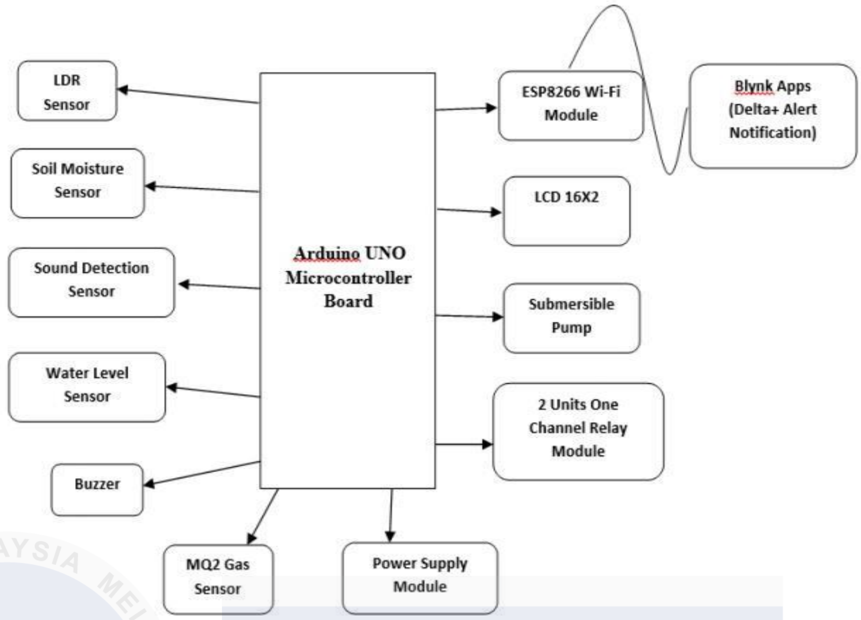

Figure 2.13:Block Diagram of the complete scheme

The ESP8266 Wi-Fi and the Arduino UNO boards are connected to 5 sensors, an MQ2 sensor, an LDR, a sound sensor, a rain detection sensor, and a soil moisture sensor. The overall concept block diagram is shown in Figure 2.13, using an LCD screen for data display and a buzzer for alarm indication, as discussed in [7]. The prototype model of the IoT weather monitoring system project includes the hardware components, including Arduino UNO, ESP8266 Wi-Fi module board as shown in Figure 2.13, and the sensors including LDR, Sound, Soil Moisture, Rain Detection, and MQ2 sensor. Figure 2.13 shows the utilization of an LCD screen, and a buzzer for indication of the alarm [7].

The project will involve the hardware and the software. The hardware involves connecting the sensors physically to Arduino UNO board on the PCB and using LCD, buzzer, and power supply unit. The software involves creating the program in Arduino IDE, displaying data on the mobile app- Blynk Apps, and sending alert notifications.

The Sound Detection board is a small device containing a microphone and several electronic components for detecting different levels of sound. It also has a LDR, photo resistor, or Light Dependent Resistor, a MQ2 gas detector which is a popular detector and belongs to the MQ sensor family, also known as the Metal Oxide Semiconductor (MOS). Figure 2.13 refers to the soil moisture sensors, used to measure the amount of water in the soil. All these sensors are directly connected to the Arduino UNO Board, which calculates and displays specific parameters of the environment in the online computing framework and the alphanumeric display.

The board uses the ESP8266 WiFi module, shown in a block diagram of the system in Figure 2.13. All sensors are connected to the Arduino UNO, has a 10-bit Analog-to-Digital Converter which converts the analog values to digital values ranging from 0 to 1023 (2^10). The resolution represents the number of distinct values that can be generated within the Analog Value.

Connecting the ESP8266 board to Wi-Fi is required to send sensor data to the end user and save it in the cloud for future purposes. Using Wi-Fi to send data to the internet is done periodically. The serial monitor might show incorrectly positioned data like "Connecting to Wi-Fi," warning messages like "Gas Detected" or "Raining," and expressions like "COOL, COOL." Furthermore, the user has the ability to review the weather data in the Blynk Apps. This framework is frequently utilized to display data, including on mobile devices.The Blynk Apps portal is utilized to access real-time data in this particular case. Blynk can be utilized for developing smartphone apps that allow user interaction.

In this scenario, Blynk Apps serves as the interface used to access the real-time data. Blynk can create smartphone apps, which can connect users with microcontrollers. A project can be shared with friends or clients, and they can monitor the connected devices but are unable to edit the project.

The data is transmitted through the Value Display Widget on the Blynk Apps. When the sensors are put into a state where they are not functioning, they all (except the LDR sensor) will display a 0 reading. Blynk Apps are able to manage the sensors. Furthermore, we have wired up a submersible pump and we can also control it through the Blynk Apps. The pump can be operated when the soil is not wet. So, the user will get the real-time weather update displayed on the LCD every 10 seconds and the update from the blynk apps.

2.3.8 Real Time Weather Monitoring using Internet of Things[8]

The title of the article is "Real Time Weather Monitoring by IoT" by the School of Information Technology and Engineering, Vellore Institute of Technology, VIT, Vellore, India.The paper discusses the project whose main aim is to make a custom design of a monitoring system using IoT for monitoring temperature, humidity, and CO2 levels. Data is sent from the sender node to the receiver node within an existing framework. The accessed data is stored into a database on a local computer using Arduino IDE at the receiver node. Serially, the monitoring is done for analysis and also use the API of weather prediction. At last, it has outlined a method for mobile apps in Android to transfer data from the Arduino IDE to the mobile device, which allows monitoring the data and receiving updates from another location.

Finding a device capable of detecting temperature, humidity, gas level, and water level, and providing alerts for adjusting values is a challenge, but essential for creating a precise and cost-efficient system. The main issue that farmers of today's generation are still facing is the insufficient availability of suitable meteorological and environmental monitoring tools for use in our nations. Utilizing their instincts for success, a more cost-effective device is being prepared. Ensuring the safety of all data, including water level, gas level, humidity, and current temperature, to simplify the farmer's understanding of the area's activities. Moreover, our Android app can assist in ensuring precise environmental monitoring, deriving benefits from it, and promptly implementing necessary changes.

The minimum effort we can make is to create a positive impact on the environment and farmers, ensuring a sustainable habitat for future generations to thrive in safely. Individuals deserve access to accurate information about global events, and ignoring this now will lead to regret in the future. This inspired me to create a new Internet of Things system that constantly monitors parameters like temperature, humidity, water level, and gas levels to ensure accurate calculations in the surroundings. This system can help people stay informed and potentially prevent accidents and natural disasters. This affordable weather monitoring system guarantees to provide highly accurate data at any location and time within this plan. Designing an affordable weather monitoring system that is accurate, reliable, and meets all requirements is a challenging task. Equally challenging is ensuring that the user receives the data. Moreover, the calculated system value should align with the actual values.

Nineteen sensors in the current proposed system will be used to continuously monitor the levels of carbon dioxide, light, temperature, and vibration in the working areas. This application has been given the moniker POVOMON. High dependability, network connection, tolerance to the current WLAN infrastructure, real-time data access, and database storing are among the design's most crucial features.

The GUI enables access to this. The main objective is to collect data from the surrounding areas inside the building. In order to accomplish this, it is necessary to have low-power nodes equipped with integrated peripheral components and a customizable sensory setup. This paper utilizes various peripherals such as timers, DC, DMA channels, DACs, and multiple serial interfaces. Formed utilizing the original API network. Various evaluations, such as network analysis and monitoring of air quality, are considered and then stored in a SQL database [8].

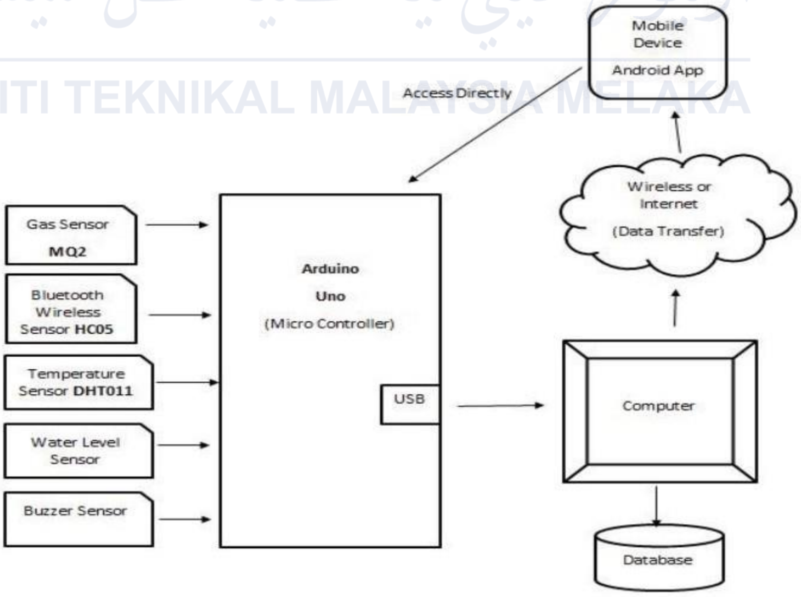

Figure 2.14: System Architecture

The architecture of the IoT Enabled Sensing and Monitoring System is illustrated in Figure 2.14. It utilizes sensors for water level, gas, Bluetooth, temperature, and buzzer [8].

2.3.9 An Approach for Implementing Innovative Weather Monitoring System with DHT11 Sensor and Arduino Uno Tool based on IoT [9]

The suggestion being put forth by Saveetha University is titled "A Method for Installing a Creative Weather Monitoring System Using DHT11 Sensor and Arduino Uno Tool based on IoT." Two teams are participating in the planned development of the weather forecasting system in 2022. The ARDUINO UNO and RASPBERRY PI devices are the initial selection utilizing IoT for a creative weather monitoring setup. Each group consists of 10 samples, and the overall sample size is calculated based on the historical study results from kaggle.com,with a set significance level of 0.05%, power of 80% using G Power, a 95% confidence interval, and enrolment ratio[9].

The Proteus PCB is used to mimic the system. After feature extraction, tabulation is performed. Before feeding the temperature and humidity dataset into the internet of things model, it must be processed from Kaggle. The cleaned dataset is provided for testing and training. Data standardization, mean or median value substitution for null values, and missing data removal are examples of data processing. The Raspberry Pi receives the preprocessed dataset with features as input. Out of the total sample size, 70% of the data is provided for training, and the remaining 30% is provided for testing [9].

2.3.9.1 Utilizing a Raspberry Pi device in a weather monitoring system [9]

This system is capable of tracking and refreshing weather conditions using the internet. The framework screens three borders: temperature, humidity, and precipitation, displaying them on an LCD and updating them through the Internet of Things network. The customer can acquire information about the weather in a specific region by reading on the screen. Ultimately, this model proves beneficial for assessing the regional weather conditions. The system includes an LCD screen, a bell, a thermometer, a rain sensor, and a Raspberry Pi for electricity. Once the system is activated, WIFI is utilized to link the system to the website. The framework monitors three parameters: temperature, moisture content, and precipitation. The framework, which monitors and adjusts IoT status in dry regions, is useless in the face of changing climate. When droplets are identified by the system, it shows the amount of precipitation increase. The value is changed when the temperature increases. The client can monitor the local weather conditions from any remote location.

2.3.9.2 An Arduino Uno instrument found in a weather tracking system[9]

The Internet of Things (IoT) plays a significant role in the lives of individuals and has established itself as a fundamental structure to sensor networks in our environment. It is perceived as a layer design, which generates raw information termed data, which is stored and collected on cloud platforms and used by applications. This study discusses implementing a layer design consisting of temperature and humidity sensors, and the outcome of the implementation, which is used to activate temporary corresponding responses, which is exemplified by temperature control and cooling. The information is then sent to the cloud based on the data from a layered design via an Android application and presented to end users. The layered structure includes sensors;

a DHT11 sensor, ESP8266 Wi-Fi module and an Arduino UNO board for transmitting IoT data to interact with a public IoT API platform, an API is built to connect the Cloud platform to yield results to the end users. The outcomes of the trial show the capabilities of the design.

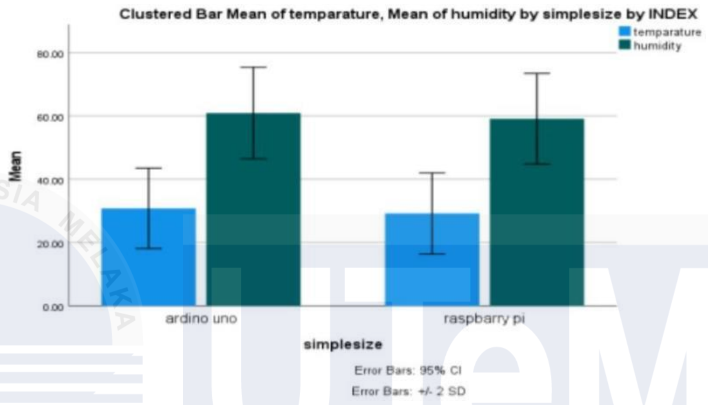

Figure 2.15: Examination of Arduino Uno tool using DHT11 sensor compared to Raspberry Pi tool in mean accuracy.

Figure 2.15 displays the Arduino Uno has a higher average accuracy than the Raspberry Pi, and the Arduino Uno also boasts a lower standard deviation compared to the Raspberry Pi. Comparison of bar graphs between Arduino Uno tool with an accuracy of 95.856% and Raspberry Pi tool with 88.678% accuracy. Algorithm on the x-axis, mean accuracy on the y- axis with standard deviation plus or minus one standard error.

In our research project, we found that Arduino Uno has higher precision than Raspberry Pi tool. The weather monitoring system forecast also shows Arduino UNO tool had an accuracy of 94.987% and loss of 5.013% while Raspberry Pi tool had accuracy of 88.678% and loss of 11.322%. In a study, the results were also similar, with Arduino Uno having 87% precision and Raspberry Pi having 80% precision in terms of Internet of Things. The Internet of Things is an update on the data that will be automatically updated, making it unnecessary for an individual to update it. The results we had in this study were the comparison of Arduino Uno (82% precision) and Raspberry Pi (80%). In this study, no conflicting results.

The drawbacks of the weather monitoring system based on Arduino Uno include the need for manual data input, increased time consumption, and higher power usage compared to typical systems. In the future, sensors, microprocessors, and microcontrollers will be incorporated into the circuit design process to address this issue. The results produced by the Arduino Uno are more accurate, significant, and efficient compared to those of the Raspberry Pi [9].

2.3.10 IoT-based Automatic Weather Station [10]

The Vishwakarma Institute of Technology in Pune, India, has put forward a research paper called "IoT-based Automatic Weather Station." The proposed system described in the research paper is designed to establish an Internet of Things (IoT) capable of monitoring temperature, humidity, barometric pressure, and rainfall around the weather station's designated site. Various sensors on the weather station collect the data, which is subsequently uploaded to a webpage after being detected. The Node MCU's IP address can be found by accessing the serial monitor in the Arduino IDE after uploading the program. By entering the IP address, any approved web browser can present the collected weather data on a simple GUI. The recommended method was successful, resulting in accurate retrieval of 95% of the parameters [10]. The system is extremely user-friendly and easy to install, plus it is very affordable. Figure 2.16 shows the circuit diagram of the proposed system, The circuit diagram is made on the software named Fritzing.The testing of the system was successful, and all the requirements were met[10].

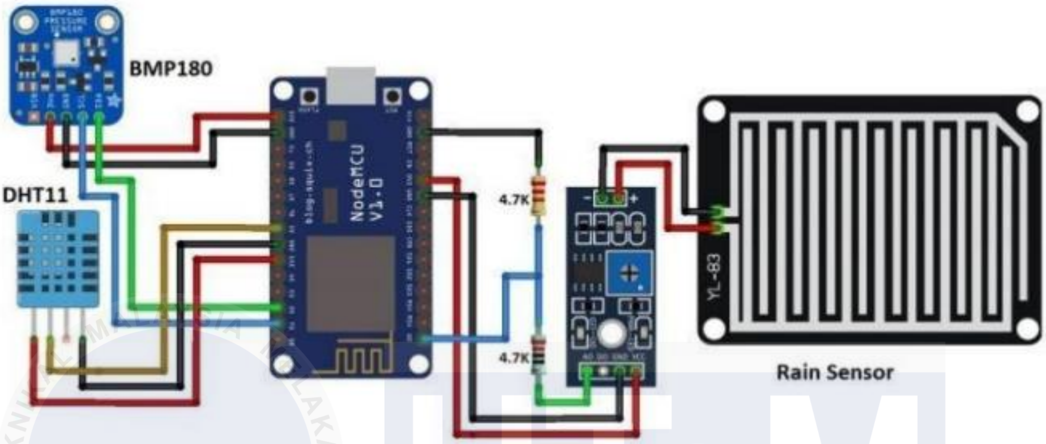

Figure 2.16: Circuit diagram of the proposed system[

2.3.10.1 The following describes how the suggested system operates[10]

Using the circuit schematic as a guide, assemble the circuit first. Now insert the Node MCU board's main ino file. Upon uploading the program and providing power to the microcontroller, each sensor will function in accordance with its specifications. We will link the hardware and software that we used to establish our automated workstation. Once the sensor detects the specified parameters, the weather station will start collecting data from its environment. Retrieve the ESP8266 IP address shown on the serial monitor upon program completion and paste it onto a web server. The circuit diagram of the recommended system, designed with Fritzing software, can be seen in Figure 2.16[10].

Therefore, the setup consists of an ESP8266, a rain sensor, a temperature and humidity sensor, and a BMP180 pressure sensor. Every sensor effectively performs its designated function, and the project functions as intended in its entirety. Our weather station showed all the data rather properly, albeit with a few errors particularly with the temperature, which steadies over time before producing an output that was extremely accurate [10] . The system was set up in our institute's IoT laboratory and allowed to operate under standard room temperature and weather parameters. This allowed the system to be successfully tested in an environment with typical atmospheric conditions and to function as intended.

2.4 Comparison with Related Works

Table 2:Comparison of advantages and disadvantages with Related Works

AUTHOR	TITTLE	ADVANTAGE	DISADVANTAGE
[1]	Arduino Powered Smart Weather Monitoring System	Accurate Readings	Limited Distance
[2]	Design of IoT based Weather Monitoring System	Low cost and user friendly	Network needed
[3]	IoT based weather monitoring system using arduino-UNO	Efficient	Work for a small or specialized area
[4]	Internet of Things (IoT) Based Cost Effective Weather Monitoring Station	Budget-friendly	only the present region
[5]	An Implementation of Internet of Things-based Live Temperature and Humidity Monitoring System	affordable	inaccurate monitoring
[6]	Weather monitoring system using blynk application	Always tracking the weather and locations.	The condition may not always be detected by the sensor.
[7]	IoT based Smart Weather Monitoring System	Alert for emergencies	network required.
[8]	Real Time Weather Monitoring using Internet of Things	high accuracy	Short distance work conditions
[9]	An Approach for Implementing Innovative Weather Monitoring System with DHT11 Sensor and Arduino Uno Tool based on IoT	More accurate	consumes more energy
[10]	IoT-based Automatic Weather Station	Pocket-friendly	restricted range

2.5 Summary

This chapter briefly describes the literature review of the project. Besides, this chapter describe about previous projects that have the same objectives but using a different concept such as the sensors or type for the microcontroller board used. It is for the reference to develop this project and describing the identification of the equipment to be used. Then, there are differences with the current project that will be developed that is portable. While the system is to analyse the situation for outdoor environment, this project is used in limited area such as room or farm.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The process Designing and developing the weather monitoring system with the ESP32 and mobile application interface requires work on one side and testing and validation on the other. During this stage, concepts need to be made tangible. The hardware model needs to be established, and it consists of the NODE MCU ESP32, temperature and humidity sensors, a rain sensor, a barometer pressure sensor, and an application. In order for the system to operate as per the flow charts, this phase also requires a minimum level of hardware and software compatibility. The following sub-processes, which are limited to evaluating the efficacy and efficiency of the system, are part of the testing and validation. User, integration, and unit testing ensure that the weather monitoring system works properly.

3.2 Project Overview

A project plan shows in Figure 3.1 which allows the project to be a success by following the flow chart below. If at the end of the project schedule, we keep up with it meaning that this project can finish on time.

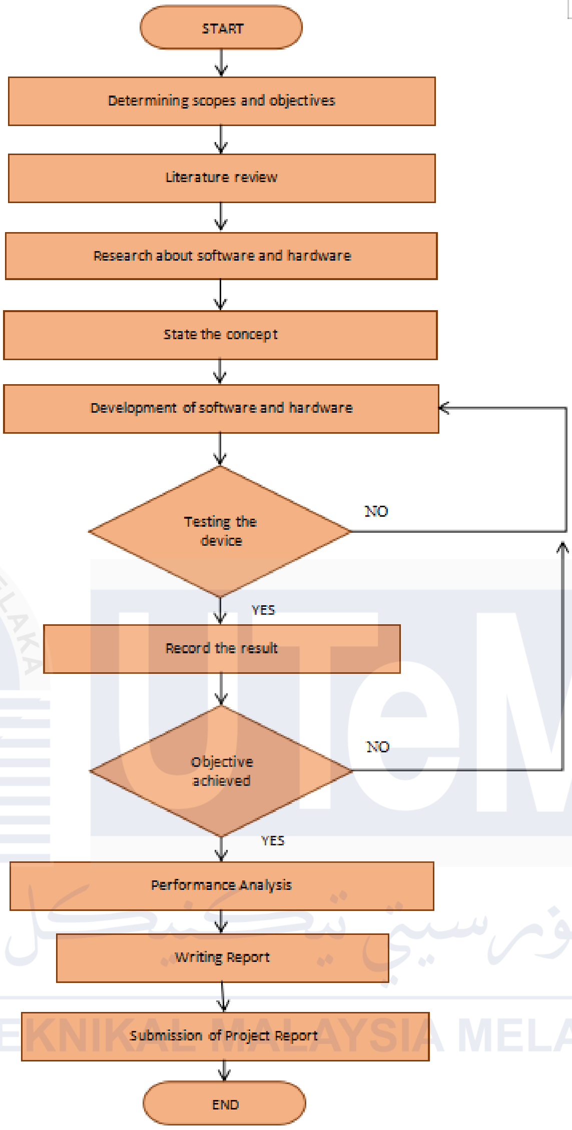

Figure 3.1:Flowchart of full project

3.3 System Hardware Design

The first planning done is the hardware design before it is developed. Hardware design ensures that this the project is clear on what are the objectives and goal of the project to make sure this project is successful. When the time of developing the hardware design, the next step to take to complete the project.

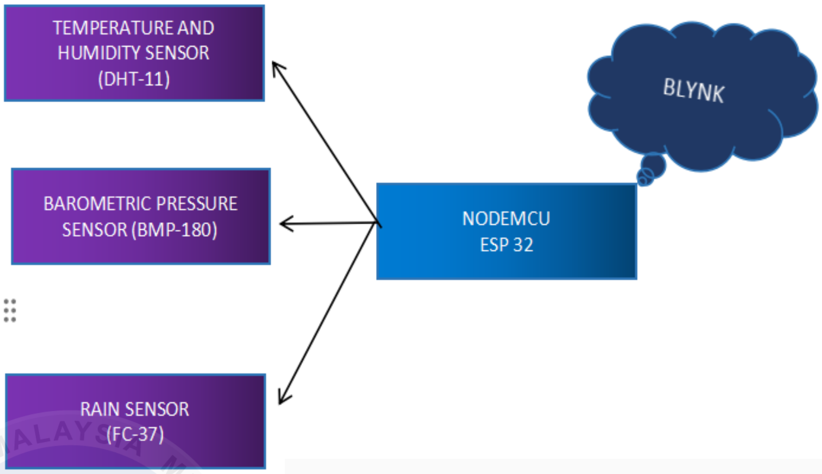

Figure 3.2:Project block diagram

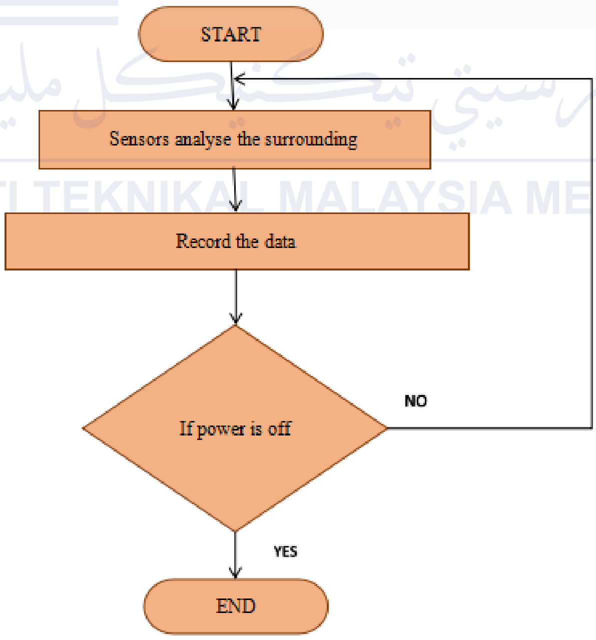

Figure 3.3: Flow of devices

3.3.1 Temperature and humidity sensor (DHT11)

Recognized as an influential digital sensor for measuring temperature and humidity, the DHT11 sensor offers dependable measurements with easy interfacing. The sensor is capable of delivering measurements within a temperature range of 0-50° C and a relative humidity range of 20-90% RH, with accuracies of ±2°C and ±5% RH, respectively.As a example of how The DHT11 sensor is functional in the voltage range of 3.3V to 5.5V and works on a 1- wire digital interface which is very easy to connect to a microcontroller or any other digital system. The sensor you can count on to give you results in less than five seconds, this makes it suitable for applications like home automation, weather monitoring, and environmental sensing. With its simple connection features, DHT11 sensor finds applications in many projects from low end hobbyist projects upto basic climate monitoring in professional equipment.

3.3.2 Barometric pressure sensor (BMP-180)

BMP-180 sensor finds predominance in applications requiring accurate atmospheric pressure and a high precision barometric pressure sensor, the BMP-180 is often used. The barometer is designed to work within a range of 300 to 1100 hPa pressures and gives a temperature reading as low as -40 to 85°C with no compromise on accuracy. BMP-180 not only offers small footprints and low power consumptions but also interfaces with microcontrollers as well as digital systems through I2C or SPI buses. BMP-180 is used extensively in activities that pose the constraint of creating compact and portable devices like altimeters, weather stations, GPS modules and smart phones. These devices require accurate altitude data yet at the same time introducing a way of improving navigation system’s accuracy.

3.3.3 Rain sensor (FC-37)

The rain sensor FC-37 is a basic but powerful tool, which has been widely used to detect rainfall. It consists of a rainfall sensor plate and a sensor control module.The rain sensing plate is set up to identify the rainfall by using the electrical circuit of the plate that closes when raindrops appear on the plate, causing the resistance to change that sends a signal to the module. The sensor produces both an analogue signal as well as a digital signal. The quantity of rain is detected in the analogue signal itself, and the digital signal is used to identify whether the rain is happening or not. The FC-37 operates by 3.3V to 5V, suitable for interfacing the sensor to microcontroller because of the voltage requirement; as an effect, it can be useful for weather monitoring systems, automatic wiper control in vehicles, or smart irrigation systems. Many people choose the FC-37 rain sensor because it can detect the rain easily without making it complicated to use, and it is reliable.

3.3.4 ESP 32

A powerful microcontroller with Bluetooth, Wi-Fi, and BLE connection, the ESP32 is ideal for smart devices and the Internet of Things. It has many GPIO ports for connecting to sensors and actuators, an inbuilt ADC and DAC for processing analog and digital signals, and a dual-core processor to improve multitasking efficiency. It enables seamless peripheral connectivity by supporting PWM, SPI, I2C, and UART protocols. It is perfect for battery-operated applications due to its low power consumption and deep sleep capability, and its robust security features ensure safe data transfer.

3.4 Software Design

The software implementation step is included in this section of the design, as illustrated in Figure 3.4 of the creation of the weather monitoring system with an Arduino and mobile application interface.

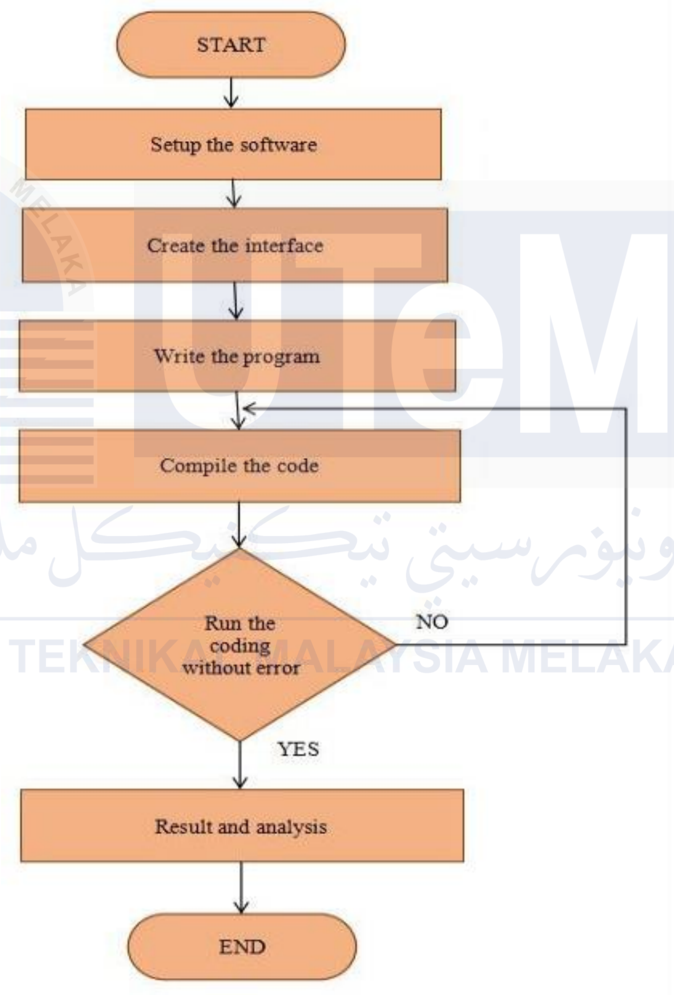

Figure 3.4: Flowchart of software implementation

To implement the software, the Arduino software needs to be installed on the computer by downloading the latest version of the Arduino software from its official website. To integrate the device, it requires the USB cable. The USB cable is used to connect the ESP32 with the computer. Next, an interface should be designed to communicate with the sensor and control the system by microcontroller. After that, the program code must be written and to compile the code and verify whether the program code is correct. If it is correct and did not find any errors, program implementation was found successful, the code can be uploaded to the board. Otherwise, if the program code has an error, and the upload process will stop, and an error alert message will be displayed on the Arduino IDE software.

3.5 Project flow chart

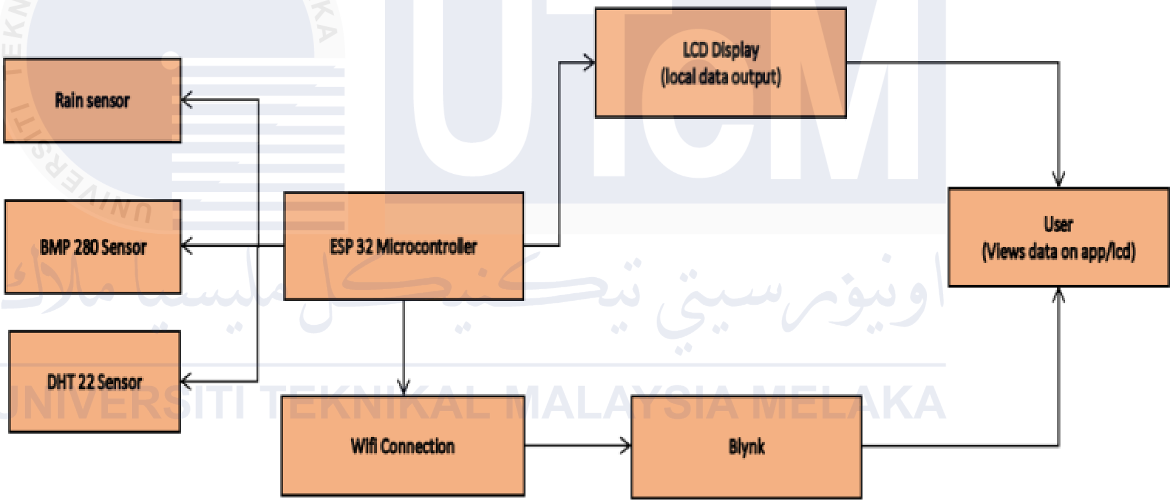

Figure 3.5:Block diagram of weather monitoring system

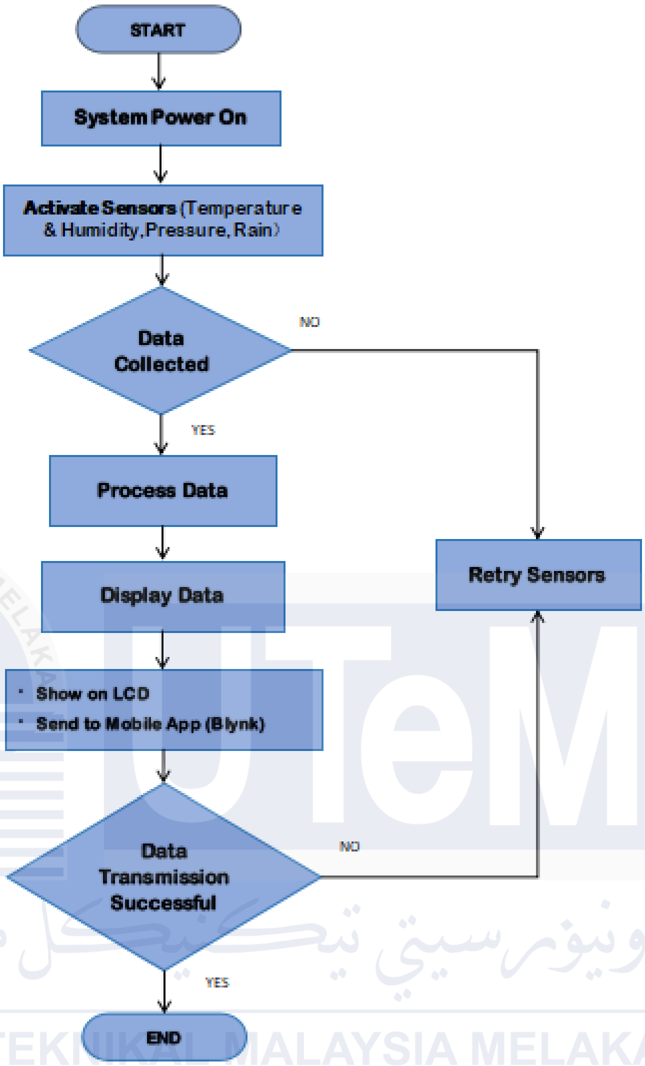

Figure 3.6:Flowchart of weather monitoring system

Figure 3.5 and Figure 3.6 describes a weather monitoring system whereby environmental data is acquired using a set of sensors that show the information to the user either on the LCD display or on a mobile application. At the heart of it is an ESP32 Microcontroller, acting as the CPU. Rain Sensor, BMP280 Sensor-for atmospheric pressure, and DHT22 Sensor-for temperature and humidity-are attached with the ESP32 microcontroller to collect data concerning weather.

This processing from the sensors is then sent to a local LCD display by the ESP32 for immediate use. At the same time, the ESP32 also connects to a Wi-Fi network to enable the option for remote transmission. The system utilizes the Blynk platform to send the weather data to the user's mobile device or application. It ensures that locally, the real-time weather information will be viewed on an LCD while users can also create options to monitor it via the app, depending on their comfort in handling either.

3.6 Summary

This chapter provides an explanation of the project run process, the components to be used, and the identification of the relevant variables. Therefore, various ways are used to determine the project design for example by reading examples, find related journals or related project reports. Searching from the internet is also one of the sources to find the project that is found to be appropriate. Besides that, the findings are also based on the discussion with the supervisor and other lecturer.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the results and analysis on the development of weather monitoring system using ESP 32 with mobile application interface with sensors. Fritzing simulations are used to demonstrate how the project will operate.

4.2 Hardware & Software Configuration Testing

4.2.1 Circuit design

In Fritzing Software, the initial output simulation will be presented in this chapter.The simulation will demonstrate to show how the project will function and gain result. The outcome will be displayed using Fritzing Software.

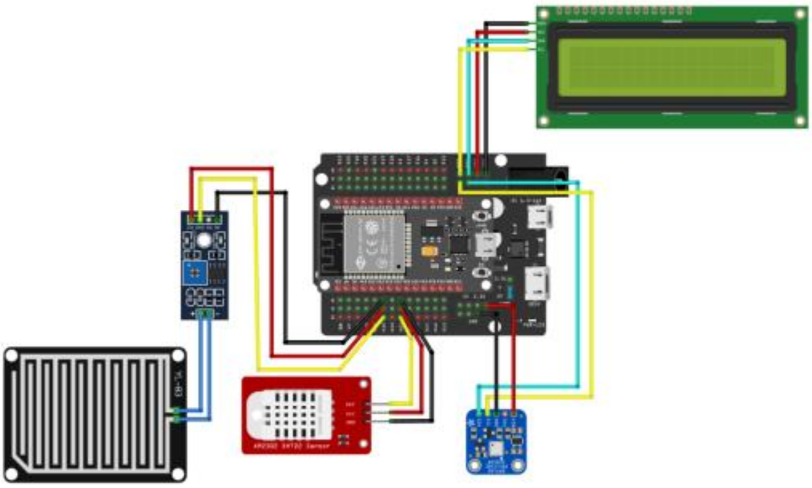

Figure 4.1:Circuit Design for development of the weather monitoring system using the ESP 32 with mobile application interface.

This is the initial simulation done on Fritzing to demonstrate development of the weather monitoring system using the ESP 32 with mobile application interface. As shown in Figure 4.1, the circuitry diagram consists of ESP 32, three sensors, LCD to display the data collected. This chapter presents preliminary findings that were made in the previous chapter. Through simulation using Fritzing including understanding the level of success of development of the weather monitoring system using the Arduino with mobile application interface.

4.2.2 Hardware Setup

This can be achieved by interfacing the sensor nodes and peripherals that are required with the central processing unit, which is ESP32, for the development of the weather monitoring system. One could interface a variety of sensors and components using an ESP32 microcontroller because it was fabricated with multiple GPIOs. The entire system is powered through an on-board USB connection to the ESP32 or through an external battery pack, while sensors and peripherals can be powered via 3.3V or 5V output pins on the ESP32.

The rain sensor responsible for detecting rainfall is connected to one of the analog input pins of the ESP32. This, therefore, enables it to send variable voltage signals, which represent the level of rain back to the ESP32 for processing. The BMP280 sensor used in measuring atmospheric pressure is connected with the ESP32 using the I2C method of communication. This means connecting the SDA and SCL pins of the sensor to the ESP32's I2C-dedicated pins, which are usually GPIO 21 and GPIO 22.

The DHT22 sensor, which will be used for measuring temperature and humidity, is connected to one of the ESP32's digital GPIOs. An LCD module for local data display will likewise be connected to it. Depending on the kind of LCD in use, there's an I2C interface for simplicity or parallel communications in case faster data transfer is required. In addition, key weather parameters are displayed on the LCD, thus providing users with real-time feedback directly from the system.

In this ESP32, there is also a configuration to connect to Wi-Fi so it can send the data that has been collected to Blynk's IoT platform for remote monitoring. Careful consideration has been taken to provide the right connections that give power and ensure communication among the components is stable. With this neat hardware configuration, the system is compact, effective, and capable of acquiring and presenting weather data.

4.2.3 Software

The development of the software for the weather monitoring system involves the programming of an ESP32 microcontroller to collect data from sensors connected, process it, and broadcast both to a local display and on the Blynk IoT platform. The programming and uploading are done to the ESP32 using the Arduino IDE. Essential libraries like the DHT library for the DHT22 sensor, the Adafruit BMP280 library for the BMP280 sensor, and the Blynk library itself are installed to integrate all the hardware components smoothly.

First, the code initializes all sensors and sets up the communication protocols. The BMP280 is configured to communicate with the ESP32 using the I2C interface, and the DHT22 is initialized to read temperature and humidity data through a given GPIO. Besides that, the rain sensor's analog or digital readings are processed to get the rainfall intensity. The LCD display is prepared for the real-time representation of data with respect to weather conditions, using either the Liquid Crystal library or the LiquidCrystal_I2C library, depending on the form factor of the applied LCD module.

The system is programmed to insert the Wi-Fi credentials and the authentication token the Blynk application has provided into the ESP32 for remote monitoring, connecting it to the Blynk cloud server. In the Blynk application, it was configured with various widgets to be able to monitor temperature, humidity, pressure, and rainfall intensity data; therefore, gauges, labels, and graphs are available in this work. The ESP32 was programmed to periodically take readings from sensors, showing it on an LCD and sending that to a Blynk server for real-time monitoring via the application.

The software setup ensures that the system works perfectly: data is acquired highly accurately and communication between hardware and the IoT platform is reliable. Besides, the modular code structure allows for easy modifications and addition of new features if needed. This robust software design is crucial in ensuring efficiency and functionality within the weather monitoring system.

4.3 Results and Analysis

4.3.1 System Operation

Results from the weather monitoring system depicted that it can measure and show all environmental parameters of the ambiance: temperature, humidity, atmospheric pressure, and rainfall intensity. This system was able to integrate all sensors and components by showing data locally on LCD and sending it in real time to the Blynk IoT platform. The rain sensor provided reliable analogue readings reflecting the intensity of rainfall, while the BMP280 and DHT22 sensors gave consistent measurements of atmospheric pressure, temperature, and humidity. Their data was found to be accurate when compared to the reference measurements taken using the standard meteorological devices.

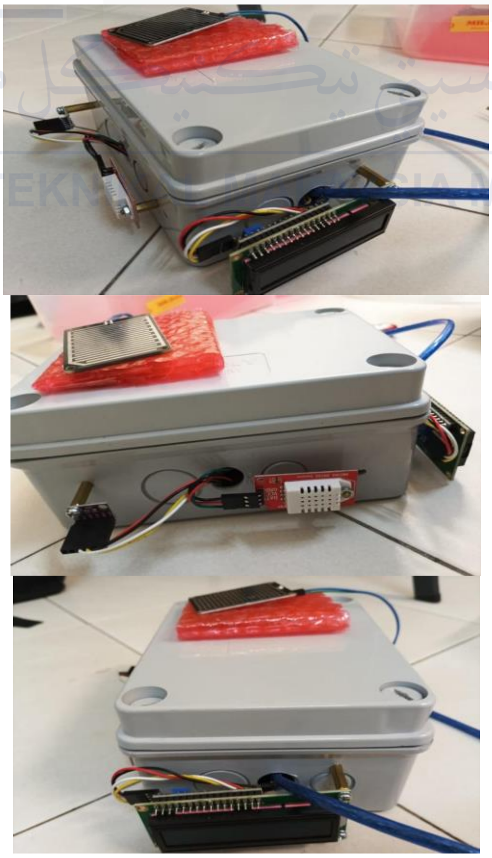

Figure 4.2: Prototype of the project

The performance analysis of the system indicated that it worked well under various weather conditions. Data on the LCD was updated in real time, and users could observe the current weather conditions locally. Similarly, integration with the Blynk platform enabled users to monitor trends remotely through the app, and historical data could be accessed for easy analysis of weather patterns. Wi-Fi connectivity and data transmission to the cloud were stable; there is very little latency in the updating of data on the Blynk interface.

It was a robust system with reliability and achieved most of the goals of its current design in real-time with remote access. Analytical tests pointed out scalability, such as the fact that incorporating further sensors or features is largely possible with minimal modification. These results confirm this could be a low-cost solution with effectiveness for total weather monitoring at both a personal and industrial level of requirements.

4.3.2 Data Analysis

The data analysis of the weather monitoring system was in regard to evaluating sensor reading accuracy, consistency, and responsiveness because of changing environmental conditions. The data included temperature, humidity, atmospheric pressure, and rainfall intensity that were logged and monitored both on the LCD locally and remotely via the Blynk IoT platform. Each sensor output was compared with data from standard meteorological instruments to guarantee reliability. Of these, the BMP280 and DHT22 sensors showed the least deviation; temperature readings gave an average error margin of ±0.5°C, and humidity measured within ±2% relative humidity showed their precision.

Rainfall data from the rain sensor were analyzed based on the analogue output, which had corresponding rainfall intensities. The sensor was able to effectively distinguish between light, moderate, and heavy rainfall. Results were plotted to observe trends that may occur with time. Data transmitted to the Blynk platform allowed further analysis, such as observing weather patterns over long durations and finding correlations between various parameters, like temperature and humidity fluctuations during rainfall events.

The analysis has further revealed that the system updates in real-time data; no considerable delay in the transmission of sensor readings up to the Blynk interface was observed. This means users can comfortably rely on the system for timely weather updates. Secondly, the visualization of data by the Blynk application through graphs and charts enhanced the further understanding of weather trends. In general, data analysis confirmed that the system is efficient in collecting and transmitting precise environmental information, which could be a very useful tool for real-time monitoring and long-term trend analysis.

Figure 4.3: Output on the LCD

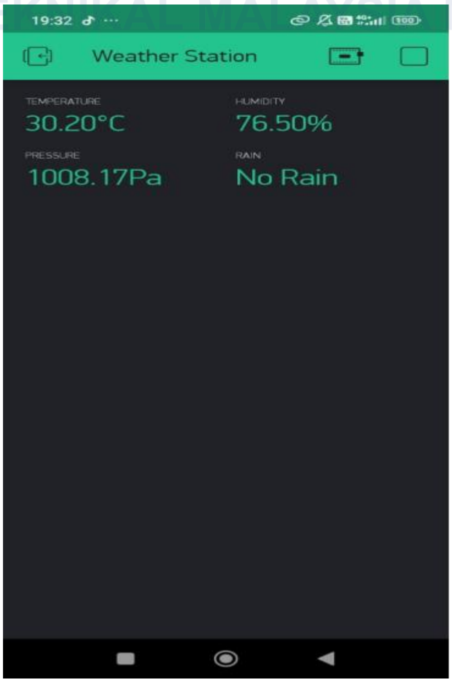

Figure 4.4: Output on the Blynk application

4.4 Testing the parameters

In this testing, the parameters of the devices will be tested at some times to find out the performance of the sensor. Besides, this test also to find the changes that can be detect by the sensor. This sensor will be test at outdoor as this device is a portable and it more appropriate to be placed in a place that is small such as farm or room.

Table 3: Data parameter

DAY	Temperature(Celsius)	Humidity(%)	Pressure(hpa)	Raining or no rain
1	28	86.80	1007.69	Raining
2	30.10	79.40	1009.32	No rain
3	30.20	76.50	1008.17	No rain
4	29.90	77.00	1008.15	Raining
5	30.30	76.00	1008.17	No rain
6	30.00	78.30	1008.28	Raining
7	30.20	79.10	1008.30	No rain
8	31.10	75.40	1011.73	No rain
9	30.10	78.00	1011.53	No rain
10	30.20	77.90	1011.55	No rain

The table 3 showcases a 10-day weather tracking record produced by the weather monitoring system, documenting essential environmental factors including temperature, humidity, atmospheric pressure, and rainfall conditions. The temperature readings stayed stable, varying from 28°C to 31.1°C, indicating standard daytime variations. Humidity ranges were between 75.4% and 86.8%, with elevated humidity associated with wet weather. For example, on Day 1, during the rain, humidity reached its highest level (86.8%), and on other rainy occasions, like Day 4 and Day 6, humidity levels also rose slightly when compared to dry conditions.

The atmospheric pressure measurements varied from 1007.69 hPa to 1011.73 hPa. Generally, lower pressure values were linked to rainy weather, as observed on Day 1 and Day 6, which corresponds with the usual pattern of weather systems, since reduced atmospheric pressure frequently points to stormy or wet conditions. On days without rain, the pressure readings stayed elevated and consistent, indicating clear or mild weather conditions.

This data shows the system's capability to precisely capture and relate weather variables, revealing distinct patterns among pressure, humidity, and precipitation. The system efficiently records changes, offering helpful insights for examining weather trends and aiding applications that require real-time monitoring.

4.5 Summary

This chapter covers the design and background of the development of the weather monitoring system using the ESP 32 with sensors and its development phase along with its results and analysis. The working of this project was shown by simulation software known as Fritzing.The accuracy with which the weather monitoring system records important environmental variables such as temperature, humidity, air pressure, and rainfall is demonstrated by the 10-day weather observation data. Temperature readings showed typical daily variations, ranging from 28°C to 31.1°C. The range of humidity readings was 75.4% to 86.8%, with higher values seen on days 1, 4, and 6 when it rained. Measurements of atmospheric pressure ranged from 1007.69 hPa to 1011.73 hPa, showing that rainy weather was linked to lower pressure readings while clear weather was linked to higher values. Reliable real-time observation and practical insights for evaluating weather trends were provided by the system's successful recognition of patterns among weather variables.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the weather monitoring system efficiently delivers accurate and timely weather data by integrating an ESP32 microcontroller, rain sensor, BMP280, DHT22, LCD screen, and the Blynk IoT platform. Important environmental parameters including temperature, humidity, atmospheric pressure, and rainfall intensity are precisely measured by the system, which displays the data locally on an LCD and enables remote viewing using the Blynk app. When compared to traditional meteorological instruments, the data analysis demonstrated negligible mistakes, confirming the precision and dependability of the sensor observations. Additionally, the system performed well in a range of weather situations, ensuring steady Wi-Fi connectivity and seamless cloud data transfer. Its usefulness and potential uses were greatly enhanced by the addition of real-time visualization and trend analysis features. This cost-effective, scalable, and efficient solution is perfect for industrial, educational, and personal uses and offers a helpful tool for monitoring and understanding weather patterns.

5.2 Potential for Commercialization

The weather station market is expected to grow significantly over the coming years, with several critical drivers prevailing. Companies operating in the market are adopting product innovation and development as a key strategy, keeping in mind the changing tastes and preferences of their customers. Other notable strategies for business growth include entering emerging economies and strategic collaborations or partnerships. In the perspective of future market growth, improvement in the quality of products is highly required through investments in research and development, together with advanced technological enhancements. Additionally, the rapid pace of digitalization and use of sophisticated technology will further open paths for increased growth and innovation in the marketplace in the years to come.

5.3 Future Works

For future work, some recommendations for the project structure are as stated, which is also based on the faced problems and some considerations.

a) Utilizing more sensors

Future research might concentrate on adding more sensors, like air quality sensors, UV index sensors, and wind speed and direction sensors. A more thorough grasp of environmental circumstances would result from these additions. For instance, wind sensors could aid in disaster management by identifying storm activity, and urban air quality measurements could aid in monitoring pollution levels. By giving farmers accurate environmental data, monitoring microclimatic parameters like dew point and UV levels in agricultural fields could maximize crop production.

b) Using Predictive Analytics with Machine Learning

The prediction capabilities of the system would be improved by incorporating machine learning algorithms to examine past weather data. By making it possible to forecast weather patterns and provide early warnings for extreme conditions, this would increase its value for sectors like agriculture and disaster management. Furthermore, by recognizing long-term environmental trends and their consequences, machine learning may aid in the study of climate change.

c) Cloud-Based Data Processing and Storage Implementation

By using cloud-based technologies for data processing and storage, the system might become more scalable and resilient. This would provide centralized monitoring and data access through large-scale deployment across several locations. Real-time and historical meteorological data might be effectively handled and disseminated to stakeholders via the cloud, enhancing decision-making and teamwork.

d) Switching to Solar-Powered Devices

The system's sustainability would be improved and deployment in isolated or off-grid locations would be made possible by switching to solar-powered gear. The system would be less dependent on conventional power sources thanks to solar energy, which would also enable it to be used in places with erratic or nonexistent electrical supplies. This would help underprivileged populations and expand the system's applicability.

REFERENCES

[1] N. Nanthini, A. Shuhaina, K. Shakthi Sharuni, C. Sneka, and S. Sasipriya, "Arduino Powered Smart Weather Monitoring System," in *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*, Institute of Electrical and Electronics Engineers Inc., Aug. 2021, pp. 229–235. doi: 10.1109/ICESC51422.2021.9533016.

[2] M. H. Anik, M. Haque, F. R. Sajid, and M. M. Khan, "Design of IoT based Weather Monitoring System," in *Proceedings - 6th International Conference on Computing Methodologies and Communication, ICCMC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICCMC53470.2022.9753911.

[3] L. Das, A. Kumar, S. Singh, A. R. Ashar, and R. Jangu, "IoT based weather monitoring system using arduino-UNO," in *Proceedings of 2nd International Conference on Computation, Automation and Knowledge Management, ICCAKM 2021*, Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 260–264. doi: 10.1109/ICCAKM50778.2021.9357748.

[4] S. Chakraborty, T. Joshi, and S. Agarwal, "Internet of Things (IoT) Based Cost Effective Weather Monitoring Station," in *Proceedings - IEEE International Conference on Device Intelligence, Computing and Communication Technologies, DICCT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 7–11. doi: 10.1109/DICCT56244.2023.10110037.

[5] R. L. Khan, D. Priyanshu, D. Ather, and H. Allataifeh, "An Implementation of Internet of Things-based Live Temperature and Humidity Monitoring System," in *Proceedings of the 2022 11th International Conference on System Modeling and Advancement in Research Trends, SMART 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 277–281. doi: 10.1109/SMART55829.2022.10046723.

[6] M. H. Bin Zohari and M. F. Bin Johari, "Weather monitoring system using blynk application," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 1315–1319, Nov. 2019, doi: 10.35940/ijitee.L3666.119119.

[7] N. Kumar, M. J. Haque, A. Mittal, A. Singh, A. Rayal, and S. Oli, "IoT based Smart Weather Monitoring System," in *2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems, AICERA/ICIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/AICERA/ICIS59538.2023.10420070.

[8] C. Vanmathi, R. Mangayarkarasi, and R. Jaya Subalakshmi, "Real Time Weather Monitoring using Internet of Things," in *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, Institute of Electrical and Electronics Engineers Inc., Feb. 2020. doi: 10.1109/ic-ETITE47903.2020.348.

[9] A. Indu and S. Magesh Kumar, "An Approach for Implementing Innovative Weather Monitoring System with DHT11 Sensor and Arduino Uno Tool based on IoT," in *6th International Conference on I-SMAC (IoT*

in Social, Mobile, Analytics and Cloud), I-SMAC 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 274–278. doi: 10.1109/I-SMAC55078.2022.9987289.

[10] V. Gaikwad, N. Kamtalwar, H. Karadbhajne, M. Karmarkar, H. Kendre, and O. Ketkar, "IoT- based Automatic Weather Station," in *IDCIoT 2023 - International Conference on Intelligent Data Communication Technologies and Internet of Things, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 826–830. doi: 10.1109/IDCIoT56793.2023.10053407.

[11] *Weather Monitoring Equipment Market Size, Potential Analysis: Unlocking Growth Potential and Share Projections for 2024-2031*. (n.d.). Retrieved June 13, 2024, from https://www.linkedin.com/pulse/weather-monitoring-equipment-market-size-potential swpsc

[12] *The Benefits of Weather Sensors on Commercial Buildings: Improving Safety, Efficiency, and Building Automation Systems | Comptus*. (n.d.). Retrieved June 13, 2024, from https://www.comptus.com/the-benefits-of-weather-sensors-on-commercial buildings-improving-safety-efficiency-and-building-automation-systems/

[13] *Arduino Pin Configuration: A Detailed Guide (2021) | Robu.in*. (n.d.). Retrieved June 13, 2024, from https://robu.in/arduino-pin-configuration/

[14] *DHT11 Sensor Definition, Working and Applications*. (n.d.). Retrieved June 13, 2024, from https://www.elprocus.com/a-brief-on-

dht11-sensor/

[15] *BMP180 Sensor Pinout, Configuration, Specifications, Circuit & Datasheet*. (n.d.).

Retrieved June 13, 2024, from

https://components101.com/sensors/bmp180-

atmospheric-pressure-sensor

[16] *What is BMP180 Sensor with Arduino | LK-Tronics*. (n.d.).

Retrieved June 13, 2024, from https://lk-tronics.com/bmp180-

sensor-with-arduino/

[17] *Guide for Rain Sensor FC-37 or YL-83 with Arduino | Random Nerd Tutorials*. (n.d.).

Retrieved June 13, 2024, from

https://randomnerdtutorials.com/guide-for-rain-sensor fc-

37-or-yl-83-with-arduino/

[18] *NodeMCU ESP8266 Specifications, Overview and Setting Up*.

(n.d.). Retrieved June 13, 2024, from https://www.make-

it.ca/nodemcu-details-specifications/

[19] *What is NodeMCU ESP8266: Complete Guide – Robocraze*. (n.d.).

Retrieved June 13, 2024, from

https://robocraze.com/blogs/post/what-is-nodemcu-esp8266

[20] *(PDF) WEATHER MONITORING SYSTEM USING ARDUINO UNO 1*. (n.d.).

Retrieved June 13, 2024, from

https://www.researchgate.net/publication/352282068_WEATHER_MO

NITORING_S YSTEM_USING_ARDUINO_UNO_1

[21] IOT BASED WEATHER MONITORING SYSTEM USING

ARDUINO-UNO. (n.d.). https://ssrn.com/abstract=4113699

[22] Gsm Based Weather Monitoring System Using Arduino. (n.d.). Retrieved June 13, 2024, from https://projectsfactory.in/product/gsm-based-weather-monitoring-system using-arduino/

[23] IoT Based Weather Monitoring System Using Arduino – DIY Electronics Projects. (n.d.). Retrieved June 13, 2024, from https://electronics-project-hub.com/iot-based weather-monitoring-system-using-arduino/

[24] *Weather Monitoring System using Arduino | Arduino Weather Station NRF*. (n.d.). Retrieved June 13, 2024, from https://techatronic.com/weather-monitoring-system using-arduino-weather-station-using-nrf/

APPENDICES

A.Source Code

```
#define BLYNK_PRINT Serial
#define rainThreshold 2000
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <BlynkSimpleEsp32.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <SimpleTimer.h>
#include "DHT.h"
#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp;  // use I2C interface
Adafruit_Sensor *bmp_temp = bmp.getTemperatureSensor();
Adafruit_Sensor *bmp_pressure = bmp.getPressureSensor();

#define DHTPIN 25      // Digital pin connected to the DHT sensor
#define DHTTYPE DHT22  // DHT 22  (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

#define RainSensorPin 32
#define LEDWiFi 2


// Set up the LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Set the LCD address to 0x27 for a 16x2 display

// WiFi credentials
#define ssid "Shal15"
#define pass "Kushalhini15"

// Blynk auth
char auth[] = "pPn02xcSFhVWZbqykz_i3ValSPRm_lCQ";

SimpleTimer timer;
bool notificationEnabled = true;
unsigned long NotificationInterval = 30000;

unsigned long previousToggleMillis = 0;  // Store the last time the display toggled
```

```
unsigned long toggleInterval = 5000;     // Set toggle interval to 5 seconds
(5000ms)
String temperatureStr, humidStr, pressureStr;
String raincondition;
bool ReadSensorFinished;

void sendsensorData() {
  // Read sensor data
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  sensors_event_t pressure_event;
  bmp_pressure->getEvent(&pressure_event);
  int rainValue = analogRead(RainSensorPin);
  raincondition = rainValue < rainThreshold ? "Raining" : "No Rain";
  temperatureStr = String(temperature, 2);
  humidStr = String(humidity, 2);
  pressureStr = String(pressure_event.pressure, 2);
  // Send data to Blynk
  Blynk.virtualWrite(V0, temperatureStr);
  Blynk.virtualWrite(V1, humidStr);
  Blynk.virtualWrite(V2, pressureStr);
  Blynk.virtualWrite(V3, raincondition);
  ReadSensorFinished = true;
}

void lcd_scroll_line(String text, int row) {
  int len = text.length();
  if (len <= 16) {
    // If the text fits on one line, just display it
    lcd.setCursor(0, row);
    lcd.print(text);
  } else {
    // Scroll text if it's longer than 16 characters
    for (int i = 0; i < len - 15; i++) {
      lcd.setCursor(0, row);
      lcd.print(text.substring(i, i + 16));
      delay(300);  // Adjust delay for scrolling speed
    }
  }
}


void setup() {
  Serial.begin(115200);
  Serial.println ("Starting");
  unsigned status = bmp.begin();
  if (!status) {
    Serial.println("Could not find BMP280 sensor");
```

```
    while (1) delay(10);
  }

  bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,
                  Adafruit_BMP280::SAMPLING_X2,
                  Adafruit_BMP280::SAMPLING_X16,
                  Adafruit_BMP280::FILTER_X16,
                  Adafruit_BMP280::STANDBY_MS_500);

  dht.begin();

  // Initialize LCD
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Connecting WiFi");

  // Start WiFi and Blynk
  Blynk.begin(auth, ssid, pass, "vultrmqtt.makerz-mqtt.com", 8080);

  digitalWrite(LEDWiFi, HIGH);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WiFi Connected");
  delay(3000);
  timer.setInterval(1000, sendsensorData);  // Send data every 1 second
}

void loop() {
  Blynk.run();
  timer.run();
  // Check if the toggle interval has passed
  if (ReadSensorFinished) {
    unsigned long currentMillis = millis();
    if (currentMillis - previousToggleMillis >= toggleInterval) {
      previousToggleMillis = currentMillis;

      // Toggle between temperature/humidity and pressure/rain display
      static bool showStaticData = true;

      if (showStaticData) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Temp:");
        lcd.print(temperatureStr);
        lcd.print(" *C");

        lcd.setCursor(0, 1);
```

```
      lcd.print("Hum: ");
      lcd.print(humidStr);
      lcd.print(" %");
    } else {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Pres:");
      lcd.print(pressureStr);
      lcd.print("hPa");

      lcd.setCursor(0, 1);
      lcd.print("Rain: ");
      lcd.print(raincondition);
    }

    showStaticData = !showStaticData;
  }
 }

}
```