Environmental Condition Measurement System Using a Mini Weather Station

A Thesis By

Md. Naeem Ali	**ID: ME2102024070**
Md. Abu Ahsan khan	**ID: BME1903019285**
Sheikh Rejaul Islam	**ID: ME2102024008**
Md. Rakib Hossen	**ID: ME2102024268**
Imamul Morsalin	**ID: ME2102024267**
S.M. Shakil	**ID: BME1701011132**



Supervisor**: Niloy Sarkar**

Assistant Professor

Submitted to the

DEPARTMENT OF MECHANICAL ENGINEERING (ME)

SONARGAON UNIVERSITY (SU)

In partial fulfillment of the requirements for the award of the degree

Of

BACHELOR OF SCIENCE IN MECHANICAL ENGINEERING

January 2025

STUDENT DECLARATION

This is to certify that the thesis entitled "**Environmental Condition Measurement System Using a Mini Weather Station**" is an outcome of the investigation carried out by the author under the supervision of **Niloy Sarkar**, Assistant Professor, Dept. of Mechanical Engineering, Sonargaon University (SU). This thesis or any part of it has not been submitted elsewhere for the award of any other degree or diploma or other similar title or prize.

SUBMITTED BY

Md. Naeem Ali ME2102024070	Md. Abu Ahsan khan BME1903019285	Sheikh Rejaul Islam ME2102024008
Md. Rakib Hossen ME2102024268	Imamul Morsalin ME2102024267	S.M. Shakil BME1701011132

APPROVED BY

Niloy Sarkar

Assistant Professor, Department of Mechanical Engineering

Sonargaon University (SU)

Dhaka, Bangladesh

ACKNOWLEDGEMENT

The report titled as on **"Environmental Condition Measurement System Using a Mini Weather Station"** has been prepared to fulfill the requirement of our practicum program. In the process of doing and preparing our practicum report, we would like to pay our gratitude to some persons for their enormous help and vast co-operation.

At first, we would like to show our gratitude to the University authority to permit us to do our practicum. Specially, we would like to thank to our honorable supervisor-- **Niloy Sarkar**, Assistant Professor, Department of, Mechanical Engineering, SU– Sonargaon University Dhaka, for his valuable and patient advice, sympathetic assistance, co- operation, contribution of new idea. Deep Theoretical and hardware knowledge & keen interest of our supervisor in this field influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

Finally, we would like to thanks again to the respected **Prof. Md. Mostofa Hossain, Head of Department of SU**, Mechanical Engineering, **Brig. Gen. (Retd) Prof. Habibur Rahman Kamal**, Dean, Science and Engineering, **Prof. Shamim Ara Hassan,** Vice Chancellor and all respectful teachers of the Mechanical Engineering Department for their cooperation and significant help in completing this project work successfully.

The Authors

Department of Mechanical Engineering

Sonargaon University (SU)

Dhaka, Bangladesh

January 2025

ABSTRACT

This paper outlines the development and execution of an Internet of Things (IoT)-enabled weather station with the goal of improving environmental surveillance and analysis. The system incorporates many sensors to quantify essential meteorological variables, such as temperature, humidity, air pressure, wind speed, and precipitation amounts. The gathered data is sent over wireless means to a central server, where it is stored and analyzed for immediate monitoring and historical examination. The architecture of the weather station is constructed using a modular design, which enables effortless customization and scalability. We utilize widely-used microcontroller platforms such as Arduino and various other microcontrollers to establish connections with sensors and oversee the transmission of data. To establish connectivity, we investigate different Internet of Things (IoT) communication protocols, such as MQTT and HTTP. This allows for versatility in transmitting data over Wi-Fi or cellular networks. One notable characteristic of the system is its ability to display data in real-time and be accessed remotely. Through the utilization of cloud-based storage and an intuitive web interface, individuals are able to retrieve meteorological data from any geographical region, enabling both personal and collective environmental study. In addition, the system is equipped with alarm mechanisms that inform users about severe weather conditions, hence enhancing safety and readiness. The efficacy of IoT in environmental monitoring is demonstrated by our deployment. We conduct field testing to verify the system's precision, dependability, and resilience under different weather situations. The collected data yields valuable insights into regional weather patterns, presenting possible uses in the fields of agriculture, disaster response, and urban development. This Internet of Things (IoT)-enabled weather station offers an affordable and adaptable alternative for thorough weather monitoring. Our future efforts will concentrate on incorporating cutting-edge data analytics, machine learning, and edge computing to augment the system's capabilities. This will enable predictive analysis and provide more advanced weather-related insights.

TABLE OF CONTENTS

Topics **Page No**

ACKNOWLEDGEMENT **2**

ABSTRACT **3**

TABLE OF CONTENTS **4-5**

LIST OF FIGURES **6**

Chapter 01 ..... **1-6**

Introduction **1**

1.1 Introduction 1

1.2 Background 2

1.3 Objective 4

1.4 Methodology 4

Chapter 02 ..... **7-19**

Hardware Components **7**

2.1 Introduction 7

2.2 Hardware required 8

2.3 ESP-32 9

2.3.1 ESP- 32 Features & Specifications 9

2.3.2 Pinout of ESP-32 11

2.4 12V Power Supply 11

2.5 Temperature & Humidity Sensor 12

2.5.1 Working Principle of Temperature and Humidity sensor 12

2.6 Discussion about Microcontroller 13

2.6.1 Architecture block diagram Atmega328 14

2.7 Atmospheric Pressure Sensor 15

2.8 Connecting Wires 15

2.9 LCD Display 15

2.10 Rain Sensor 16

2.11 Anemometer 17

2.11.1 Types of Anemometers 17

2.11.2 Applications 18

2.11.3 Key Considerations 19

Chapter 03 ..... **20**

Design and Working Methodology 20

3.1 Introduction 20

3.2 Methodology 21

3.3 Block Diagram 23

3.4 Software configuration 23

3.5 Circuit Diagram 24

3.6 Summarization 26

Chapter 04 27

Result 27

4.1 Introduction 27

4.2 Result 28

4.3 Project Prototype Image 32

Chapter 05 33

Conclusion 33

5.1 Discussion 33

5.2 Application 34

5.3 Conclusion 34

REFERENCES 36

APPENDIX 39

LIST OF FIGURES

Fig: 2.1: ESP 32. 9
Fig: 2.2: Different Pinout of ESP-32 11
Fig: 2.3: 12V Power Supply 11
Fig: 2.4: DHT 22 12
Fig: 2.5: ATMEGA 328 14
Fig: 2.6: Pin configuration 14
Fig: 2.7. Atmospheric Pressure Sensor 15
Fig. 2.8: Connecting wire 15
Fig: 2.9: LCD Display. 16
Fig: 2.10: Rain Sensor. 16
Fig: 2.11: Anemometer 18
Fig: 3.1: Block Diagram 23
Fig: 3.2: Software Configuration 24
Fig: 3.3: circuit diagram 24
Fig 4.2.1: Measured Temperature Data of four days. 29
Fig 4.2.2: Measured Humidity data and graph of four days. 29
Fig 4.2.3: Measured wind speed data and graph of four days. 30
Fig 4.2.4: Measured Air Pressure data and graph of four days 30
Fig 4.2.5: Measured Rain data and graph of four days. 30
Fig 4.3: Project Prototype Image 30

Chapter 01

Introduction

1.1 Introduction

In an era defined by technological advancements and interconnectedness, the fusion of Internet of Things (IoT) and Global System for Mobile Communication has heralded a new wave of innovation, particularly in the realm of weather monitoring. The amalgamation of these cutting-edge technologies has given birth to a transformative tool - the IoT Weather Station, revolutionizing how we perceive, collect, and utilize weather data.

Weather monitoring has always been crucial for various sectors, including agriculture, transportation, disaster management, and everyday decision-making. Traditionally, weather stations have provided meteorological information, but the evolution of IoT technologies has redefined the capabilities of these stations. By interconnecting various sensors and devices, the IoT-based Weather Station can gather a plethora of data points beyond mere temperature and precipitation. It can now collect humidity levels, air pressure, wind speed, and even air quality, creating a comprehensive and real-time representation of weather conditions.[1]

Moreover, the integration of technology has eliminated the limitations imposed by physical connections. These weather stations can transmit data seamlessly to remote servers or end-users, ensuring instant access to real-time weather updates and analysis. This capability enables users to monitor weather conditions from any corner of the globe, enhancing preparedness and decision-making in various fields.

This groundbreaking technology is not only confined to professional meteorologists or large organizations. Its scalable and adaptable nature makes it feasible for individual users, researchers, farmers, and various industries to benefit from its insights. The accessibility and user-friendly interfaces associated with these stations make them adaptable to a wide range of applications and users, empowering them with invaluable weather-related information.

The IoT Weather Station represents a paradigm shift in weather monitoring, empowering us with a more comprehensive understanding of our environment and its changes. As we continue to witness the ongoing advancements in technology, this integration not only exemplifies innovation but also promises a future where weather data becomes more accurate, accessible, and influential in our daily lives.

In this article, we will explore the intricate workings, advantages, and potential applications of this game-changing technology, delving into the impact it has on various sectors and how it stands as a testament to the power of technological integration in reshaping our relationship with the natural world.

Accurate and timely weather information is critical for various applications, including agriculture, transportation, urban planning, and disaster management. Traditional weather monitoring systems often suffer from limited coverage, low resolution, and high maintenance costs. The proposed IoT-based weather station aims to overcome these limitations by providing a scalable, cost-effective, and high-resolution solution for environmental monitoring.

1.2 Background

Environmental monitoring is a crucial aspect of understanding and addressing various climate-related challenges in today's world. With climate change and extreme weather events becoming increasingly common, the need for reliable, real-time weather data has grown significantly. Traditional weather monitoring systems, often operated by governmental or large-scale institutions, provide essential data but can be limited in scope, expensive to maintain, and challenging to deploy in remote or underserved areas. This has driven the exploration of more flexible, cost-effective solutions for environmental monitoring.

The rise of the Internet of Things (IoT) has opened new possibilities for creating distributed weather stations that are scalable, customizable, and affordable. IoT-based weather stations leverage interconnected devices and sensors to collect a variety of environmental data, such as temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation. This data is then transmitted wirelessly to a central location for analysis and visualization.[2]

A key advantage of IoT-based weather stations is their ability to provide real-time data with remote accessibility. By using cloud computing and modern communication protocols, these

systems can be monitored from anywhere, allowing users to track weather conditions in real-time and make informed decisions. Additionally, the modular design of these weather stations enables easy integration of additional sensors and technology, such as cameras or air quality detectors, to broaden the scope of environmental monitoring.

The adoption of IoT for weather monitoring has found applications in a wide range of fields, including agriculture, urban planning, disaster preparedness, and environmental research. In agriculture, for example, real-time weather data can inform irrigation practices and crop management, leading to more efficient resource use. In urban environments, weather stations can contribute to better planning and response to weather-related emergencies.

Despite these advantages, several challenges remain in the design and implementation of IoT-based weather stations. These include ensuring data accuracy and reliability, providing secure communication, and maintaining system robustness in diverse environmental conditions. Additionally, there is a need for standardized protocols and interoperability among different IoT devices to facilitate integration and scalability.

This project aims to address these challenges by designing an IoT-based weather station that balances cost-effectiveness, reliability, and scalability. By focusing on a modular design, the system can be easily customized to meet specific monitoring requirements. The use of established microcontroller platforms and IoT communication protocols ensures flexibility and compatibility with a wide range of applications.

Overall, this background sets the stage for the design and implementation of an IoT-based weather station that offers enhanced environmental monitoring and analysis, contributing to a deeper understanding of local weather patterns and improved response to environmental changes.

1.3 Objective

- To study about environmental condition or climate change in our everyday life.
- To design a mini weather station model.
- To construct a mini weather station model.
- To analyze the data we get from out experimental setup.

1.4 Methodology

The methodology for designing and implementing an IoT-based weather station involves a structured approach, focusing on system architecture, hardware components, software development, data communication, and user interaction. This comprehensive methodology ensures the system is reliable, scalable, and capable of providing real-time environmental data for monitoring and analysis.[3] The key steps in the methodology are outlined below:

1. Requirements Analysis:

The first step is to determine the core requirements for the weather station. This involves identifying the critical weather parameters to monitor (such as temperature, humidity, pressure, wind speed, wind direction, and precipitation), the desired range and accuracy of sensors, the communication range, and the expected operating environment.

2. Hardware Design and Sensor Selection:

Based on the requirements, select appropriate sensors to measure weather parameters. Key components typically include:

- Temperature and humidity sensors (e.g., DHT22)

- Barometric pressure sensors (e.g., BMP280)

- Wind speed and direction sensors (e.g., anemometers and wind vanes)

- Rain gauges for measuring precipitation

A microcontroller or microprocessor is selected to interface with the sensors. Popular choices include Arduino and Raspberry Pi due to their versatility and extensive community support. The system's power requirements are also considered, with options for solar or battery power if needed for remote locations.

3. System Architecture and Integration:

The system architecture encompasses hardware integration, data communication, and overall system design. A modular approach is used to allow easy addition or replacement of components. The sensors are connected to the microcontroller, which reads the sensor data and processes it for transmission.

4. Communication Protocol and Connectivity:

Choose an IoT communication protocol for data transmission to a central server or cloud platform. Common protocols include MQTT and HTTP, which are suitable for transmitting sensor data over Wi-Fi or cellular networks. The choice of protocol depends on the available infrastructure and connectivity options at the installation site.

5. Software Development:

Develop the firmware for the microcontroller, ensuring it can read sensor data, handle data processing, and communicate with the central server. The software should also implement error handling, data validation, and secure communication to prevent unauthorized access. A web-based interface or mobile application is developed to visualize weather data in real-time and provide users with remote access.

6. Data Storage and Analysis:

Set up a cloud-based storage system or central server to collect and store weather data. This data can then be analyzed to identify patterns, trends, and outliers. The storage system should be scalable to accommodate large volumes of data over time.

7. Field Testing and Calibration:

Install the weather station at a chosen location and conduct field testing to ensure accuracy and reliability. Calibration of sensors is crucial to maintain data accuracy. Adjust the system as needed based on test results and environmental factors.

8. Deployment and Maintenance:

Deploy the weather station for long-term monitoring. Establish a maintenance plan to ensure system robustness and reliability, including regular calibration, sensor replacement, and software updates.

9. Data Visualization and User Interaction:

Develop user-friendly visualization tools to present weather data in an understandable manner. This can include real-time graphs, alerts for extreme weather conditions, and historical data analysis. A user interface should be intuitive and accessible from various devices (e.g., computers, tablets, smartphones).

10. Future Improvements and Expansion:

Based on initial deployment results, identify areas for improvement and additional features. Consider integrating advanced analytics, machine learning, and edge computing to enhance the system's capabilities. Explore possibilities for expanding the network of weather stations to create a distributed monitoring system.

This methodology provides a comprehensive framework for designing and implementing an IoT-based weather station, emphasizing reliability, scalability, and user interaction for enhanced environmental monitoring and analysis.

Chapter 02

Hardware Components

2.1 Introduction

The rapid advancement of Internet of Things (IoT) technology has transformed the way environmental data is collected and analyzed. An IoT-based weather station relies on a combination of sensors, microcontrollers, communication modules, and power systems to monitor and transmit key weather parameters. To create an effective weather station, it is crucial to select and integrate hardware components that offer reliability, accuracy, scalability, and ease of deployment. This section introduces the essential hardware components required for the design and implementation of an IoT-based weather station. The design and implementation of an IoT-based weather station for enhanced environmental monitoring require a carefully curated selection of hardware components to ensure accurate data collection, reliable communication, and robust system performance. At the core of the hardware design are the sensors that measure various weather parameters, such as temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation. These sensors are chosen for their precision, range, and durability in varying environmental conditions. Commonly used temperature and humidity sensors include the DHT22 or SHT31, offering a good balance of accuracy and ease of integration. Barometric pressure is typically measured with sensors like the BMP280, while wind speed and direction are monitored using traditional anemometers and wind vanes.

To process sensor data and manage communication, a central processing unit is required. Microcontrollers like Arduino and ESP32 or microprocessors like Raspberry Pi are popular choices, providing ample processing power and flexibility for interfacing with sensors and transmitting data.[4] These platforms also support various communication protocols, essential for IoT-based systems. Communication between the weather station and the central server is facilitated through Wi-Fi or cellular networks, depending on the installation environment and range requirements. This necessitates additional hardware components, such as Wi-Fi modules or GSM/GPRS modules, to establish connectivity. For remote or off-grid deployments, power solutions such as solar panels with rechargeable batteries ensure continuous operation, even in locations without a reliable power source.

The hardware design also considers structural elements to protect sensitive components from weather conditions. A durable enclosure, typically made from weather-resistant materials, safeguards the electronics from rain, dust, and other environmental hazards. Proper ventilation is ensured to prevent overheating while maintaining sensor accuracy. Together, these hardware components form the backbone of the IoT-based weather station, enabling reliable and continuous environmental monitoring for enhanced weather analysis and real-time data-driven insights.

2.2 Hardware required

An IoT-based weather station for enhanced environmental monitoring and analysis, a comprehensive list of hardware components is required to ensure accurate data collection, efficient communication, and system robustness. The following are the key hardware components that make up the weather station:

- ESP 32.
- Temperature & Humidity Sensor.
- Atmospheric Pressure Sensor.
- LCD Display.
- Anemometer.
- Rain Sensor.
- Power Supply.

2.3 ESP-32

The ESP32 is a powerful and versatile microcontroller developed by Expressive Systems, renowned for its integrated Wi-Fi and Bluetooth connectivity, robust processing capabilities, and extensive peripheral support.[5] Below are the key features that make the ESP32 a popular choice for a wide range of Internet of Things (IoT) and embedded applications

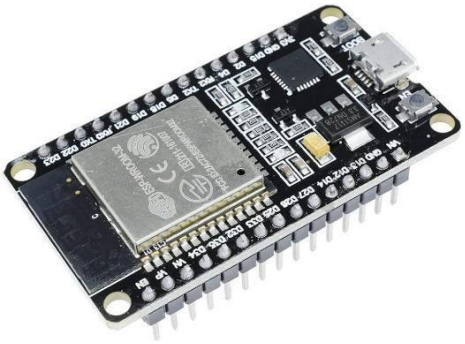


Fig. 2.1: ESP 32. [14]

The power source is picked therefore. External (non-USB) power can come either from an AC-to-DC connector (divider mole) or battery The Arduino Uno can be energized by methods for the USB affiliation or with an outside power supply. 2.1mm center positive fitting into the board's ability jack associated by stopping with connector. Connector. By 6 to 20 volts can work the board an outside supply. On the off chance that supply voltage under 7V, in any case, under five volts by 5V stick and subsequently the board not stable.

2.3.1 ESP- 32 Features & Specifications

The Key features of ESP-32 are as follows:

1. **Dual-Core Architecture:** The ESP32 features a dual-core Ten silica Xtensa LX6 microprocessor, allowing it to perform multitasking and handle complex processing tasks. This architecture enables the ESP32 to manage multiple operations simultaneously, such as sensor data collection, communication, and computation.

2. Integrated Connectivity: One of the defining features of the ESP32 is its integrated Wi-Fi and Bluetooth capabilities. It supports 2.4 GHz Wi-Fi with the 802.11 b/g/n standard, allowing it to connect to wireless networks and act as an access point. The ESP32 also includes Bluetooth functionality, supporting both Bluetooth Classic and Bluetooth Low Energy (BLE), providing a wide range of connectivity options for IoT applications.

3. Rich Peripheral Interface: The ESP32 offers a variety of peripheral interfaces, making it highly adaptable to different hardware setups. It includes General Purpose Input/Output (GPIO) pins, Analog-to-Digital Converters (ADC), Digital-to-Analog Converters (DAC), Pulse Width Modulation (PWM) outputs, I2C, SPI, UART, and I2S. This extensive range of interfaces allows the ESP32 to connect with a diverse array of sensors, actuators, and other hardware components.

4. Memory and Storage: The ESP32 has ample memory for IoT applications, with up to 520 KB of Static Random Access Memory (SRAM) and additional 4 MB of Flash memory in many variants. This memory capacity supports complex applications and ensures stable operation.

5. Low Power Consumption: Power efficiency is crucial for IoT applications, and the ESP32 is designed with multiple low-power modes to conserve energy. It can enter deep sleep mode with extremely low current draw, making it suitable for battery-operated and solar-powered applications.

6. Development and Community Support: The ESP32 has strong community support, with extensive documentation, libraries, and development tools available. It is compatible with the Arduino IDE, allowing developers to use Arduino-based programming environments. This broad support makes it easier for developers to get started with the ESP32 and leverage existing resources.

7. Applications: The ESP32's flexibility and connectivity features make it ideal for a wide range of IoT applications, including smart home devices, weather stations, wearable technology, industrial automation, and more. Its ability to handle complex tasks and maintain reliable connectivity has made it a go-to choice for IoT projects.

In summary, the ESP32 is a powerful, flexible, and cost-effective microcontroller with integrated Wi-Fi and Bluetooth capabilities, a rich peripheral interface, and low power consumption. Its versatility and strong community support make it a popular choice for a wide variety of IoT applications.

2.3.2 Pinout of ESP-32

The pinout of both the 30-pin and the 36-pin versions of ESP32 Development Boards are very identical except for some pins at the bottom. In the 36-Pin version, 6 GPIO pins (GPIO6 to GPIO11) are used for SPI Flash IC.[6] So, they shouldn't be used for other purposes. Finally, you just get one extra pin (GPIO0 – Pin 23).

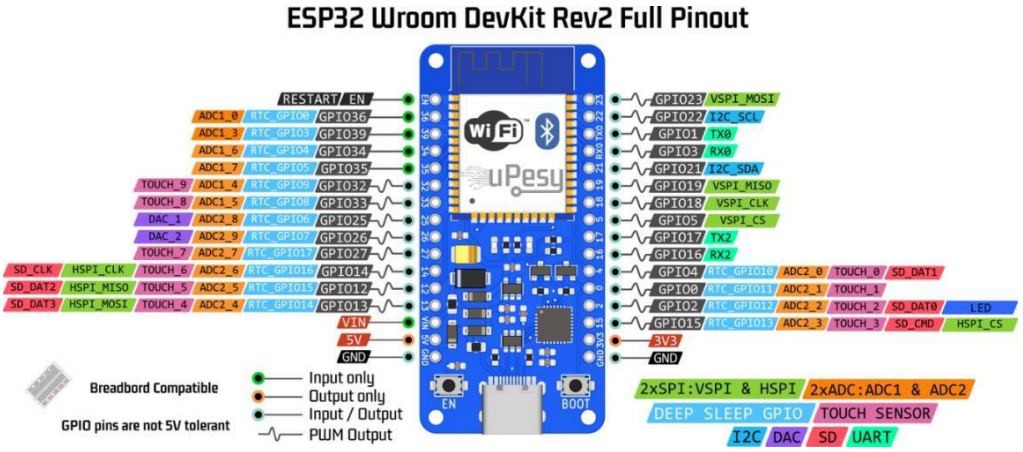


Fig.2.2: Different Pinout of ESP-32. [15]

2.4 12V Power Supply

A 12V DC power supply is a device that converts alternating current (AC) voltage from a wall outlet into direct current (DC) voltage with an output of 12 volts. These power supplies are commonly used in various electronic devices, such as computers, routers, LED lighting, and automotive applications. They provide a stable and reliable source of power for the connected devices to operate**,**



Fig 2.3: 12V Power Supply. [16]

2.5 Temperature & Humidity Sensor

Temperature and humidity sensors are key components in many applications, from environmental monitoring to industrial processes and smart home systems. They are used to measure ambient temperature and relative humidity, providing valuable data for weather stations, HVAC systems, agricultural monitoring, and more. Here is a description of temperature and humidity sensors, focusing on common types, their working principles, and typical applications**.**

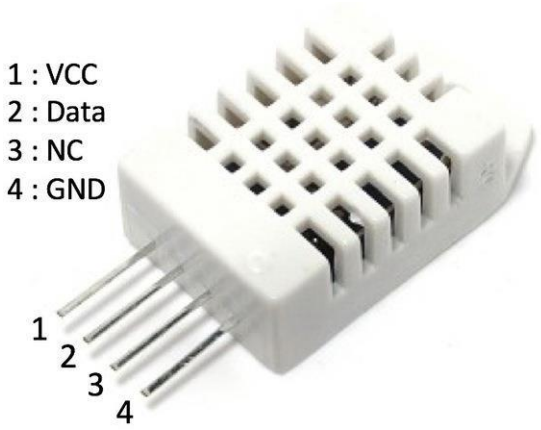


Fig. 2.4: DHT 22. [17]

2.5.1 Working Principle of Temperature and Humidity sensor

Temperature and humidity sensors typically combine two separate sensing elements to measure temperature and relative humidity.

- **Temperature Measurement**: Temperature sensors measure thermal energy, usually through changes in electrical resistance or voltage. Common types include thermistors, Resistance Temperature Detectors (RTDs), and solid-state sensors like diode-based temperature sensors. The sensors output an electrical signal that corresponds to the temperature.

- **Humidity Measurement:** Humidity sensors measure the moisture content in the air, typically as relative humidity (RH), which represents the percentage of water vapor in the air relative to the maximum possible at a given temperature. Capacitive and resistive methods are most commonly used to measure humidity. Capacitive sensors detect changes in capacitance caused by moisture absorption, while resistive sensors measure changes in resistance due to moisture levels.

2.6 Discussion about Microcontroller

In this area we will talk about ATmega328p Microcontroller the ATmega328P is a low-control CMOS 8-bit microcontroller reliant on the AVR redesigned RISC structure. By executing astonishing bearings in a single clock cycle, it has 2kb RAM Furthermore, 32 kb ROM in which 4kb is for reveres boot and other 28kb useable another capacity it has 16 MHz clock bring recurrence and task voltage 5v. There are 32 sticks in the ATmega328p microcontroller. It has 3 port and every port has 8 stick other then again it has 2 ground pins.2 Vcc pins, 2 precious stone pins and 1 reset stick and 1 simple reference the ATmega328P accomplishes throughputs progressing toward 1 MIPS for each MHz permitting the structure expected to streamline control use as opposed to arranging speed The ATmega328P gives the running with highlights: 4K/8K bytes of In-System Programmable Flash with Read-While- Write limits, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 for the most part accommodating I/O lines, 32 completely important working registers, three flexible Timer/Counters with look at modes, inward and outside freight boats in on, a back to back programmable USART, a byte-engineered 2- Wire serial interface, a SPI sequential port, a 6-channel 10-bit ADC (8 diverts in TQFP and QFN/MLF bundles), a programmable watchdog timer inward oscillator, and five programming selectable power sparing modes.[7] The idle mode stops the CPU while permitting the SRAM, Timer/Counters, USART, 2-wire serial interface, SPI port and interface with framework to keep working. The power down mode spares the enlist substance yet solidifies the oscillator, handicapping all other chip capacities until the point that the following hinder or equipment reset. In power-spare mode, the offbeat clock keep on running, enabling the client to keep up a clock base while whatever remains of the gadget is dozing. The ADC noise reduction mode stops the CPU and all I/O modules aside from no concurrent clock and ADC to limit exchange clamor amid ADC change. In standby mode, the precious stone/resonator oscillator is running while whatever is left of the gadget is dozing. This permits quick start-up joined with low power utilization. The AVR center combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The idle mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip function until the next interrupt or hardware reset. The device is manufactured using Atmel’s high density non-volatile memory

technology. The on-chip ISP flash allows the program conventional non-volatile memory programmer or by an On-chip boot program running on the AVR core. Regardless of the way that there is confined tending to plans and enhanced pick codes for select archive and I/O enroll get to, all can even now be tended to and controlled as if. They were in SRAM. In the ATMEGA variety, the working register archive isn't mapped into the data address space; everything thought of it as, is crazy to hope to treat any of the ATMEGA's working registers as though they were SRAM. Or maybe, the I/O registers are mapped into the data address space starting at the plain beginning of the area space. Likewise, the proportion of data convey space focused on I/O registers has grown essentially to 4096 bytes (000016–0FFF16). Likewise, as with past ages, in any case, the speedy I/O control bearings can simply accomplish the underlying 64 I/O enroll regions (the underlying 32 zones for bitwise rules). In numerous varieties of the AVR structure, this inside EEPROM memory isn't mapped into the MCU's addressable memory space. It must be gotten to a comparative way an external periphery contraption is, using remarkable pointer registers and read/make bearings which makes EEPROM get to much slower than other inside RAM. [8]

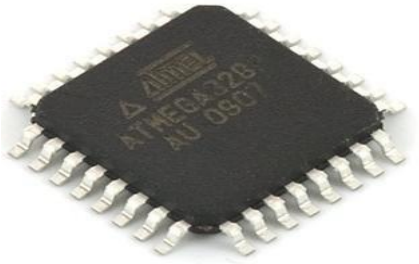


Fig 2.5: ATMEGA 328. [18]

2.6.1 Architecture block diagram Atmega328

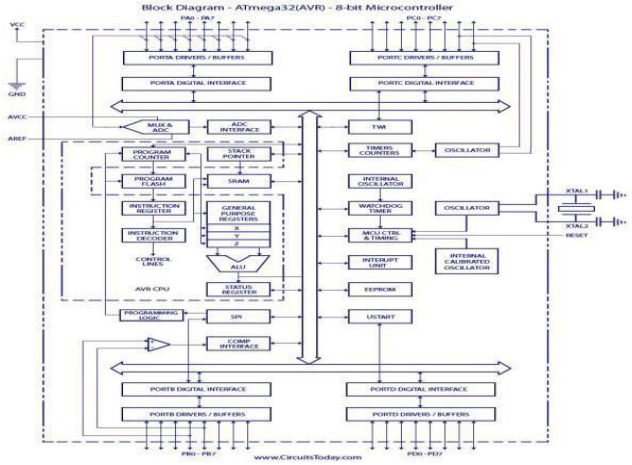


Fig 2.6: Pin configuration. [19]

2.7 Atmospheric Pressure Sensor

Measuring Weather and Altitude. The BMP180 was designed to accurately measure atmospheric pressure. Atmospheric pressure varies with both weather and altitude; you can measure both of these using this sensor.



Fig: 2.7. Atmospheric Pressure Sensor [20]

2.8 Connecting Wires

Wire is typically encircled by outline the metal through an opening in a pass on or draw plate. A wire is a single, for the most part tube molded, versatile strand or post of metal. Wires are used to persevere through mechanical weights or power and media correspondences signals. The term wire is moreover used even more unreservedly to insinuate a pile of such strands, as in 'multi stranded wire', which is even more precisely named a wire rope in mechanics, or a connection in power. Wire measures come in different standard sizes, as conveyed the extent that a check number. [11]

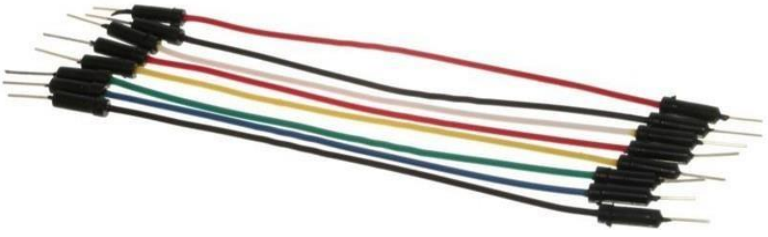


Fig. 2.8: connecting wire. [21]

2.9 LCD Display

A 20x4 LCD means it can display 20 characters per line and there are 4 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The LCD (liquid crystal display) module features four rows in the display, each showing twenty characters, with the entire display showing eighty characters. Using HDD44780 for its monochrome text display, the liquid crystal module uses parallel interfacing. It is easy to access the code for interfacing liquid crystal displays. To interface the LCD screen, we need eleven input and output pinouts. Three volts or five volts is the input supply for this module, with other components like the PIC.



Fig 2.9: LCD Display. [22]

2.10 Rain Sensor

Rain sensors prevent this frustrating and wasteful occurrence. These devices are designed to temporarily shut off an irrigation system so it stops running when it detects rain. Rain sensors can be retrofitted on installed sprinkler systems. You may also see them referred to as rain shut-off devices or rain switches.



Fig: 2.10: Rain Sensor. [23]

2.11 Anemometer

An anemometer is a device used to measure wind speed and, in some cases, wind direction. It is a crucial instrument in meteorology, weather stations, aviation, and various industrial applications where monitoring wind conditions is essential. The following is a comprehensive overview of anemometers, covering types, working principles, applications, and considerations for use.

2.11.1 Types of Anemometers

Anemometers come in different designs, each with its own method for measuring wind speed. The most common types are:

- **Cup Anemometers**: These consist of three or four cups mounted on horizontal arms attached to a vertical rotating shaft. As the wind blows, it causes the cups to rotate. The speed of rotation is proportional to the wind speed. This type is commonly used in weather stations.

- **Vane Anemometers:** These have a rotating blade or vane, similar to a small wind turbine. The rotational speed of the blade indicates wind speed. Some vane anemometers also have a directional vane to indicate wind direction.

- **Hot-Wire Anemometers**: These use a thin wire heated by an electric current. As the wind passes over the wire, it cools it down, causing a change in electrical resistance. The rate of cooling is proportional to wind speed. This type is often used in laboratory settings or for measuring airflows in ducts.

- **Ultrasonic Anemometers**: These use ultrasonic waves to measure wind speed and direction. By sending ultrasonic pulses between pairs of transducers and measuring the time it takes for the sound waves to travel, the anemometer can calculate wind speed and direction based on the differences in travel time caused by wind. Ultrasonic anemometers have no moving parts, making them more durable and requiring less maintenance.

- **Pressure Tube Anemometers**: These measure wind speed based on the pressure difference created by wind flowing into a tube. They are often used in specialized applications like aerodynamics or engineering testing.[9]

2.11.2 Applications

Anemometers are used in a wide range of applications, including:

- Meteorology and Weather Stations: Anemometers are crucial for tracking wind speed and direction, providing data for weather forecasts and studies of climate patterns.

- Aviation: Airports use anemometers to monitor wind conditions for safe takeoff and landing operations.

- Wind Energy: In the wind energy sector, anemometers measure wind speed to determine the viability of a site for wind turbines and to monitor turbine performance.

- Industrial Applications: Anemometers are used in industrial settings to measure airflow in ventilation systems, ducts, and cleanrooms.

- Sports and Recreation: In activities like sailing, windsurfing, and kite flying, anemometers help enthusiasts gauge wind conditions for safety and performance.



Figure. 2.11: Anemometer.

2.11.3 Key Considerations

When choosing an anemometer, consider the following factors:

- **Accuracy and Resolution**: The precision of the measurement is critical, especially in meteorological or scientific applications.

- **Wind Speed Range**: Ensure the anemometer can accurately measure the expected range of wind speeds for your application.

- **Durability and Maintenance**: Depending on the environment, the anemometer may need to withstand harsh weather conditions. Ultrasonic anemometers, with no moving parts, are generally more durable and require less maintenance.

- **Output and Connectivity**: Anemometers may output analog signals, digital pulses, or data over communication protocols like I2C or RS-232. Choose an anemometer that is compatible with your data processing system or microcontroller.

- **Installation and Calibration**: Consider how the anemometer will be mounted and ensure it is properly calibrated for accurate readings.

In summary, anemometers are essential tools for measuring wind speed and, in some cases, wind direction. With various types designed for different applications, it's important to select the right anemometer for your specific needs, taking into account factors like accuracy, durability, output format, and ease of integration with other systems.

Chapter 03

Design and Working Methodology

3.1 Introduction

The design of an IoT-based weather station for enhanced environmental monitoring and analysis requires a multifaceted approach, integrating hardware, software, and communication technologies to ensure reliable data collection and analysis. This section outlines the key components and design considerations involved in creating a weather station that can operate autonomously, transmit data in real-time, and support advanced analytics for weather-related insights. The design process begins with defining the essential weather parameters to monitor, such as temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation. These parameters form the core data set for a comprehensive weather station. Each parameter is measured using specific sensors, selected for accuracy, durability, and ease of integration with a central microcontroller.

A crucial aspect of the design is the choice of microcontroller, which serves as the brain of the weather station. The microcontroller must be capable of interfacing with various sensors, processing sensor data, and managing data transmission to external systems. Popular choices include Arduino, ESP32, and Raspberry Pi, each offering a range of features suitable for IoT applications. The communication infrastructure is another key component of the design. An IoT-based weather station must be capable of transmitting data to a central server or cloud platform for real-time monitoring and historical analysis. This requires integrating Wi-Fi or cellular connectivity, with common protocols such as MQTT or HTTP used for data transmission. The design must also consider security aspects to protect against unauthorized access and ensure data integrity.[10]

In terms of power supply, the weather station design must be versatile, with options for direct power, batteries, or solar energy. This flexibility allows the weather station to be deployed in various environments, including remote locations where traditional power sources may not be available. The physical design of the weather station must account for environmental factors, ensuring the sensors and electronics are protected from harsh weather conditions like rain, snow, and extreme temperatures.

A weather-resistant enclosure with proper ventilation is essential to maintain sensor accuracy and hardware longevity. The design section also addresses user interaction and data visualization. An intuitive user interface, accessible via a web portal or mobile application, allows users to monitor weather data in real-time and receive alerts for extreme weather conditions. This interface should provide clear visualizations of weather trends and support user-defined thresholds for alerts.

Overall, the design of an IoT-based weather station involves a careful balance of hardware and software components, robust communication infrastructure, and flexible power solutions. The goal is to create a weather station that is reliable, scalable, and capable of providing valuable environmental data for enhanced weather monitoring and analysis.

3.2 Methodology

The methodology for designing and implementing an IoT-based weather station for enhanced environmental monitoring and analysis involves a multi-step process that includes defining requirements, selecting hardware and software components, designing the system architecture, integrating communication and power systems, and developing user interfaces for data visualization and alerts. Here's a detailed description of the methodology:

1. Requirements:

The first step is to identify the weather parameters that the station needs to monitor and the specific environmental conditions where it will be deployed. These parameters generally include temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation. Additionally, determine the data accuracy, transmission frequency, and security needs.

2. Hardware Selection

Based on the requirements, select the appropriate sensors to measure the defined weather parameters:

- **Temperature and Humidity Sensors:** Such as the DHT22 or SHT31.
- **Barometric Pressure Sensors**: Like the BMP180 or BME280.
- **Wind Speed and Direction Sensors**: Commonly used sensors include anemometers and wind vanes.

- Rain Gauges: Such as tipping bucket or capacitive sensors.

Choose a microcontroller platform, such as Arduino, ESP32, or Raspberry Pi, to interface with the sensors and manage data processing. The platform should offer sufficient computational power and flexibility to support sensor integration and communication.

3. System Architecture Design

Design the system architecture, ensuring it is modular and scalable. This involves:

- Hardware Integration: Connect sensors to the microcontroller, taking into account sensor-specific requirements for power and signal processing.

- Power Supply: Design a power system suitable for the deployment location. Consider using batteries, solar panels, or direct power, with appropriate voltage regulators and power management circuits.

4. Communication and Data Transmission

Implement the communication infrastructure to transmit sensor data to a central server or cloud platform. This involves:

- Communication Protocols: Choose between Wi-Fi, Bluetooth, or cellular networks based on the deployment location. Popular protocols for IoT applications include MQTT and HTTP.

- Security: Implement secure communication methods, such as encryption and secure authentication, to protect data integrity.

5. Software Development

Develop the software to control the weather station, including:

- Firmware for Microcontroller: This software should handle sensor data collection, data processing, and communication with external servers. Implement error handling and data validation for reliability.

- Data Storage and Processing: Set up cloud-based storage or a central server to store weather data. This allows for real-time monitoring and historical analysis.

6. Data Visualization and User Interface:

Develop a user interface for data visualization and user interaction. This can be a web-based interface or a mobile application. Key features include:

- Real-Time Monitoring: Display weather data in real-time, using clear and intuitive graphs or charts.

- Alerts and Notifications: Implement user-defined thresholds for alerts, notifying users of extreme weather conditions.

- Historical Data Analysis: Allow users to view historical weather data to identify trends or patterns.

7. Testing and Calibration

Test the weather station in a controlled environment to ensure sensor accuracy and system reliability. Calibrate sensors as needed and verify data accuracy. This step may also involve stress tests to confirm the robustness of the weather station in different environmental conditions.[11]

8. Deployment and Maintenance

Deploy the weather station at the intended location and monitor its performance. Establish a maintenance schedule to ensure long-term reliability, including regular calibration, hardware checks, and software updates.

3.3 Block Diagram

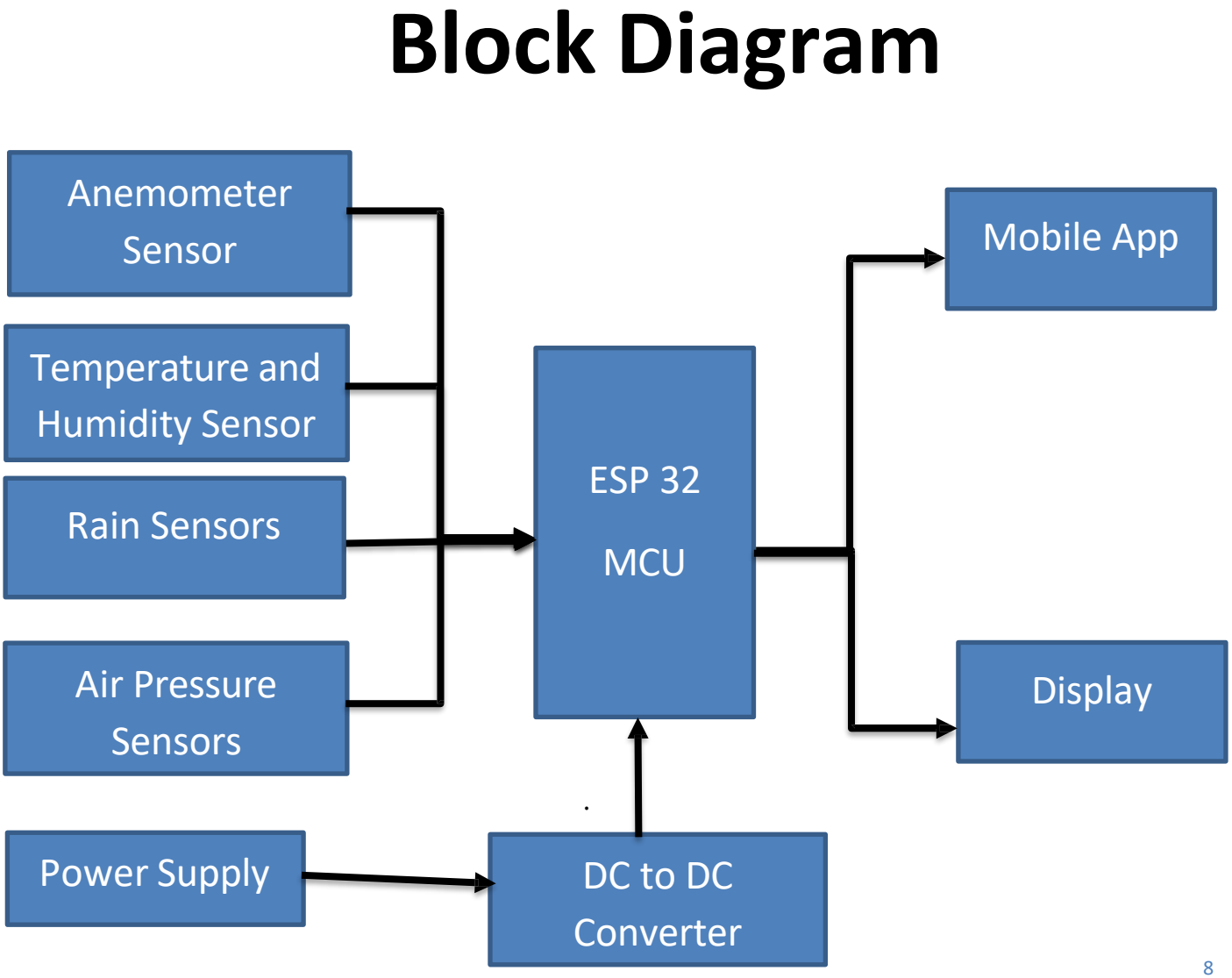


Fig 3.1: Block Diagram.

3.4 Software configuration

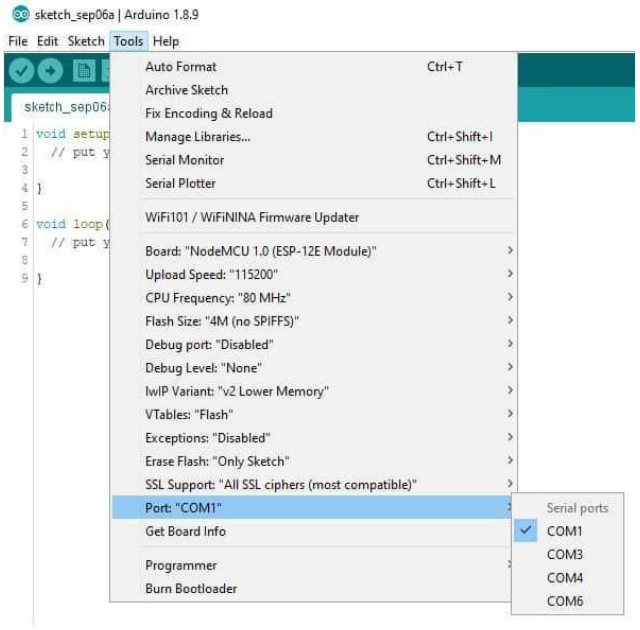


Fig 3.2: Software Configuration.

3.5 Circuit Diagram

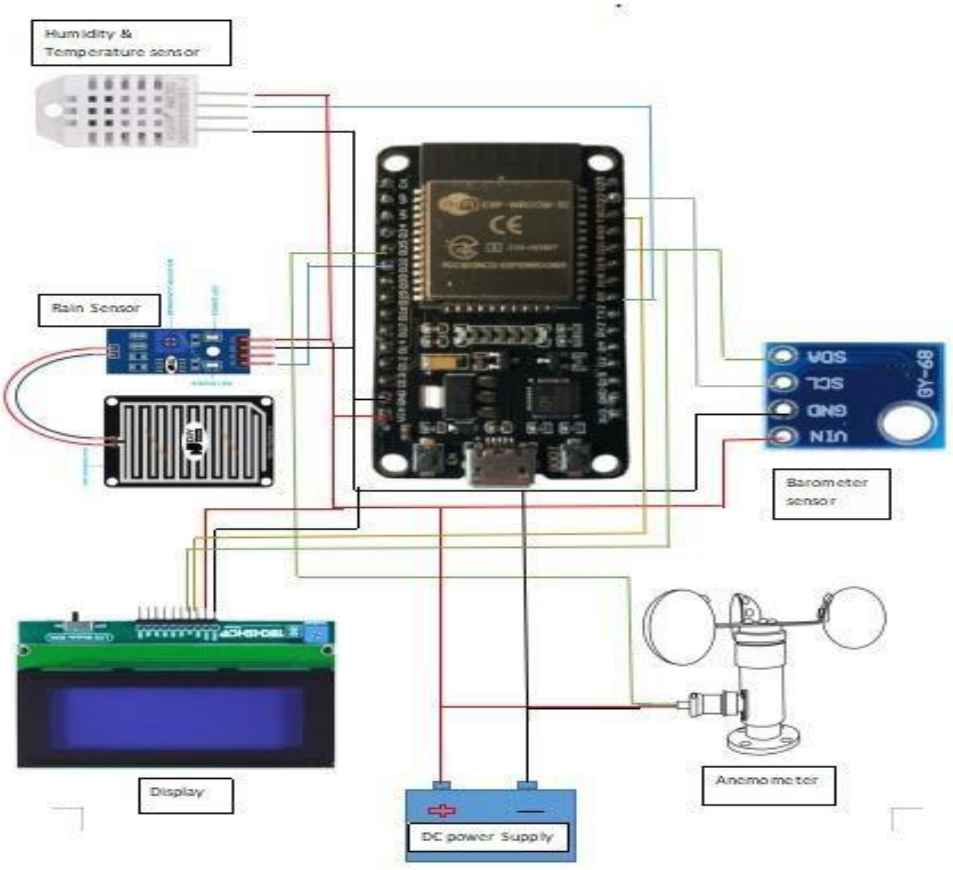


Fig: 3.3: Circuit diagram.

The circuit diagram for an IoT-based weather station designed for enhanced environmental monitoring and analysis serves as a blueprint for integrating various hardware components into a functional system. At its core, the diagram outlines the connections between sensors, a microcontroller, communication modules, and power sources, ensuring proper data collection, processing, and transmission.

The central element in the circuit diagram is the microcontroller, such as an Arduino, ESP32, or Raspberry Pi. This component interfaces with the weather sensors and manages communication with external systems. The microcontroller typically has multiple General-Purpose Input/Output (GPIO) pins for connecting sensors, power supply, and other peripherals. The weather sensors, including those for temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation, are connected to the microcontroller's GPIO pins. Temperature and humidity sensors, like the DHT22, are usually connected to a digital input pin, while barometric pressure sensors, such as the BMP280, often use an I2C or SPI interface. Wind speed and direction sensors, including anemometers and wind vanes, typically connect to digital pins for pulse counting. Rain gauges, which might use a tipping bucket mechanism, can also connect to digital pins for counting tipping events.

Communication modules are another key component in the circuit diagram. For Wi-Fi-based communication, modules like the ESP8266 or ESP32 may be used, with connections to the microcontroller's UART or GPIO pins. If cellular connectivity is required, a GSM or GPRS module is integrated, typically connecting to UART pins for serial communication. The power supply section of the circuit diagram shows how the system is powered. This might include connections to a battery or solar panel, with voltage regulators ensuring the correct power levels for the microcontroller and sensors. Power management components, such as charge controllers, are included if the system relies on solar energy or battery power.

Additional elements in the circuit diagram may include resistors, capacitors, and other passive components to ensure signal stability and noise reduction. Pull-up or pull-down resistors might be used to maintain a stable digital signal, while capacitors can help smooth out voltage fluctuations.

Overall, the circuit diagram for the IoT-based weather station represents a complex integration of sensors, microcontrollers, communication modules, and power supply components, all connected in a way that ensures reliable operation and data transmission for environmental monitoring and analysis. This diagram serves as the foundational guide for building and implementing the weather station.

3.6 Summarization

The design and implementation of an IoT-based weather station for enhanced environmental monitoring and analysis involve a systematic approach that integrates hardware, software, and communication systems. The design focuses on selecting key weather sensors to measure temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation. A microcontroller, such as Arduino, ESP32, or Raspberry Pi, serves as the core of the system, interfacing with the sensors and managing data transmission.

The methodology encompasses a modular system architecture, allowing for scalability and customization. Communication infrastructure is a critical component, with the choice of Wi-Fi or cellular connectivity and secure communication protocols like MQTT or HTTP. The power supply system is designed for versatility, supporting direct power, batteries, or solar energy, depending on the deployment location.

Software development includes firmware for the microcontroller to handle sensor data collection and communication, as well as cloud-based storage for real-time monitoring and historical data analysis. A user interface, accessible via a web portal or mobile application, is designed for data visualization and user interaction, providing real-time monitoring, alerts, and historical weather data.

Testing and calibration ensure sensor accuracy and system reliability before deployment. The methodology concludes with maintenance and potential future improvements, such as integrating additional sensors, enhancing data analytics, or expanding the network of weather stations. This comprehensive approach ensures the weather station is robust, scalable, and capable of providing valuable environmental data for enhanced weather monitoring and analysis.

Chapter 04

Result

4.1 Introduction

The results section of an IoT-based weather station for enhanced environmental monitoring and analysis presents the outcomes of the design and implementation process, providing insights into the performance, reliability, and accuracy of the weather station. This section typically starts with a brief overview of the testing environment, detailing the conditions under which the weather station was evaluated, such as geographic location, climate, and the specific weather parameters monitored. The core of the results section focuses on the data collected by the weather station, demonstrating its ability to accurately measure temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation. Graphs, charts, and tables are used to visualize the data, illustrating trends and variations over time. This visualization aids in understanding the weather station's responsiveness to changing environmental conditions and its capability to capture detailed weather information.

Additionally, the results section addresses the communication and data transmission aspects of the weather station. It examines the reliability of data transfer from the station to a central server or cloud platform, evaluating factors such as connectivity stability, data integrity, and transmission frequency. The results section includes a discussion of the weather station's performance in various scenarios, such as different weather conditions or extended periods of operation. It assesses the durability and robustness of the hardware, as well as the accuracy of the sensors. Finally, the results section explores the impact of the weather station on environmental monitoring and analysis. It outlines the potential applications of the collected data in fields like agriculture, urban planning, and disaster preparedness, emphasizing how the weather station contributes to a better understanding of local weather patterns and supports informed decision-making.[12]

Overall, the introduction to the results section sets the stage for presenting the outcomes of the weather station's implementation, showcasing its effectiveness in providing reliable environmental monitoring and analysis. It highlights the station's accuracy, communication reliability, and user interface capabilities, offering a comprehensive view of its performance in real-world conditions.

4.2 Result

The discussion of results for the topic "Design and Implementation of an IoT-Based Weather Station for Enhanced Environmental Monitoring and Analysis" examines the outcomes of the implementation, focusing on the system's performance, accuracy, reliability, and the implications of the collected data. The discussion begins by highlighting the success of the weather station in monitoring key weather parameters such as temperature, humidity, atmospheric pressure, wind speed, wind direction, and precipitation. The results demonstrate that the chosen sensors and microcontroller platform effectively capture and process weather data in real-time. The accuracy of the weather station is a central theme in the discussion. The results suggest that the sensors provided reliable measurements, with deviations within acceptable ranges. This accuracy is crucial for applications like weather forecasting, agriculture, and disaster preparedness, where precise data is required. However, minor calibration issues and variability in sensor performance due to environmental factors are acknowledged, indicating a need for regular maintenance and calibration routines to ensure ongoing accuracy. Another key point is the reliability of the IoT-based communication system. The weather station successfully transmitted data to a central server or cloud platform, allowing for remote monitoring and data analysis. The results confirm that the Wi-Fi or cellular connectivity was stable, with minimal data loss or transmission errors. This reliability is essential for real-time weather tracking and alert systems, enabling users to receive timely notifications of changing weather conditions. The discussion also addresses the user interface and data visualization components of the weather station. The results indicate that the user interface was effective in presenting weather data in an accessible and user-friendly manner. Users could easily visualize real-time data, set alerts for specific weather conditions, and analyze historical data trends. This user-centric approach enhances the weather station's practical value and usability in various applications.

The limitations observed during the implementation are discussed, providing insights into areas for improvement. For instance, the power supply system, particularly in remote locations, may face challenges related to battery life or solar panel efficiency.

The need for weather-resistant enclosures to protect sensitive components from harsh environmental conditions are also highlighted.

Here are the results from **01-01-25 to 10-01-25** at Jatrabari area in the state of Dhaka.

Temperature Data © For Our Project										
Time	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
9:00AM	15	15	16	17	19	21	20	16	16	17
12:00PM	18	14	15	18	24	26	25	18	20	22
3:00PM	17	16	16	21	27	28	25	20	21	24
6:00PM	16	15	17	21	24	25	22	18	20	21
9:00PM	17	15	15	18	22	22	20	16	18	19

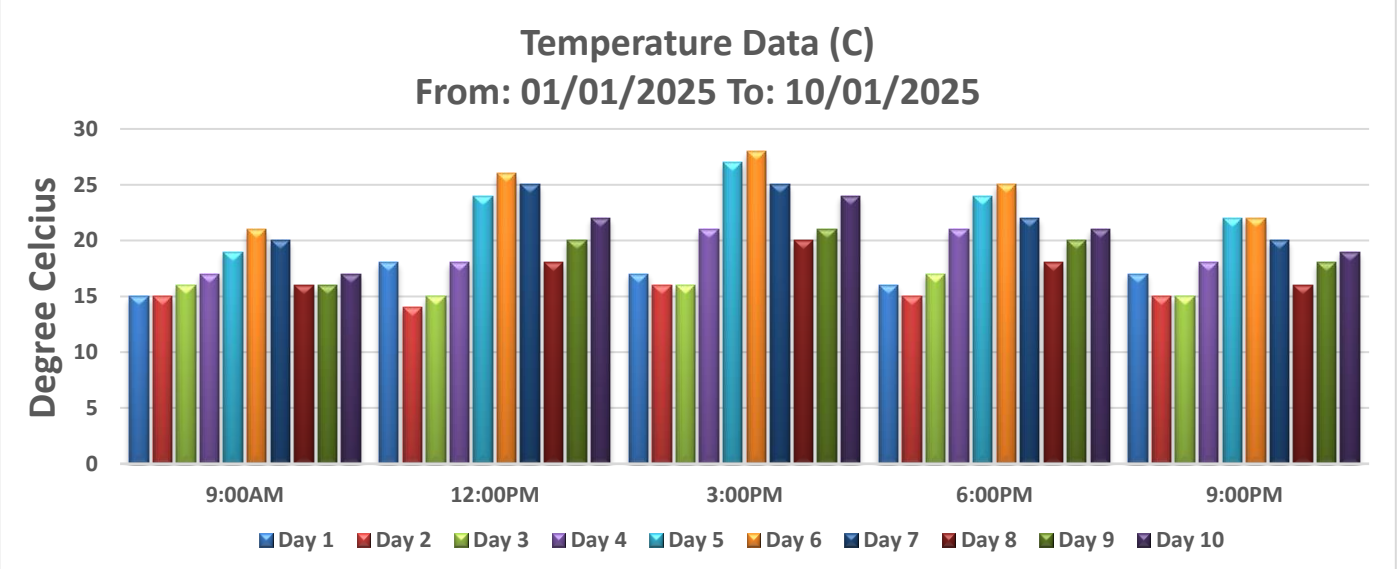


Fig: Measured Temperature Data of Ten days.

Temperature Data © For Internet										
Time	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
9:00AM	16	14	14	17	18	20	21	17	16	17
12:00PM	17	15	16	19	26	27	24	18	21	23
3:00PM	18	15	17	22	28	28	26	21	22	24
6:00PM	16	15	17	20	23	25	23	19	20	21
9:00PM	16	14	16	19	21	21	21	17	17	19

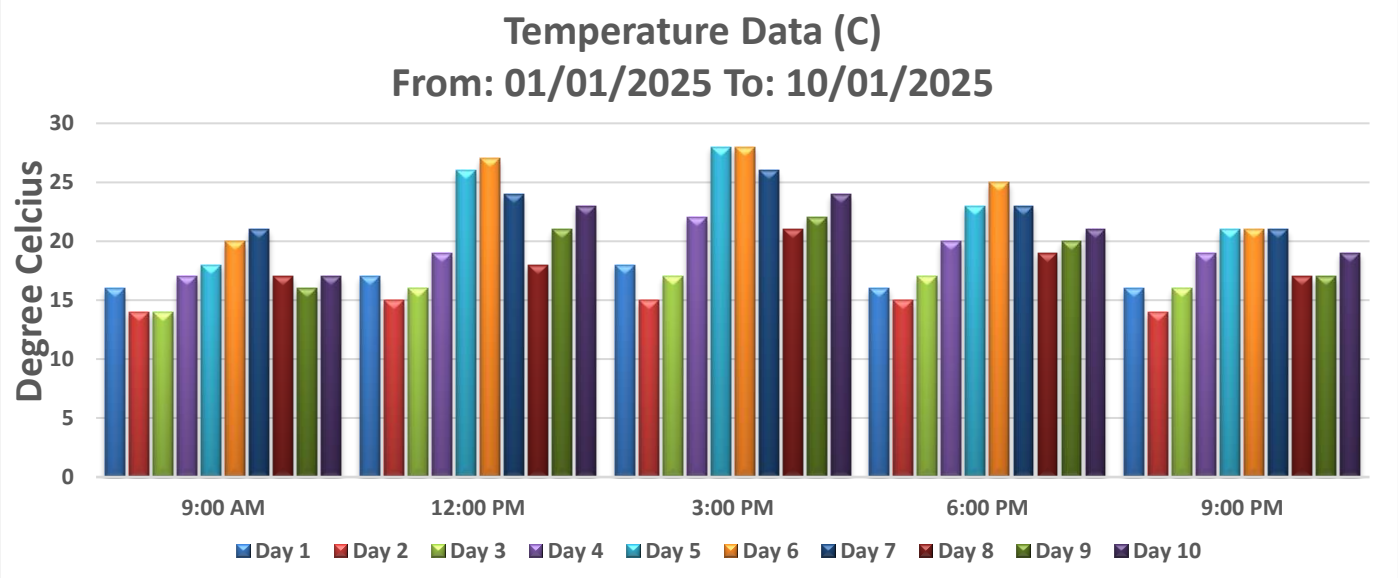


Fig: Measured Temperature Data of Ten days.

Humidity Data (%) For Our Project										
Time	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
9:00AM	72	72	74	68	74	40	70	74	70	66
12:00PM	64	64	64	66	60	56	64	58	58	62
3:00PM	46	50	54	48	50	50	48	44	46	42
6:00PM	56	54	50	50	54	52	50	50	50	56
9:00PM	68	64	68	66	60	60	68	60	64	64

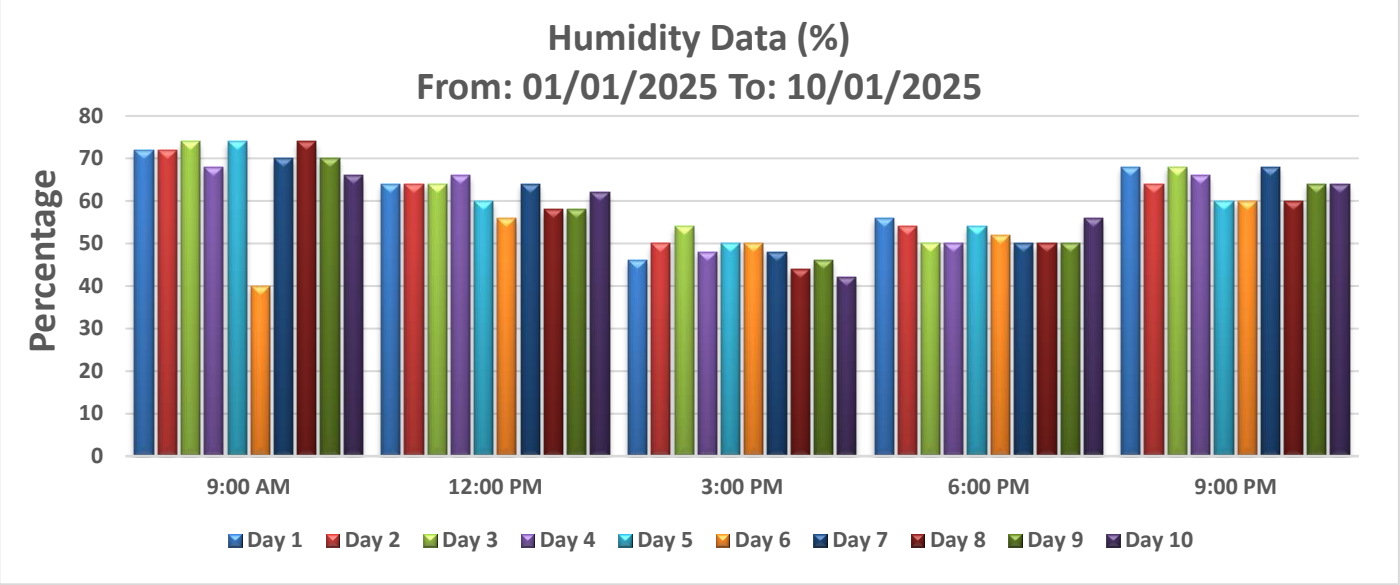


Fig: Measured Humidity data and graph of Ten days.

Humidity Data (%) For Internet										
Time	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
9:00AM	74	70	72	68	70	38	72	70	72	68
12:00PM	62	64	60	64	60	58	62	60	58	62
3:00PM	44	48	50	46	48	50	46	42	46	40
6:00PM	58	50	52	54	56	50	52	48	52	54
9:00PM	68	62	66	68	64	60	64	62	64	60

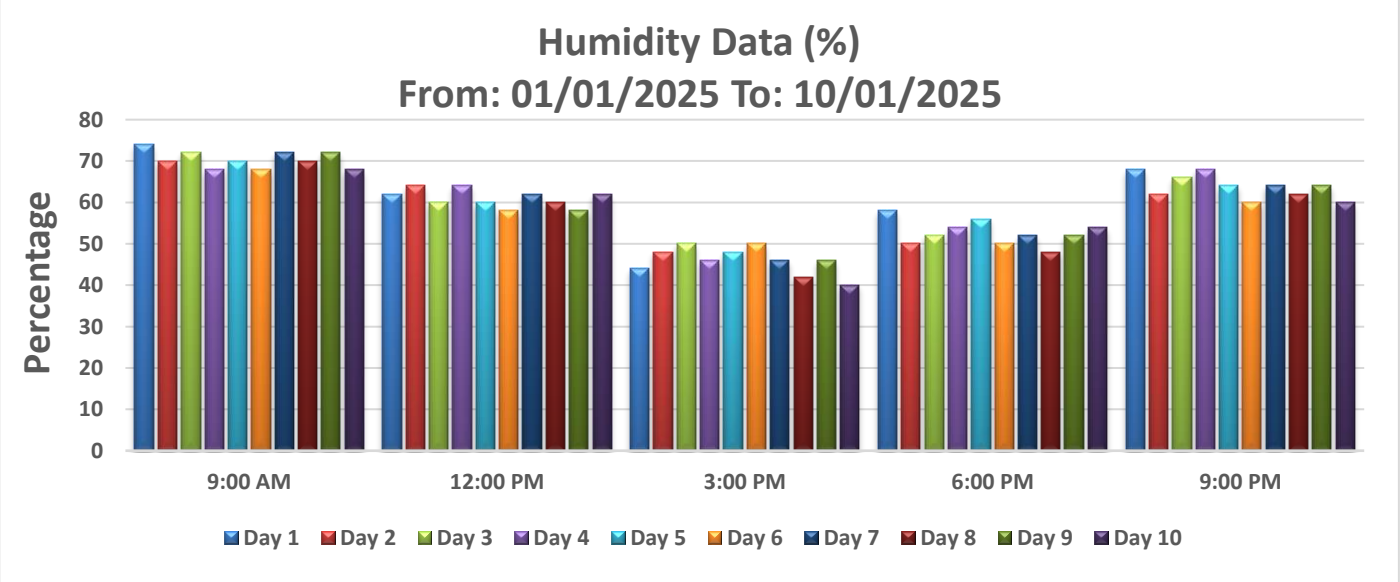


Fig: Measured Humidity data and graph of Ten days.

Wind Speed Data (KM/H) For Our Project										
Time	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
9:00AM	4.8	5.7	4.8	3.7	4.2	3.2	3.3	6.8	11.3	6.8
12:00PM	5.9	5.8	4.4	4.8	3.3	4.5	4.7	10.4	7.7	3.2
3:00PM	6.7	4.4	3.4	3.4	6.9	3.3	4.5	5.5	5.6	5.2
6:00PM	4.4	4.2	3.8	3.2	4.2	3.4	4.8	5	4.6	3.4
9:00PM	5.6	5.5	3.6	4.8	3.2	3.7	3.3	6.8	3.2	3.3

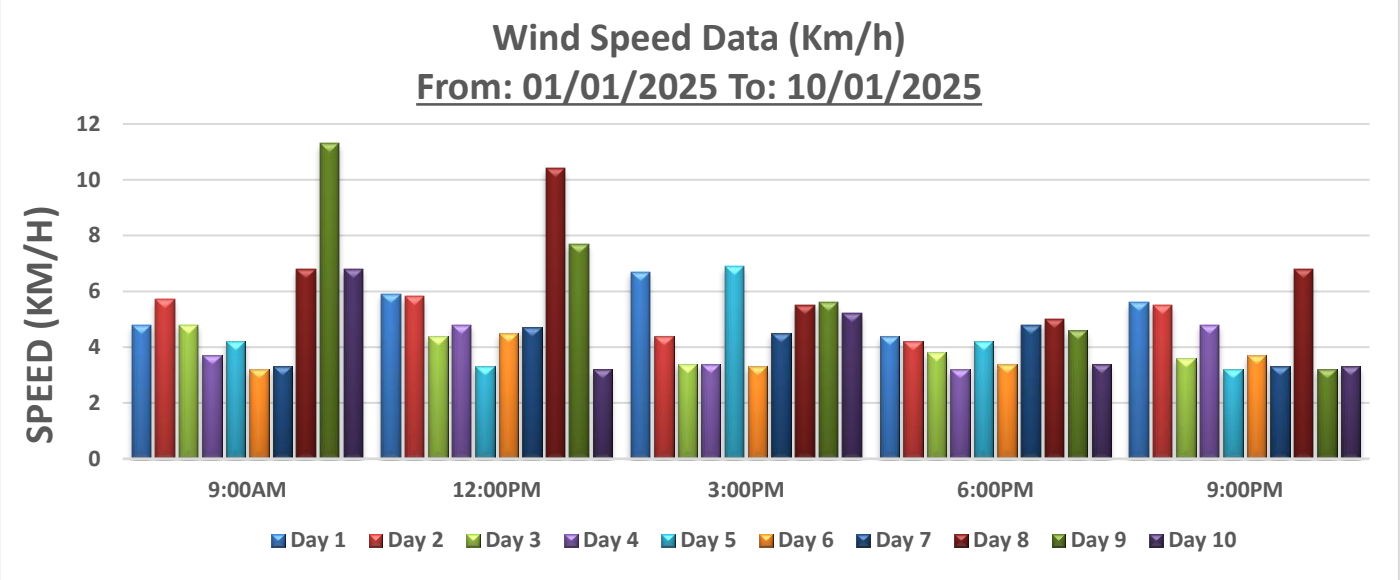


Fig: Measured wind speed data and graph of Ten days.

Wind Speed Data (KM/H) For Internet										
Time	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**
9:00AM	4.6	5.7	4.6	3.5	3.8	3.2	3.4	6.9	11.5	6.9
12:00PM	5.7	5.7	4.6	4.6	3.5	4.6	4.6	10.4	8	3.4
3:00PM	6.9	4.6	3.4	3.4	6.9	3.4	4.6	5.7	5.7	5.5
6:00PM	4.6	4.6	3.4	3.4	3.6	3.4	4.4	4.2	4.6	3.4
9:00PM	5.7	5.7	3.4	3.7	3.4	3.5	3.4	6.9	3.4	3.4

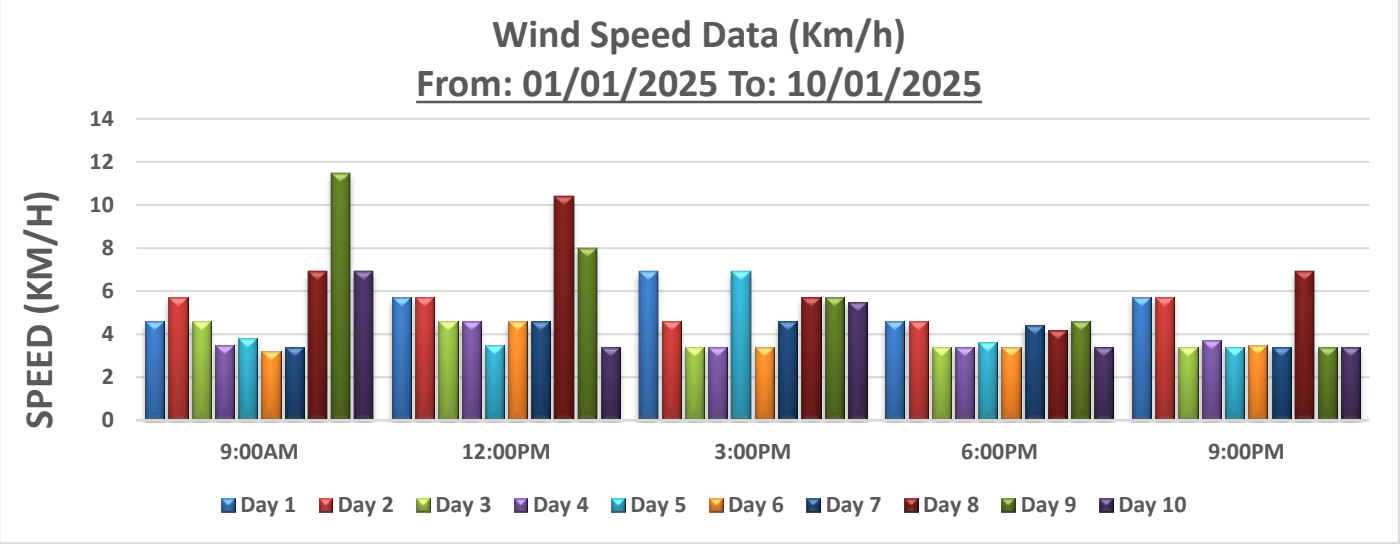


Fig: Measured wind speed data and graph of Ten days.

Air Pressure Data (pa) For Our Project										
Time	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
9:00AM	101794	101896	101794	101693	101896	101388	101591	101693	101896	102199
12:00PM	101388	101490	101490	101388	101591	101286	101388	101591	101693	101794
3:00PM	101185	101286	101591	101286	101388	101185	101286	101286	101388	101591
6:00PM	101155	101591	101388	101185	101286	101085	101085	101490	101591	101490
9:00PM	101591	101693	101490	101490	101591	101185	101185	101591	101693	101896

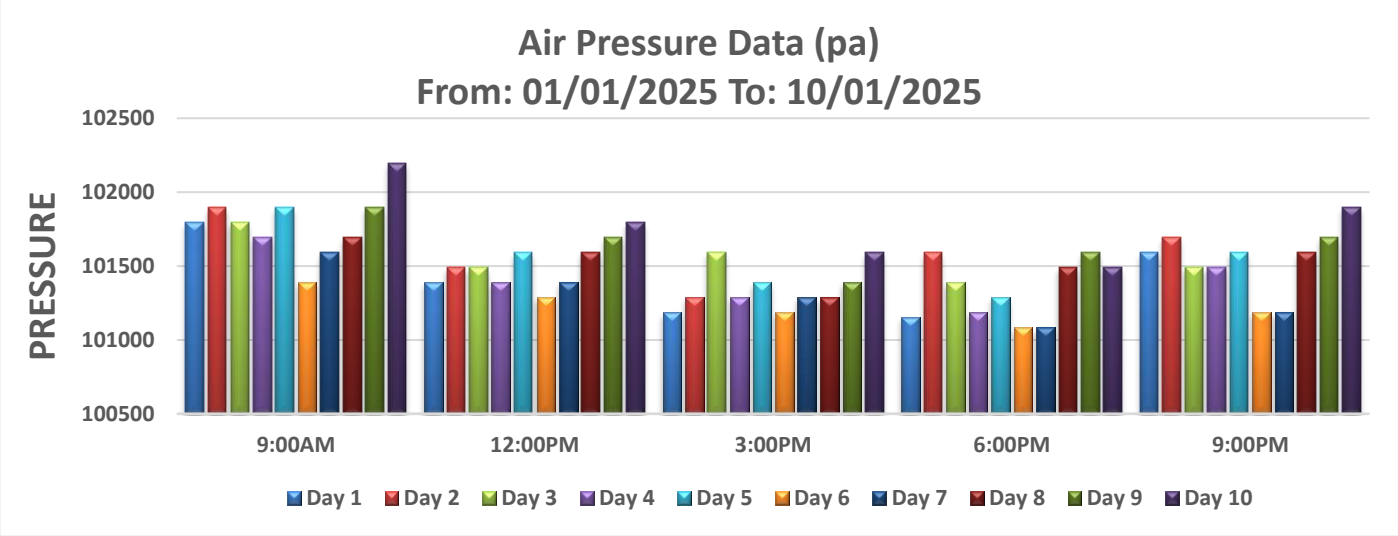


Fig: Measured Air Pressure data and graph of Ten days.

Air Pressure Data (pa) For Internet										
Time	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
9:00AM	101591	101794	101693	101591	101693	101490	101490	101591	101896	102099
12:00PM	101490	101591	101591	101388	101490	101388	101286	101591	101693	101896
3:00PM	101286	101388	101388	101185	101286	101083	101185	101286	101490	101591
6:00PM	101286	101388	101388	101185	101286	101185	101185	101388	101490	101490
9:00PM	101490	101591	101490	101388	101490	101286	101083	101591	101693	101794

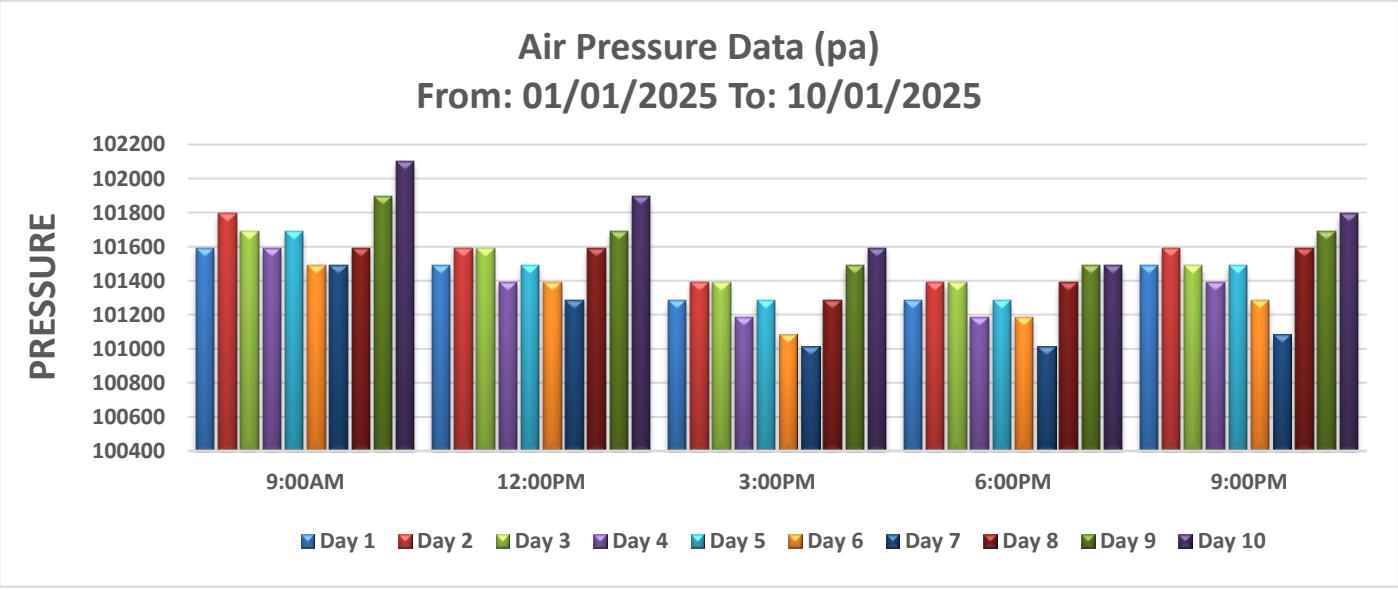
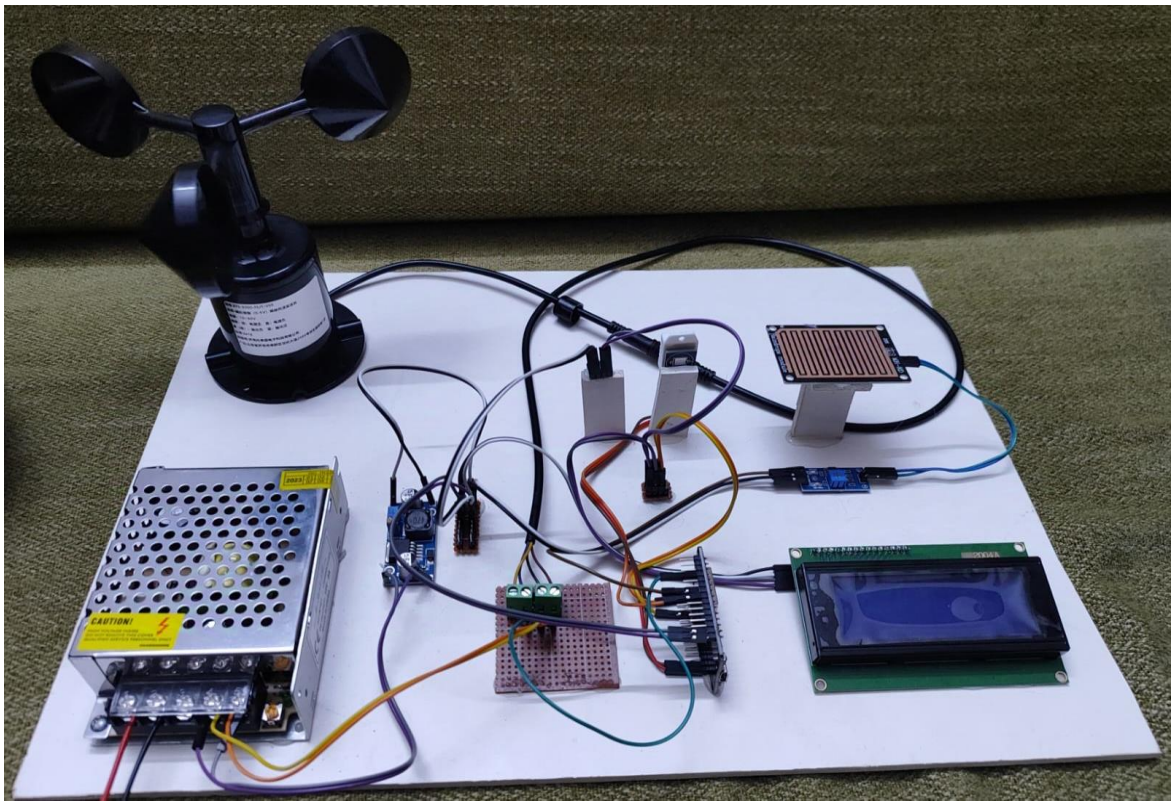


Fig: Measured Air Pressure data and graph of Ten days.

Rain Drop Data (mm)										
Time	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
9:00AM	No Rain Detected	Rain Detected	Rain Detected	Rain Detected	Rain Detected	Rain Detected	Rain Detected	Rain Detected	Rain Detected	Rain Detected
12:00PM	Rain Detected	Rain Detected	Rain Detected	No Rain Detected	No Rain Detected	Rain Detected	No Rain Detected	No Rain Detected	No Rain Detected	Rain Detected
3:00PM	Rain Detected	No Rain Detected	Rain Detected	No Rain Detected	No Rain Detected	No Rain Detected	No Rain Detected	No Rain Detected	Rain Detected	Rain Detected
6:00PM	Rain Detected	Rain Detected	No Rain Detected	No Rain Detected	Rain Detected	Rain Detected	No Rain Detected	Rain Detected	Rain Detected	No Rain Detected
9:00PM	No Rain Detected	Rain Detected	No Rain Detected	No Rain Detected	No Rain Detected	No Rain Detected	Rain Detected	Rain Detected	No Rain Detected	Rain Detected

Measured Rain data and graph of Ten days.

4.3 Project Prototype Image



Chapter 05

Conclusion

5.1 Discussion

The discussion delves into the effectiveness of the IoT-based communication system. The stable data transmission via Wi-Fi or cellular networks to a central server or cloud platform demonstrates the robustness of the design. The use of secure communication protocols like MQTT and HTTPS ensures data integrity and user privacy. Additionally, the user interface, whether web-based or mobile, has shown to be user-friendly, allowing for real-time monitoring, alerts, and historical data analysis.

However, the discussion also addresses some challenges and limitations encountered during the implementation. For instance, sensor calibration and accuracy in varying environmental conditions may require additional attention. Factors such as extreme temperatures, humidity, or mechanical wear can affect sensor performance, necessitating regular calibration and maintenance. The power supply system, particularly in remote locations, might also face challenges with battery life or solar panel efficiency during low-light conditions.

Another key point in the discussion is the scalability and adaptability of the weather station. The modular design allows for the integration of additional sensors or components, but this flexibility may introduce complexities in terms of system stability and data management. Future iterations might need to focus on ensuring scalability without compromising performance.

Furthermore, the discussion explores the broader applications and benefits of the weather station. The collected data can be valuable for various fields, such as agriculture, urban planning, and disaster preparedness. The ability to monitor weather conditions in real-time and predict weather patterns can contribute to more informed decision-making and enhanced safety measures.

5.2 Application

- In applications like flood lines, reservoirs, hydroelectricity, water treatment, and urban drainage a weather station to measure rainfall, soil moisture.
- In airports, weather stations can monitor real-time wind direction, wind speed, air temperature, air pressure and other meteorological elements to provide the necessary meteorological information for flights and ensure the safety and punctuality of flights.
- Ports are also one of the important areas where weather stations are set up. In the harbor, the weather station can monitor the wind direction, wind speed, air temperature, air pressure and other meteorological elements in real time, providing necessary meteorological information for ships to ensure the safety and navigation of ships.
- Weather stations are also beginning to be used in campuses. In the campus, the weather station can monitor the weather elements in real time, let the students understand the weather knowledge through the way of popular science education, and improve the students' understanding of the natural environment and protection awareness.

In short, weather stations are widely used in various fields, providing important support and guarantee for people's life and production activities.

An IoT-based weather station can be incredibly useful across various domains:

- **Agriculture**: Farmers can use real-time weather data to optimize irrigation schedules, monitor soil moisture, and predict weather patterns for better crop management.
- **Urban Planning**: City authorities can utilize data from weather stations to manage infrastructure such as drainage systems, traffic lights, and public transportation during adverse weather conditions like heavy rainfall or snowstorms.
- **Disaster Management**: Weather stations can provide crucial information for disaster preparedness and response teams. They can help predict and monitor natural disasters like hurricanes, tornadoes, or floods, enabling timely evacuation and resource allocation.

- **Energy Management**: Utility companies can optimize energy distribution based on weather forecasts. For instance, they can adjust the supply of renewable energy sources like solar and wind power according to weather conditions.
- **Transportation**: Real-time weather data can improve road safety by alerting drivers to hazardous conditions like ice, fog, or heavy rain. It can also help logistics companies optimize routes and schedules to minimize delays.
- **Outdoor Events**: Organizers of outdoor events such as concerts or sports games can use weather data to make informed decisions about scheduling, seating arrangements, and contingency plans in case of inclement weather.
- **Research**: Meteorologists and climate scientists rely on accurate and extensive weather data for research and analysis to understand long-term climate patterns and trends.

Overall, an IoT-based weather station offers a wide range of applications across industries and sectors, providing valuable insights for decision-making and resource management.

5.3 Conclusion

The **"Environmental Condition Measurement System Using a Mini Weather Station"** has been designed and tested successfully. The hardware components and of all integrated features has used to developed the system. All of the Presence of each module has been placed carefully and reasoned out, as the unit can contribute for best working. Thus, an IoT-based weather station presents a multifaceted solution with numerous benefits across various sectors. By leveraging the power of interconnected sensors and data analytics, these stations offer real-time and accurate weather information that can revolutionize decision-making processes. From agriculture to urban planning, disaster management to energy optimization, transportation to personal use, the applications are diverse and impactful.

With the ability to monitor and predict weather conditions with precision, IoT weather stations enhance efficiency, safety, and resource management in numerous industries. They enable proactive measures in response to adverse weather events, optimize resource allocation, and facilitate informed decision-making at both individual and organizational levels. As technology continues to advance, IoT-based weather stations are poised to play an increasingly crucial role in our lives, helping us adapt to and mitigate the impacts of weather variability and climate change.

REFERENCES

[1] Kulkarni, V. A, Satpute G. M (2017). "Weather Reporting System Using FPGA : A Review," vol. 4, no. 11, pp. 319–320.

[2] Carlos, M, Jorge, P.B, Daniel F, Pablo S (2018). "Design, Development and Implementation of a Weather Station Prototype for Renewable Energy System," Journal Energies, 11(9), 2234, pp. 1-13.

[3] Karim F, Karim F and Frihida A (2017). "Monitoring system using web of things in precision agriculture," Procedia Computer Science., vol. 110, pp. 402–409.

[4] Kodali R K, Yerroju S and Sahu S (2018). "Smart Farm Monitoring Using LoRa Enabled IoT," Proceedings 2nd International Conference Green Computing Internet Things, ICGCIoT 2018, pp. 391–394.

[5] Gahlot N, Gundkal V, Kothimbire S and Thite A (2015). "Zigbee based weather monitoring system," Internnational Journal Engineering Science, vol. 4, no. 4, pp. 2319–1813.

[6] Kodali R K and Sahu A (2016). "An IoT based weather information prototype using WeMos," Proceedings 2016 2nd International Conference Contemporary Computing Informatics, IC3I 2016, no. 1, pp. 612–616.

[7] Joe F, and Joseph J (2019). "IoT Based Weather Monitoring System for Effective Analytics," International Journal of Engineering and Advanced Technology (IJEAT), no. 4, pp. 311– 315.

[8] Nallakaruppan, M.K, and Kumaran U.S (2019). "IoT based Machine Learning Techniques for Climate Predictive Analysis," International Journal of Recent Technology and Engineering (IJRTE), no. 5, pp. 171–175.

[9] Monteiro M S, De Caldas Filho F L, Barbosa L A, Martins L M C E, De Menezes J T M and Da Silva Filho D A(2019). "University campus microclimate monitoring using IoT," WCNPS 2019 - Workshop Communication Networks Power Systems, no. Wcnps, pp. 3–7.

[10] N. Gahlot, V. Gundkal, S. Kothimbire, A. Thite , " Zigbeebased weather monitoring system", The IJES, vol. 4, pp.61 -62, 2015.

[11] V. Pujari, M. Raste, A. Pujari , "Cost Effective Automated Weather Station – A Review", IJEEE, vol. 8, pp. 582 -585, January-June 2016.

[12] K.A Noordin, C.C Onn , M.F Ismail, "A Low -Cost Microcontroller-based Weather Monitoring System", CMU Journal, vol. 5(1), pp.33 -36, 2006.

[13] H. Mohammed, S.Dave , "Wireless Weather Monitoring System using Arduino DUE and GSM Technology", The IRJET, vol. 5, pp 746 -748, April2018.

[14] https://www.google.com/url?sa=i&url=https%3A%2F%2Fcommunity.simplefoc.com%2Ft% 2Fesp32-wroom-32d-dev-board-schematics-pcb%2F1293&psig=AOvVaw1uPFYftxsyQRlAz9toO18i&ust=1715759495170000&source= images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCNiHm4bVjIYDFQAAAAAdAAA AABAE

[15] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.upesy.com%2Fblogs%2Ftut orials%2Fesp32-pinout-reference-gpio-pins-ultimate-guide&psig=AOvVaw3UBrI9OvREK8p3Im_57-qu&ust=1715759577885000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFw oTCNi4hfrUjIYDFQAAAAAdAAAAABAE&h=675&hcb=2&ved=2ahUKEwi949vR1IyGA xWkT2wGHXeEA-sQM3oECF8QAA

[16] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.ubuy.com.bd%2Fen%2Fpro duct%2F1VT8I46-100v-240v-to-24v-1a-ac-dc-switching-power-supply-adapter-with-5- selectable-adapter-plugs&psig=AOvVaw1ccuAYyoAtVt9ufsnWIY0f&ust=1715759768928000&source=images &cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCNjN3K7VjIYDFQAAAAAdAAAAABA E

[17] https://www.google.com/url?sa=i&url=https%3A%2F%2Fstore.roboticsbd.com%2Fsensors% 2F923-dht22-original-am2302-digital-temperature-humidity-sensor-module-robotics-bangladesh.html&psig=AOvVaw3gQxRNRVNdNrhd2SxWKis2&ust=1715759837669000& source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCKDSxs7VjIYDFQAAAA AdAAAAABAE

[18] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.alibaba.com%2Fproduct- detail%2FNew-and-original-Atmega328-Atmega-328_1600491712380.html&psig=AOvVaw0YmYlSl_YDoeAwiqFTmyBC&ust=1715759891 691000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCODM2evVjIYDF QAAAAAdAAAAABAE

[19] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.rs-online.com%2Fdesignspark%2Fbasics-of-atmega328&psig=AOvVaw1bduxI63WFOJ-IUXrKz1UB&ust=1715759969306000&source=images&cd=vfe&opi=89978449&ved=0CBI QjRxqFwoTCMj0_ZPWjIYDFQAAAAAdAAAAABAE

[20] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.altronics.com.au%2Fp%2Fz 6395-barometric-air-pressure-sensor-breakout-for-arduino%2F&psig=AOvVaw0afvhLeBoM5Eclcte7IRFl&ust=1715760052335000&source=i mages&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCNjXhbnWjIYDFQAAAAAdAAA AABAK

[21] https://www.google.com/url?sa=i&url=https%3A%2F%2Fm.indiamart.com%2Fproddetail%2 Fconnecting-wire-10738053962.html&psig=AOvVaw1cTK-yBxFSn1pH2JuhIxNk&ust=1715760141903000&source=images&cd=vfe&opi=89978449&v ed=0CBIQjRxqFwoTCJiht9_WjIYDFQAAAAAdAAAAABAE

[22] https://www.google.com/url?sa=i&url=https%3A%2F%2Fstore.roboticsbd.com%2Fdisplay% 2F1642-lcd-16x2-blue-backlight-robotics-bangladesh.html&psig=AOvVaw3NIIxka5tRrxtfSmx-zfUJ&ust=1715760191898000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqF woTCJDZ0oPXjIYDFQAAAAAdAAAAABAE

[23] https://www.google.com/url?sa=i&url=https%3A%2F%2Fstore.roboticsbd.com%2Fsensors% 2F665-humidity-detection-sensor-module-rain-detection-for-arduino.html&psig=AOvVaw2GdM6JMoScg-NUI2x1YiVd&ust=1715760265396000&source=images&cd=vfe&opi=89978449&ved=0CB IQjRxqFwoTCNCKs53XjIYDFQAAAAAdAAAAABAE

[24] Reference for Coding: Used Arduino IDE software library and YouTube

APPENDIX [24]

```
#define BLYNK_PRINT Serial

#define BLYNK_TEMPLATE_ID "TMPL66shV2hFI"
#define BLYNK_TEMPLATE_NAME "IoT Weather Station"
#define BLYNK_AUTH_TOKEN "as4Awtcvg5wO0bjPjUEvMZQRK5T60Sr6"


#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#include <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;
int pressure;

int wind = 34;
int wind_data;

#define DHTPIN 25      // what digital pin we're connected to
#define DHTTYPE DHT22  // DHT 22  (AM2302), AM2321
DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27, 20, 4);
int pos = 0;
char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "student";
char pass[] = "iotstudent";


int rain = 33;
int raindata;

// int pump = D3;

int flag = 0;
int pinValue;

float h;
float t;

void setup() {

  // Debug console
  Serial.begin(9600);
  pinMode(wind, INPUT);

  bmp.begin();

  dht.begin();
```

```
  lcd.init();
  lcd.backlight();

  // Set ESP8266 baud rate
  Blynk.begin(auth, ssid, pass);
  //timer.setInterval(1000L, myTimerEvent);
}


void loop() {
  Blynk.run();

  //timer.run(); // Initiates BlynkTimer

  raindata = analogRead(rain);
  Serial.println(raindata);

  pressure = bmp.readPressure();

  wind_data = analogRead(wind);
  float outvolt = wind_data * (3.9 / 4090);
  int level = 50 * outvolt;

  h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  t = dht.readTemperature();


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

  Serial.print("Temperature = ");
  Serial.print(t);
  Serial.println("C");

  Serial.print("Humidity = ");
  Serial.print(h);
  Serial.println("%");

  Serial.print("wind:");
  Serial.println(level);

  Serial.println(".....");

  Blynk.virtualWrite(V0, t);         //virtual pin V0
  Blynk.virtualWrite(V1, h);         //virtual pin V0
  Blynk.virtualWrite(V2, pressure);  //virtual pin V0
  Blynk.virtualWrite(V3, level);     //virtual pin V0

  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Wind Speed:");
  lcd.print(level);
```

```
  lcd.print("Km/h");

  lcd.setCursor(0, 1);
  lcd.print("Pressure:");
  lcd.print(pressure);
  lcd.print("Pa");

  lcd.setCursor(0, 2);
  lcd.print("T:");
  lcd.print(t);
  lcd.print("C");

  // lcd.setCursor(0, 1);
  lcd.print(" | H:");
  lcd.print(h);
  lcd.print("%");


  if (raindata < 500) {
    Blynk.virtualWrite(V4, "Rain Detected!");
    lcd.setCursor(0, 3);
    lcd.print("Rain Detected!");
  }

  if (raindata > 500) {
    Blynk.virtualWrite(V4, "No Rain Detected");
    lcd.setCursor(0, 3);
    lcd.print("No Rain Detected");
  }
}
```