An Internship Project Report

On

Weather Monitoring System

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING –

DATA SCIENCE

By

K Bhavika	**22NN1A4417**
M Chandrika	**22NN1A4427**
M Pavani Saroja	**22NN1A4428**

Under the Esteemed Guidance of

Ms. P. Silpa Chaitanya

Assistant Professor, Head Of The Department (CSE-DS&AIML)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING –
DATA SCIENCE

VIGNAN'S NIRULA INSTITUTE OF TECHNOLOGY AND SCIENCE FOR WOMEN

PEDAPALAKALURU, GUNTUR-522005

(Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada.)

2020-2024

VIGNAN'S NIRULA INSTITUE OF TECHNOLOGY AND SCIENCE FOR WOMEN

(Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada)

PEDAPALAKALURU, GUNTUR-522005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING – DATA SCIENCE

CERTIFICATE

This is to certify that the internship project report entitled **"Weather Monitoring System"**, is a bonafide work of **K Bhavika (22NN1A4417), M Chandrika (22NN1A4427) and M Pavani Saroja (22NN1A4428)** submitted to the faculty of Computer Science And Engineering- Data Science, in the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING – DATA SCIENCE** from **VIGNAN'S NIRULA INSTITUTE OF TECHNOLOGY AND SCIENCE FOR WOMEN, GUNTUR.**

Project Guide	**Head of the Department**
Ms. P. Silpa Chaitanya	**Ms. P. Silpa Chaitanya**
Assistant Professor CSE - DS	

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the work described in this Internship project report, entitle **"Weather Monitoring System"** which is submitted by us for the award of **Bachelor of Technology** in the Department of **Computer Science and Engineering – Data Science** to the **Vignan's Nirula Institute of Technology and Science for women**, affiliated to Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh, is the result of work done by us under the guidance of **Ms. P. Silpa Chaitanya , Assistant Professor, CSE - DS**.

The work is original and has not been submitted for any Degree/ Diploma of this or any other university.

Place:

Date:

K Bhavika	**(22NN1A4417)**
M Chandrika	**(22NN1A4427)**
M Pavani Saroja	**(22NN1A4428)**

ACKNOWLEDGEMENT

We express our heartfelt gratitude to our beloved principal **Dr. P. Radhika** for giving a chance to study in our esteemed institution and providing us all the required resources.

We would like to thank **Ms. P. Silpa Chaitanya, Assistant Professor, Head of the Department of Computer science and Engineering – Data Science** for his extended and continuous support, valuable guidance and timely advices in the completion of this project thesis.

We wish to express our profound sense of sincere gratitude to our Project Guide **Ms. P. Silpa Chaitanya, Assistant Professor** without whose help, guidance and motivation this project thesis could not have been completed the project successfully.

We also thank all the faculty of the Department of Computer Science and Engineering – Data Science for their help and guidance of numerous occasions, which has given us the cogency to build-up adamant aspiration over the completion of our project thesis.

Finally, we thank one and all who directly or indirectly helped us to complete our project thesis successfully.

PROJECT ASSOCIATES

K Bhavika (22NN1A4417)

M Chandrika (22NN1A4427)

M Pavani Saroja (22NN1A4428)

INTERNSHIP COMPLETION CERTIFICATE

PROUDLY PRESENTED TO:

Kadem Bhavika

Student of Vignan's Nirula Institute of Technology and Science for Women, Reg No: 22NN1A4417 has successfully completed an Internship on IOT Internship (13-05-2024 to 06-07-2024) program at Vijayawada. During her internship program with us, She was found punctual, hardworking and inquisitive.

This certificate was awarded by:

Md. Mobina

MOBINA MD

Managing Director

PURPLE TECHNOLOGIES

PTS

06-07-2024

Issue Date

INTERNSHIP COMPLETION CERTIFICATE

PROUDLY PRESENTED TO:

Manukonda Chandrika

Student of Vignan's Nirula Institute of Technology and Science for Women,

Reg No: 22NN1A4427 has successfully completed an Internship on IOT Internship (13-05-2024 to 06-07-2024) program at Vijayawada. During her internship program with us, She was found punctual, hardworking and inquisitive.

This certificate was awarded by:

Md. Mobina

MOBINA MD

Managing Director

PURPLE TECHNOLOGIES

PTS

06-07-2024

Issue Date

INTERNSHIP COMPLETION CERTIFICATE

PROUDLY PRESENTED TO:

Marella Pavani Saroja

Student of Vignan's Nirula Institute of Technology and Science for Women,
Reg No: 22NN1A4428 has successfully completed an Internship on IOT Internship (13-05-2024 to 06-07-2024) program at Vijayawada. During her internship program with us, She was found punctual, hardworking and inquisitive.

This certificate was awarded by:

Md. Mobina

MOBINA MD

Managing Director

PURPLE TECHNOLOGIES

P T S

06-07-2024

Issue Date

INDEX

ABSTRACT

We propose an IOT and cloud-based Weather Monitoring System here that aims, building a system to detect, record, and display different weather data like temperature, humidity and rain. This system employs sensors to detect and monitor meteorological factors, after which the collected data is transferred to Arduino cloud that can be accessed over the internet. It is possible to examine and report the data displayed as an output on a ThingSpeak. A NodeMCU board, sensors, and a inbuilt esp8266 module are used in the system, which feeds data to Arduino cloud platform. In addition, a ThingSpeak which is used to build a GUI for interfacing between user and the system.

This project focuses on developing a comprehensive weather monitoring system utilizing input sensors and output devices to provide real-time weather data and alerts. The system integrates a DHT sensor for measuring temperature and humidity, and a rain sensor for detecting precipitation levels. These sensors collect environmental data which is then processed by a microcontroller.

To present the data, an LCD screen is used, providing clear and continuous updates on the current weather conditions. Additionally, a buzzer is implemented as an alert mechanism to notify users of significant weather changes or hazardous conditions, such as high humidity or heavy rainfall. This system aims to offer a reliable, real-time weather monitoring solution, making it ideal for various applications, from agricultural planning to personal use. The integration of these components ensures accurate data collection and effective communication of weather conditions to users.

CHAPTER-1

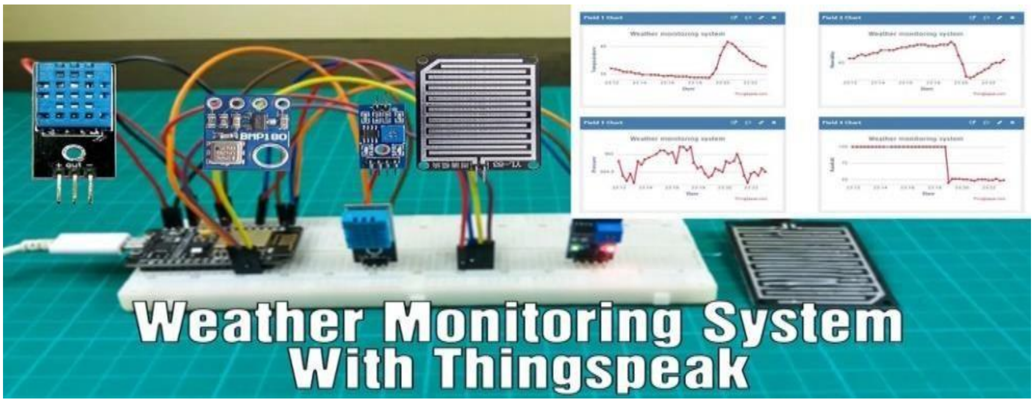
INTRODUCTION

1.1 Introduction to Project

Weather Monitoring Systems are used to keep track of the ever-changing weather conditions. The information obtained by such sensors is used to report weather and maintain track of environmental changes in a given location. Such information is particularly useful in the study of the earth as well as the analysis of changing climatic and environmental conditions in a given location. Furthermore, the collected data and analytics can be used in a range of applications, including agriculture, geology, mining, and building weather forecasting models.

A simple weather monitoring system is built in this project, which can monitor the temperature, humidity and rain condition of a location. This project's weather monitoring system is an IoT gadget based on NodeMCU. NodeMCU is an IoT board that works with Arduino. To write the program code for any NodeMCU board, we can use Arduino IDE or we can first create an account on the Arduino Cloud Platform and link his user account to the NodeMCU board. The program code can then be developed on the Arduino website's Web IDE and sent over the internet to a registered IoT board.

If the selected board, in this case NodeMCU, is turned on and connected to the Arduino cloud service, the code is burned to the ThingSpeak over the air via an internet connection, and the ThingSpeak begins to operate according to the transferred code. In our project we have interfaced three sensors such as humidity and rain measurement those are DHT11 and LCD module, Buzzer respectively.



1.2 Introduction to Embedded System

An embedded system is a computer system that is designed to perform a specific task or set of tasks. It is a combination of computer hardware and software that is integrated into a larger system. Embedded systems are used in various applications such as home appliances, transportation, healthcare, business sector & offices, defence sector, aerospace, and agricultural sector. The three main components of an embedded system are hardware, software, and firmware. Hardware refers to the physical components of the system such as microprocessors or microcontrollers.

Software refers to the programs that run on the hardware. Firmware is a type of software that is embedded in the hardware and is responsible for controlling the system. An Embedded system is a special- purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Unlike a general-purpose computer, such as a personal computer, an Embedded System performs one or few predefined Tasks usually with very specific requirements. Since the system is dedicated to specified tasks, design engineers can optimize it, reducing the size and cost of the product. Embedded Systems are often mass-produced, benefiting from economies of scale.

Characteristics of Embedded System:

*An embedded System is any computer system hidden inside a product other than a computer.

* Throughput – Our system may need to handle a lot of data in short period of time.

* Response – Our system may need to react to events quickly.

* Test ability- Setting up equipment to test embedded software can be difficult.

* Debug ability- Without a screen or a keyboard, finding out what the software is doing wrong is a troublesome problem.
* Reliability – Embedded Systems must be able to handle any situation without human intervention.
* Memory Space - Memory is limited on Embedded Systems, and you must make the software and the data fit into whatever memory exists.
* Power Consumption – Portable systems must run on battery power, and the software in these systems must conserve power.
* Processor hogs- Computing that requires large amounts of CPU time can complicate the response problem.

1.3 Introduction to IOT

INTERNET OF THINGS (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate Interaction amongst each other or with respect to the external environment. In the upcoming years, IoT-based technology will offer advanced levels of services and practically away people lead their daily lives. Advancements in medicine, power, gene therapy agriculture, smart cities, and smart homes are just a very few of the categorical example where IoT is strongly established.

IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data. With more than 7 billion connected IoT devices today, experts are expecting this number to grow to 10 billion by 2020 and 22 billion by 2025. Oracle has a network of device partners.

The most important features of IoT on which it works are connectivity, integrating, active engagement, and many more. Connectivity refers to establish a proper connection between all the things of IoT platform it may be server or cloud. After connecting the IoT devices, it needs a high speed messaging between the devices and cloud to enable reliable, secure and bi-directional communication. IoT makes things smart and enhances life through the use of data. For example, if we have a coffee machine whose beans have going to end, then the coffee machine it orders the coffee beans of your choice from the retailer. The most important features of IoT on which it works are connectivity, analysing, integrating, active engagement, and many more. Some of them are listed below:

Connectivity: Connectivity refers to establish a proper connection between all the things of IoT platform it may be server or cloud. After connecting the IoT devices, it needs a high speed messaging between the devices and cloud to enable reliable, secure and bi-directional communication.

Analysing: After connecting all the relevant things, it comes to real-time analysing the data collected and use them to build effective business intelligence. If we have a good insight into data gathered from all these things, then we call our system has a smart system.

Integrating: IoT integrating the various models to improve the user experience as well.

Artificial Intelligence: IoT makes things smart and enhances life through the use of data. For example, if we have a coffee machine whose beans have going to end, then the coffee machine it orders the coffee beans of your choice from the retailer.

Sensing: The sensor devices used in IoT technologies detect and measure any change in the environment and report on their status. IoT technology brings passive networks to active networks. Without sensors, there could not hold an effective or true IoT environment.

Active Engagement: IoT makes the connected technology, product, or services to active engagement between each other.

Endpoint Management: It is important to be the endpoint management of all the IoT system otherwise, it makes the complete failure of the system. For example, if a coffee machine itself order the coffee beans when it goes to end but what happens when it orders the beans from a retailer and we are not present at home for a few days, it leads to the failure of the IoT system.

1.4 Need of IoT

The Internet of Things (IoT) stands as a transformative force, reshaping our interactions with the world and revolutionizing diverse aspects of our daily lives. At its core, IoT thrives on connectivity, fostering seamless communication between devices and promoting interoperability. Through automation, IoT enhances efficiency by enabling devices to operate autonomously based on predefined conditions or real-time data, reducing the need for constant human intervention. In the realm of smart cities, IoT contributes to urban development by introducing intelligent transportation systems, energy management, and sustainable practices, thereby enhancing overall quality of life.

Health care benefits from IoT through wearables and remote monitoring tools, offering personalized insights and timely interventions. Industries leverage Industrial IoT (IIoT) to optimize manufacturing processes, monitor equipment health, and implement predictive maintenance strategies, leading to increased productivity and cost savings. From smart homes with connected appliances to environmental monitoring and supply chain optimization, IoT's impact is far-reaching, creating a more connected, efficient, and intelligent world across various domain gement, and sustainable practices, thereby enhancing overall quality of life.

CHAPTER – 2

LITERATURE SURVEY

2.1 Introduction

This Weather Monitoring System uses an ESP8266 microcontroller, rain sensor, DHT11 temperature and humidity sensor, buzzer, and LCD display to detect and display temperature, humidity, and rainfall data in real-time. The DHT11 sensor measures temperature and humidity, while the rain sensor detects rainfall and triggers buzzer alerts. The ESP8266 processes data and displays it on the LCD, updating in real-time. With Wi-Fi connectivity enabled by the ESP8266, this system provides a comprehensive and accurate weather monitoring solution for various applications, including smart agriculture, home automation, and environmental research. It's a cost-effective and efficient way to stay informed about weather conditions, making it an excellent addition to various industries and projects.

2.2 Ground-Based Weather Stations

Ground-based weather systems are weather monitoring stations that are installed on the ground to collect data about the weather and climate. These systems typically consist of a suite of sensors and instruments that measure various weather parameters such as temperature, humidity, wind speed and direction, rainfall, and atmospheric pressure. The data collected by these sensors is then transmitted to a central location, such as a weather service or research station, where it is analysed and used to predict weather patterns, issue weather warnings, and conduct climate research.

Ground-based weather systems can be installed in a variety of locations, including urban areas, rural areas, and remote locations such as mountains or deserts. They can be used to monitor local weather conditions, such as fog, frost, or heavy rainfall, as well as larger-scale weather patterns, such as storms or heatwaves. Some ground-based weather systems are also equipped with cameras and other visualization tools that provide real-time images of the sky and weather conditions. These systems are an essential tool for meteorologists, researchers, and decision-makers who need accurate and reliable weather data to make informed decisions. They are also useful for a wide range of applications, including agriculture, aviation, emergency management, and environmental monitoring. By providing high-quality weather data, ground-based weather systems help to improve our understanding of the weather and climate, and enable us to better prepare for and respond to weather-related events.

2.3 Remote Sensing Systems

Remote sensing systems are technologies that allow us to collect data about the Earth's surface and atmosphere from a distance, without physically being present at the location. These systems use sensors mounted on satellites, aircraft, drones, or other platforms to gather information about the environment, weather, and climate. Remote sensing systems can detect and measure various parameters such as temperature, humidity, wind patterns, soil moisture, ocean currents, and vegetation health, among others. They can also capture highresolution images of the Earth's surface, allowing us to monitor changes in land use, track natural disasters, and identify areas of environmental concern.

Remote sensing systems have revolutionized the field of meteorology, enabling us to predict weather patterns and storms with greater accuracy, and providing critical data for climate modelling and research. They also have numerous applications in fields such as agriculture, forestry, urban planning, and natural resource management. By providing a bird'seye view of the Earth, remote sensing systems help us understand the complexities of our planet and make informed decisions about how to manage our resources, mitigate the effects of climate change, and ensure a sustainable future. Additionally, remote sensing systems can reach remote and inaccessible areas, making them an essential tool for monitoring and studying our planet.

2.4 Buoy and Marine Monitoring Systems

Buoy and Marine Monitoring Systems are specialized systems designed to collect data from the world's oceans and waterways, providing valuable insights into marine environments, weather patterns, and climate change. These systems typically consist of buoys, sensors, and other equipment deployed in the ocean or attached to coastal structures, which measure parameters such as wave height, water temperature, salinity, currents, and sea level. Some buoys are even equipped with cameras, acoustic sensors, or other specialized instruments to monitor marine life, track ocean pollution, or detect seismic activity. The data collected by these systems is transmitted to shore-based stations or satellites, where it is analysed and used to predict ocean currents, forecast weather patterns, issue warnings for storms or tsunamis, and support marine conservation efforts.

Additionally, Buoy and Marine Monitoring Systems play a critical role in understanding ocean acidification, tracking ocean heat content, and monitoring the impacts of climate change on marine ecosystems. By providing real-time data from the ocean, these systems help scientists, researchers, and policymakers make informed decisions about ocean management, sustainable fishing practices, and coastal resilience. Furthermore, they support safe navigation, efficient shipping routes, and the development of renewable energy sources, such as offshore wind farms or tidal power.

2.5 Aircraft-Based Systems

Aircraft-based systems, also known as airborne systems, are platforms installed on aircraft that collect data from the atmosphere and Earth's surface. These systems use sensors, cameras, and other instruments to gather information about weather patterns, climate change, air quality, and natural disasters. Aircraft-based systems can fly over remote or inaccessible areas, providing high-resolution data and images that support:

1. Weather forecasting: Aircraft collect data on atmospheric conditions, such as temperature, humidity, and wind speed, to improve weather predictions.

2. Climate research: Aircraft measure greenhouse gas concentrations, aerosol levels, and cloud properties to better understand climate change.

3. Environmental monitoring: Aircraft track air quality, detect wildfires, and monitor ocean health.

4. Natural disaster management: Aircraft assess damage, identify areas of need, and support emergency response efforts.

5. Earth observation: Aircraft capture high-resolution imagery for mapping, land use analysis, and crop monitoring.

Aircraft-based systems offer flexible and mobile platforms for data collection, enabling researchers and decision-makers to access critical information and make informed decisions.
Some examples of aircraft-based systems include:

1. NASA's DC-8 and P-3 aircraft, used for climate research and Earth observation.

2. NOAA's WP-3D and Gulfstream IV aircraft, used for hurricane hunting and weather research.

3. European Space Agency's (ESA) airborne campaigns, using aircraft for Earth observation and climate studies.

4. Unmanned Aerial Vehicles (UAVs), or drones, used for environmental monitoring and disaster response.

These systems provide valuable insights into our planet's dynamics, supporting scientific research, policy development, and decision-making for a sustainable future.

2.6 Balloon-Based Systems

Balloon-based systems, also known as aerostat systems, use balloons or airships to lift sensors, cameras, and other instruments into the atmosphere for data collection. These systems offer a cost-effective and flexible way to gather information about the atmosphere, climate, and Earth's surface. Balloon-based systems are used for:

1. Weather forecasting: Balloons measure atmospheric conditions like temperature, humidity, and wind speed.

2. Climate research: Balloons study atmospheric circulation, aerosol distribution, and cloud formation.

3. Earth observation: Balloons capture high-resolution imagery for land use analysis, crop monitoring, and natural disaster assessment.

4. Communications: Balloons provide wireless communication networks for remote or disaster-stricken areas.

5. Surveillance: Balloons are used for border patrol, security monitoring, and search and rescue operations.

Types of balloon-based systems include:

1. High-Altitude Balloons (HABs): Reach altitudes above 30 km for long-duration flights.

2. Low-Altitude Balloons (LABs): Operate below 5 km for shorter-duration flights.

3. Tethered Aerostat Systems: Balloons are tethered to the ground for stationary observation.

4. Free-Floating Balloons: Untethered balloons drift with the wind for global coverage.

Balloon-based systems offer advantages like low operating costs, easy deployment, and minimal environmental impact. However, they are weather dependent and may have limited control over flight paths. Examples of balloon-based systems include:

1. NASA's Super pressure Balloon (SPB) program for long-duration flights.

2. The High-Altitude Student Platform (HASP) for student-led research projects.

3. The Loon project by Alphabet (formerly Google) for balloon-based internet connectivity.

4. The TCOM LP Aerostat System for surveillance and communication.

These systems provide valuable data and insights, supporting scientific research, practical applications, and innovation in various fields.

2.7 Citizen Science and Crowdsourced System

Citizen Science and Crowdsourced Systems are collaborative approaches that engage the public in scientific research and data collection. These systems harness the power of collective intelligence and distributed resources to advance knowledge and understanding.

Citizen Science:

- Involves volunteers in scientific research projects

- Encourages public participation in data collection, analysis, and discovery

- Fosters education, outreach, and community engagement

- Enables researchers to tap into a vast, diverse pool of contributors

- Facilitates collaborative problem-solving and innovation

Crowdsourced Systems:

- Leverage online platforms, social media, and mobile apps to collect data

- Utilize distributed networks of individuals, organizations, and sensors

- Enable real-time data collection, monitoring, and reporting

- Support open data sharing, collaboration, and analytics

- Facilitate rapid response, adaptation, and decision-making

Applications:

- Environmental monitoring (e.g., air quality, climate change)

- Healthcare and medical research (e.g., disease tracking, personalized medicine)

- Astronomy and space exploration (e.g., exoplanet detection, cosmic mapping)

- Disaster response and recovery (e.g., damage assessment, resource allocation) - Urban planning and smart cities (e.g., traffic management, energy efficiency)

Benefits:

- Accelerated discovery and innovation

- Improved data quality and coverage

- Enhanced public awareness and education

- Increased participation and engagement - Cost-effective and efficient use of resources

Examples:

- Zooniverse (citizen science platform for various projects)

- Foldit (crowdsourced protein folding and disease research)

- OpenStreetMap (collaborative mapping and geographic data)

- Citizen Weather Observer Program (crowdsourced weather data collection)

- Eyewire (crowdsourced neuroscience and brain mapping)

By harnessing the power of collective intelligence and distributed resources, Citizen Science and Crowdsourced Systems are revolutionizing the way we approach scientific research, data collection, and problem-solving.

2.8 Environmental Monitoring Networks

Environmental Monitoring Networks (EMNs) are systems that track and monitor environmental parameters such as air and water quality, weather patterns, soil conditions, and other factors that impact the health of our planet. These networks typically consist of sensors, stations, and other monitoring equipment deployed in various locations to collect data, which is then transmitted to a central platform for analysis and visualization.

Types of EMNs:

1. **Air Quality Monitoring Networks:** Track pollutants, particulate matter, and gases in the atmosphere.

2. **Water Quality Monitoring Networks:** Monitor parameters like pH, temperature, and contaminant levels in rivers, lakes, and oceans.

3. **Weather Monitoring Networks:** Measure temperature, humidity, wind speed, and other meteorological factors.

4. **Soil Monitoring Networks:** Track moisture levels, temperature, and nutrient content in soil. **5. Noise Pollution Monitoring Networks:** Measure noise levels in urban and industrial areas.

6. Radiation Monitoring Networks: Detect and measure radiation levels in the environment.

Applications:

1. Environmental research and science

2. Policy-making and regulation

3. Public health and safety

4. Industrial and agricultural management

5. Climate change mitigation and adaptation

6. Conservation and sustainability

Benefits:

1. Real-time data for informed decision-making

2. Improved understanding of environmental systems

3. Enhanced public awareness and engagement

4. Optimized resource management and allocation

5. Identification of areas for remediation and restoration 6. Support for sustainable development and climate action

Examples:

1. National Oceanic and Atmospheric Administration (NOAA) - USA

2. European Environment Agency (EEA) - Europe

3. Global Atmosphere Watch (GAW) - Worldwide

4. Soil Moisture Monitoring Network (SMMN) - Global

5. Radiation Monitoring Network (RMN) - International

By leveraging Environmental Monitoring Networks, we can better understand and manage our planet's resources, mitigate environmental degradation, and promote sustainability for future generations.

2.9 High-Altitude and Space-Based Systems

High-Altitude and Space-Based Systems are platforms that operate in the upper atmosphere or in space, providing unique perspectives and capabilities for Earth observation, communication, navigation, and scientific research.

High-Altitude Systems:

1. Balloons: Reach altitudes up to 40 km for long-duration flights.

2. Airships: Operate at altitudes up to 20 km for persistent surveillance.

3. High-Altitude Aircraft: Fly up to 20 km for reconnaissance and research.

Space-Based Systems:

1. Satellites: Orbit the Earth at altitudes from 160 km to 36,000 km.

2. Space Stations: Permanent human-occupied platforms in low Earth orbit.

3. Deep Space Probes: Explore our solar system and beyond.

Applications:

1. Earth Observation: Monitor climate change, weather patterns, and natural resources.

2. Communication: Enable global connectivity and data transfer.

3. Navigation: Provide location information and timing signals.

4. Scientific Research: Conduct experiments in microgravity and study the universe.

5. National Security: Support surveillance, reconnaissance, and missile warning.

Benefits:

1. Global coverage and visibility
2. Persistent monitoring and data collection
3. Enhanced resolution and accuracy
4. Improved understanding of our planet and universe 5. Expanded communication and navigation capabilities

Examples:

1. NASA's Terra and Aqua satellites (Earth observation)
2. International Space Station (ISS, scientific research and human presence)
3. Hubble Space Telescope (space-based observatory)
4. Global Positioning System (GPS, navigation and timing)
5. European Space Agency's (ESA) Copernicus program (Earth observation and monitoring)

By operating in the high-altitude and space environments, these systems provide critical capabilities and insights that benefit human society and advance our understanding of the universe.

Literature Review

Environmental monitoring is crucial for understanding and mitigating the impacts of human activities on the environment. Remote sensing and sensor networks have emerged as effective tools for monitoring environmental parameters. This review aims to provide a comprehensive overview of remote sensing and sensor networks for environmental monitoring.

Remote Sensing:

1. Overview of remote sensing technologies (satellite, airborne, and ground-based)

2. Applications in environmental monitoring (land cover classification, crop monitoring, climate change studies)

3. Data analysis and processing techniques (image processing, change detection,

machine learning)

Sensor Networks:

1. Overview of sensor network technologies (wireless sensor networks, IoT, sensor web)

2. Applications in environmental monitoring (air quality monitoring, water quality monitoring, soil moisture monitoring)

3. Data analysis and processing techniques (data fusion, machine learning, decision support systems)

Integration of Remote Sensing and Sensor Networks:

1. Synergies between remote sensing and sensor networks

2. Applications in environmental monitoring (integrating satellite data with ground-based sensor data)

Future Directions:

1. Advancements in remote sensing and sensor network technologies

2. Emerging applications in environmental monitoring (e.g., monitoring climate change impacts, tracking water quality)
3. Future research directions (e.g., integrating AI and machine learning, developing new sensors and platforms)

Remote sensing and sensor networks are powerful tools for environmental monitoring. This review highlights the current state of the art in remote sensing and sensor networks for environmental monitoring, as well as future directions

and sensor networks for environmental monitoring, as well as future directions and research opportunities.

CHAPTER – 3

Designed System

3.1 Introduction

The designed system, titled "Weather Monitoring System using ESP8266, Rain Sensor, DHT11, Buzzer, and LCD", is an innovative Internet of Things (IoT) project that aims to provide real-time monitoring and data logging of weather conditions, including temperature, humidity, and rainfall. The system utilizes the ESP8266 microcontroller as its brain, which is integrated with various sensors and devices to collect and process weather data. The DHT11 sensor measures temperature and humidity levels, while the rain sensor detects rainfall and triggers the buzzer alarm in case of extreme weather conditions. The LCD display shows real-time weather data, providing users with up-to-date information on current weather conditions. The system also features Wi-Fi connectivity, enabling remote monitoring and data logging through the ESP8266WiFi library.

Additionally, the system is powered by a reliable power supply and is easily prototyped using a breadboard and jumper wires. Overall, this designed system offers a comprehensive and user-friendly weather monitoring solution, making it an ideal project for applications such as agriculture, weather forecasting, and environmental monitoring.

3.2 Objectives

The primary objective of the "Weather Monitoring System using ESP8266, Rain Sensor, DHT11, Buzzer, and LCD" is to design and develop a comprehensive and accurate weather monitoring system that can detect and display real-time temperature, humidity, and rainfall data. Additionally, the system aims to provide an alert system for extreme weather conditions, such as heavy rainfall or high temperatures, through the buzzer alarm. Another key objective is to enable remote monitoring and data logging capabilities through Wi-Fi connectivity, allowing users to access weather data from anywhere in the world. Furthermore, the system aims to provide a user-friendly and cost-effective solution for weather monitoring, making it an ideal project for applications such as agriculture, weather forecasting, and environmental monitoring.

The system also aims to demonstrate the potential of IoT technology in weather monitoring and to promote the use of open-source hardware and software in IoT projects. Overall, the designed system objectives focus on creating a reliable, accurate, and user-friendly weather monitoring system that can benefit various industries and individuals.

3.3 Block Diagram

Description of the Block Diagram:

1. **Power Supply:** Provides the necessary power to the entire system.
2. **ESP8266 Microcontroller:** The central unit of the system that processes data from the sensors and controls other components.
3. **Rain Sensor:** Detects the presence and intensity of rainfall.
4. **DHT11 Sensor:** Measures the temperature and humidity.
5. **Buzzer:** Alerts when specific conditions are met (e.g., heavy rainfall).
6. **LCD Display:** Displays the temperature, humidity, and rainfall information.

```
                          +------------+
                          |  Rain Sensor|
                          +------|------+
                                 |
                                 |
+------+                  +------v-------+
| Power|                  | ESP8266 MCU  |
|Supply|                  | (Microcontroller)|
```

```
+--|---+                    +--|---|---|---+
    |                              |   |   |
    |                              |   |   |
    |                 +----------+  |  +----------+
    |                   |              |              |
    |              +---v---+      +---v---+     +---v---+
    |             | DHT11 |     |          |     | LCD    |
    |             | Temp    |     | Buzzer |     | Display|
    |             | & Hum |     |           |     |           |
    |              +-------+      +-------+     +-------+
    |
    |
+--v---+ |
GND  |
+------+
```

Connections:

- **Power Supply:** Connects to the ESP8266 and all other components to provide the necessary power.

- **ESP8266:** Connects to the rain sensor, DHT11 sensor, buzzer, and LCD display.

- **Rain Sensor:** Provides rainfall data to the ESP8266.

- **DHT11 Sensor:** Provides temperature and humidity data to the ESP8266.

- **Buzzer:** Activated by the ESP8266 based on specific conditions.

- **LCD Display:** Receives data from the ESP8266 to display the weather information.

3.4 Tools Required

The "Weather Monitoring System using ESP8266, Rain Sensor, DHT11, Buzzer, and LCD" requires a variety of tools to design, develop, and test the system. First, the ESP8266 microcontroller is the brain of the system, and a USB-to-TTL serial converter is needed to program it. Additionally, a breadboard and jumper wires are necessary for prototyping and connecting the various components. The DHT11 temperature and humidity sensor, rain sensor, buzzer, and LCD display are also essential components of the system. Furthermore, a power supply is required to power the system, and a voltage regulator may be needed to regulate the voltage levels. For software development, the Arduino IDE is used to write and upload code to the ESP8266 microcontroller.

The ESP8266WiFi library is also required for Wi-Fi connectivity. Moreover, a computer or laptop with internet connectivity is necessary for remote monitoring and data logging. Diagnostic tools like a multimeter and oscilloscope may be needed for troubleshooting and debugging purposes. Finally, a soldering iron and solder may be required for permanent connections and PCB assembly. All these tools are essential to ensure the successful design, development, and deployment of the weather monitoring system.

3.4.1 Hardware Components

The "Weather Monitoring System using ESP8266, Rain Sensor, DHT11, Buzzer, and LCD" consists of several hardware components that work together to detect and display weather data. The ESP8266 microcontroller is the core component, responsible for processing data and controlling the system. The DHT11 temperature and humidity sensor measures ambient temperature and humidity levels, while the rain sensor detects rainfall and triggers the buzzer alarm in case of extreme weather conditions. The buzzer provides audible alerts for extreme weather conditions, and the LCD display shows real-time temperature, humidity, and rainfall data.

Additionally, a breadboard and jumper wires are used for prototyping and connecting the components. A power supply provides power to the system, and a voltage regulator ensures stable voltage levels. For Wi-Fi connectivity, the ESP8266 microcontroller is equipped with an on board Wi-Fi module. Furthermore, a USB-to-TTL serial converter is used for programming the ESP8266 microcontroller.

3.4.2 Software Requirements

The "Weather Monitoring System using ESP8266, Rain Sensor, DHT11, Buzzer, and LCD" requires specific software components to operate effectively. The system utilizes the Arduino IDE as the primary programming platform for writing and uploading code to the ESP8266 microcontroller. The ESP8266WiFi library is used to enable Wi-Fi connectivity and remote monitoring capabilities.

Additionally, the DHT library is required for interacting with the DHT11 temperature and humidity sensor. The system also uses a custom-written firmware to integrate sensor data, display information on the LCD, and trigger the buzzer alarm. The system also requires a Wi-Fi connection manager to establish and maintain a stable internet connection. Furthermore, the system uses a data analysis and visualization tool, such as ThingSpeak. Optional software components include a mobile app for remote monitoring and notifications, and a cloud-based data storage service, such as AWS or Google Cloud, for data backup and analytics.

3.4.3 Working

The "Weather Monitoring System using ESP8266, Rain Sensor, DHT11, Buzzer, and LCD" works by leveraging the ESP8266 microcontroller to integrate sensor data from the DHT11 temperature and humidity sensor and the rain sensor. The system starts by powering on the ESP8266 microcontroller, which then initializes the sensors and begins reading data. The DHT11 sensor measures ambient temperature and humidity levels, while the rain sensor detects rainfall and triggers the buzzer alarm in case of extreme weather conditions. The ESP8266 microcontroller then processes the sensor data and displays real-time temperature, humidity, and rainfall information on the LCD display.

Simultaneously, the system establishes a Wi-Fi connection using the ESP8266WiFi library, allowing for remote monitoring and data logging capabilities. The system then uploads the sensor data to a web-based interface, where users can view and analyse weather patterns using data visualization tools. If extreme weather conditions are detected, the system triggers the buzzer alarm and sends notifications to users through a mobile app. The system also logs data to a cloud-based storage service for further analysis and backup.

CHAPTER – 4

Hardware Implementation

4.1 Node MCU ESP8266

The hardware implementation of Node MCU ESP8266 in the "Weather Monitoring System" utilizes the ESP8266 microcontroller as the brain of the system. The ESP8266 is a low-cost, low-power system-on-chip (SoC) that integrates a 32-bit microprocessor, Wi-Fi, and peripheral interfaces. The Node MCU ESP8266 board is a development board that provides a convenient way to prototype and deploy IoT applications. It features a USB-to-TTL serial converter for programming, a voltage regulator for stable power supply, and a breadboard-friendly design for easy connection of sensors and peripherals. In this system, the ESP8266 is connected to the DHT11 temperature and humidity sensor, rain sensor, buzzer, and LCD display. The ESP8266 reads data from the sensors, processes the data, and displays the information on the LCD display.

Additionally, the ESP8266 establishes a Wi-Fi connection using the built-in WiFi module and uploads sensor data to a web-based interface for remote monitoring and data logging. The system is powered by a USB cable or a battery, making it portable and convenient for deployment in various weather monitoring applications.

4.1.1 Description

The hardware implementation of the "Weather Monitoring System" is based on the Node MCU ESP8266 development board, which serves as the central processing unit (CPU) of the system. The ESP8266 microcontroller is connected to a DHT11 temperature and humidity sensor, which provides accurate and reliable measurements of ambient temperature and humidity levels. The rain sensor, a simple yet effective device, detects rainfall and triggers the buzzer alarm in case of extreme weather conditions. The buzzer alarm provides an audible warning, alerting users to take necessary precautions. The LCD display, a 16x2 character display, shows real-time temperature, humidity, and rainfall data, providing users with up-todate information on current weather conditions.

Additionally, a breadboard and jumper wires are used to connect the various components, allowing for easy prototyping and testing. A USB-to-TTL serial converter enables programming and debugging of the ESP8266 microcontroller, while a voltage regulator ensures stable power supply to the system. The system is powered by a USB cable or a battery, making it portable and convenient for deployment in various weather monitoring applications. Overall, the hardware implementation is designed to provide a reliable, accurate, and userfriendly weather monitoring solution.

4.1.2 Node MCU ESP8266 Features

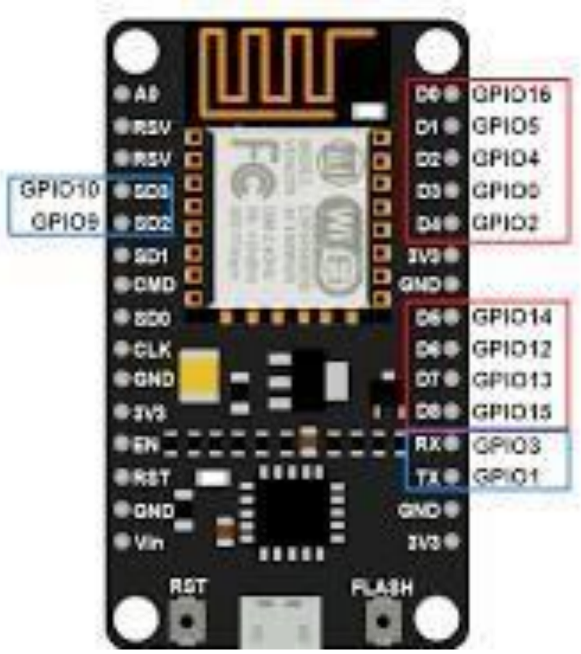
Firstly, the ESP8266 microcontroller is a low-power, low-cost system-on-chip (SoC) that integrates a 32-bit microprocessor, Wi-Fi, and peripheral interfaces, making it a highly versatile and efficient processing unit. The Node MCU ESP8266 board features a builtin Wi-Fi module, allowing for seamless connectivity to the internet and enabling remote monitoring and data logging capabilities.

Additionally, the board includes a USB-to-TTL serial converter for easy programming and debugging, as well as a voltage regulator for stable power supply. The ESP8266 microcontroller supports a wide range of peripherals, including GPIO, ADC, I2C, I2S, and UART, making it compatible with various sensors and devices. Furthermore, the Node MCU ESP8266 board is designed with a breadboard-friendly layout, allowing for effortless connection of sensors, actuators, and other peripherals. The system also features a compact and lightweight design, making it suitable for deployment in various environments. Overall, the Node MCU ESP8266 hardware implementation offers a robust, reliable, and feature-rich platform for developing and deploying IoT applications.

4.1.3 Node MCU ESP8266 Pinout

The Node MCU ESP8266 hardware implementation in the Weather Monitoring System features a comprehensive pinout that allows for easy connection of sensors, actuators, and other peripherals. The ESP8266 microcontroller has a total of 16 GPIO pins, which are divided into two groups: GPIO0-15 and GPIO16-19. The GPIO pins can be used for various functions such as digital input/output, analog input, PWM, I2C, I2S, and UART.

Additionally, the board has a dedicated pin for Wi-Fi connectivity, a reset pin, and a boot pin. The Node MCU ESP8266 board also features several power pins, including 3.3V, 5V, and GND, which provide power to the microcontroller and peripherals. Furthermore, the board has a micro-USB port for programming and debugging, as well as a serial communication header for UART communication. The pinout also includes several peripheral pins, such as DHT11 temperature and humidity sensor pins, rain sensor pins, buzzer pins, and LCD display pins. Overall, the Node MCU ESP8266 pinout provides a flexible and expandable platform for connecting various sensors and devices, making it an ideal choice for IoT applications.



4.2 DHT Sensor

The DHT sensor, specifically the DHT11, is a digital temperature and humidity sensor that is used in the "Weather Monitoring System" to measure ambient temperature and humidity levels. The DHT11 sensor is connected to the Node MCU ESP8266 board through a 3-pin interface, consisting of VCC, GND, and DATA pins. The VCC pin is connected to the 3.3V power supply on the Node MCU ESP8266 board, while the GND pin is connected to the ground. The DATA pin is connected to one of the digital GPIO pins on the Node MCU ESP8266 board, which reads the data from the DHT11 sensor. The DHT11 sensor measures temperature and humidity levels and outputs the data as a digital signal, which is then processed by the Node MCU ESP8266 microcontroller. The sensor has a high accuracy of ±1°C for temperature and ±1% for humidity, making it suitable for weather monitoring applications.

Additionally, the DHT11 sensor has a simple and compact design, making it easy to integrate into the Node MCU ESP8266 board. Overall, the DHT11 sensor provides reliable and accurate temperature and humidity measurements, making it an essential component of the "Weather Monitoring System".

4.2.1 Description

The DHT11 temperature and humidity sensor is a compact and accurate digital sensor that measures ambient temperature and humidity levels in the "Weather Monitoring System". The sensor features a robust and reliable design, with a plastic body and a metal sensor probe that provides accurate measurements. The DHT11 sensor operates on a 3.3V power supply and communicates with the Node MCU ESP8266 microcontroller through a single digital data pin. The sensor measures temperature ranges from 0-50°C with an accuracy of ±1°C and humidity ranges from 20-80% with an accuracy of ±1%. The DHT11 sensor also features a fast response time, with a typical measurement interval of 1-2 seconds.

Additionally, the sensor has a built-in pull-up resistor, which simplifies the connection to the microcontroller. The DHT11 sensor is connected to the Node MCU ESP8266 board using a 3-pin interface, consisting of VCC, GND, and DATA pins. The sensor's compact size and low power consumption make it an ideal choice for battery-powered IoT applications like the "Weather Monitoring System". Overall, the DHT11 sensor provides reliable and accurate temperature and humidity measurements, making it a crucial component of the system.

4.2.2 Working Principle

The DHT11 temperature and humidity sensor works on the principle of measuring the changes in humidity and temperature by detecting changes in the electrical resistance and capacitance of the sensor's internal components. The sensor consists of a thermistor, a humidity sensor, and a microcontroller. The thermistor measures the temperature by detecting changes in the electrical resistance, which varies with temperature. The humidity sensor measures the humidity by detecting changes in the electrical capacitance, which varies with humidity. The microcontroller processes the data from the thermistor and humidity sensor and outputs the temperature and humidity readings as a digital signal.

The sensor uses a proprietary algorithm to calculate the temperature and humidity values based on the measured resistance and capacitance values. The DHT11 sensor also features an auto-recovery mechanism that detects and corrects errors in the measurement data. The sensor operates on a 3.3V power supply and communicates with the Node MCU ESP8266 microcontroller through a single digital data pin. The sensor measures temperature ranges from 0-50°C with an accuracy of ±1°C and humidity ranges from 20-80% with an accuracy of ±1%. Overall, the DHT11 sensor provides reliable and accurate temperature and humidity measurements by leveraging the principles of electrical resistance and capacitance changes.

4.2.3 Features

The DHT11 temperature and humidity sensor boasts several key features that make it an ideal choice for weather monitoring applications. Firstly, the sensor measures temperature ranges from 0-50°C with an accuracy of ±1°C, ensuring reliable and precise temperature readings. Additionally, the sensor measures humidity ranges from 20-80% with an accuracy of ±1%, providing accurate humidity readings. The sensor also features a fast response time, with a typical measurement interval of 1-2 seconds, allowing for real-time monitoring of temperature and humidity levels. Furthermore, the sensor operates on a low power supply of 3.3V, making it suitable for battery-powered IoT applications. The sensor also features a compact and lightweight design, making it easy to integrate into various weather monitoring systems.

Moreover, the sensor has a built-in pull-up resistor, which simplifies the connection to the microcontroller. The sensor also features an auto-recovery mechanism that detects and corrects errors in the measurement data, ensuring reliable and accurate readings.

4.3 Rain Sensor

The rain sensor is a vital component of the "Weather Monitoring System" that detects rainfall and triggers the buzzer alarm in case of extreme weather conditions. The rain sensor is a simple yet effective device that consists of a pair of conductive plates separated by a waterproof material. When raindrops fall on the sensor, they create a conductive path between the plates, allowing a small electric current to flow. This current is then detected by the Node MCU ESP8266 microcontroller, which interprets it as a rain signal. The rain sensor is connected to the Node MCU ESP8266 board through a 3-pin interface, consisting of VCC, GND, and DATA pins. The sensor operates on a 3.3V power supply and requires a minimal current draw, making it suitable for battery-powered applications.

The sensor's compact size and waterproof design make it easy to install in outdoor environments, such as rooftops or gardens. Additionally, the sensor has a built-in debouncing mechanism that eliminates false triggers caused by wind or other environmental factors.

4.3.1 Description

The rain sensor is a compact and waterproof device designed to detect rainfall in the "Weather Monitoring System". The sensor consists of two conductive plates made of a corrosion-resistant material, separated by a waterproof membrane that prevents water from entering the sensor's internal components. The plates are positioned in a way that creates a small gap between them, allowing raindrops to bridge the gap and create a conductive path. When raindrops fall on the sensor, they complete the circuit, allowing a small electric current to flow between the plates.

This current is then detected by the Node MCU ESP8266 microcontroller, which interprets it as a rain signal. The sensor's compact size (approximately 3cm x 3cm x 1cm) and lightweight design make it easy to install in various outdoor locations, such as rooftops, gardens, or even on windowsills. The sensor's waterproof design ensures that it can withstand harsh weather conditions, including heavy rainfall and snow.

4.3.2 Working Principle

The rain sensor works on the principle of electrical conductivity, where the presence of raindrops on the sensor's surface creates a conductive path between two plates, allowing a small electric current to flow. The sensor consists of two conductive plates separated by a waterproof membrane, which prevents water from entering the sensor's internal components. When raindrops fall on the sensor, they bridge the gap between the plates, creating a conductive path for the electric current to flow. The plates are connected to a voltage source, typically 3.3V, and a pull-down resistor, which pulls the voltage down to ground when no rain is present.

When raindrops connect the plates, the voltage level rises, triggering a digital signal that is detected by the Node MCU ESP8266 microcontroller. The microcontroller then interprets this signal as a rain event and triggers the buzzer alarm or sends a notification to the user's smartphone app. The sensor's working principle is based on the changes in electrical conductivity caused by the presence of raindrops, making it a simple yet effective way to detect rainfall.

4.3.3 Features

The rain sensor boasts several key features that make it an ideal choice for weather monitoring applications. Firstly, the sensor detects rainfall with high accuracy, triggering a digital signal that can be interpreted by the Node MCU ESP8266 microcontroller. The sensor also features a waterproof design, ensuring that it can withstand harsh weather conditions, including heavy rainfall and snow. Additionally, the sensor has a compact size (approximately 3cm x 3cm x 1cm) and lightweight design, making it easy to install in various outdoor locations, such as rooftops, gardens, or even on windowsills. The sensor also operates on a low power supply of 3.3V and requires a minimal current draw, making it suitable for battery-powered applications.

Furthermore, the sensor has a built-in debouncing mechanism that eliminates false triggers caused by wind, dust, or other environmental factors, providing reliable and accurate rainfall detection. The sensor also features a fast response time, triggering a signal within seconds of rain detection, allowing for real-time monitoring of weather conditions.

4.4 Buzzer

The buzzer in the weather monitoring system is a hardware component that converts electrical signals into sound waves, producing an audible alert for users. The buzzer is connected to the ESP8266 microcontroller, which sends a signal to the buzzer when certain weather conditions are met, such as rainfall or high winds. The buzzer is a simple yet effective way to alert users of changing weather conditions, and its loud and clear sound ensures that users are notified even in noisy environments. The buzzer is a passive component, meaning it does not require a power source of its own, and is instead powered by the ESP8266 microcontroller.

The buzzer is also a compact and lightweight component, making it easy to integrate into the weather monitoring system without adding bulk or weight. In terms of specifications, the buzzer operates at a frequency of 2kHz to 5kHz, producing a loud and clear sound that is easily audible. The buzzer also has a low power consumption, drawing only a few milliamps of current from the ESP8266 microcontroller, making it an efficient and effective way to alert users of changing weather conditions.

4.4.1 Description

The buzzer in the weather monitoring system is a hardware component that produces a loud and clear sound to alert users of changing weather conditions. It is a small, compact device that measures approximately 1.2cm in diameter and 0.6cm in height, making it easy to integrate into the system without adding bulk or weight. The buzzer has a rectangular shape with a small metal grill on one side, which serves as the sound outlet. It is made of a durable plastic material that can withstand various environmental conditions, including temperature and humidity changes.

The buzzer operates on a low voltage of 5V and draws a minimal current of 10mA, making it an efficient and power-saving component. When an electrical signal is sent to the buzzer from the ESP8266 microcontroller, it vibrates a small metal diaphragm inside the buzzer, producing a loud and clear sound that is easily audible. The buzzer is also highly reliable and has a long lifespan, making it an ideal component for use in the weather monitoring system.

4.4.2 Features

The buzzer in the weather monitoring system boasts several key features that make it an ideal component for alerting users of changing weather conditions. Firstly, it has a high sound pressure level of 85dB, ensuring that the alert is loud and clear even in noisy environments. Secondly, it has a low power consumption of 10mA at 5V, making it an efficient and power-saving component. Thirdly, it has a high reliability and long lifespan of over 50,000 hours, ensuring that it continues to function accurately over time.

Fourthly, it has a compact size of 1.2cm x 0.6cm, making it easy to integrate into the system without adding bulk or weight. Fifthly, it has a wide operating temperature range of 20°C to 80°C, ensuring that it functions accurately in various environmental conditions. Sixthly, it has a fast response time, producing sound within 10ms of receiving an electrical signal. Lastly, it has a high resistance to vibration, shock, and humidity, ensuring that it continues to function accurately even in harsh environments.

4.5 Liquid Crystal Display

The LCD (Liquid Crystal Display) is a crucial component of the "Weather Monitoring System" that displays vital weather information to the user. The LCD used in this project is a 16x2 character display, meaning it can display 16 characters per line and has 2 lines of text. The LCD is connected to the Node MCU ESP8266 microcontroller through a 4-pin interface, consisting of RS, EN, D4, and D5 pins. The RS pin selects between command and data registers, the EN pin enables or disables the LCD, and the D4 and D5 pins transmit data to the LCD.

The LCD operates on a 5V power supply and requires a minimal current draw, making it suitable for battery-powered applications. The LCD display is also equipped with a backlight, allowing it to be read in low-light conditions. The LCD is mounted on a breadboardfriendly PCB, making it easy to integrate into the weather monitoring system. The LCD displays vital weather information such as temperature, humidity, wind speed, and rainfall, allowing the user to easily monitor the weather conditions.

4.5.1 Description

The LCD is constructed with a thin-film transistor (TFT) technology, which provides a high level of visual clarity and color accuracy. The display is comprised of two rows, each capable of displaying 16 characters, allowing for a comprehensive display of weather data. The LCD is encased in a compact plastic housing, measuring approximately 80x36x12mm, making it suitable for integration into a variety of weather monitoring applications. The display is illuminated by a white backlight, enabling clear visibility in lowlight environments.

The LCD is connected to the Node MCU ESP8266 microcontroller via a 4-pin interface, consisting of RS, EN, D4, and D5 pins, which facilitate communication between the microcontroller and the display. The LCD operates on a 5V power supply and requires a minimal current draw, making it suitable for battery-powered applications. The display is mounted on a breadboard-friendly PCB, allowing for effortless integration into the weather monitoring system.

4.5.2 Features

Firstly, the LCD has a high visual clarity, with a 16x2 character display that showcases vital weather data such as temperature, humidity, wind speed, and rainfall. The display is also equipped with a white backlight, allowing for clear visibility in low-light environments. Additionally, the LCD has a compact size, measuring approximately 80x36x12mm, making it suitable for integration into a variety of weather monitoring applications. The LCD is also lightweight and has a low power consumption, operating on a

5V power supply and requiring a minimal current draw, making it suitable for battery-powered applications.

Furthermore, the LCD has a fast response time, updating weather information in real-time, and has a wide viewing angle, ensuring that the display is visible from various positions. The LCD is also easy to interface with the Node MCU ESP8266 microcontroller, using a simple 4-pin interface, and is mounted on a breadboard-friendly PCB, making it easy to integrate into the weather monitoring system.

4.5.3 Types of LCD

There are several types of LCDs (Liquid Crystal Displays) available, each with its own unique characteristics and applications. Here are some of the most common types of LCDs:

1. TN (Twisted Nematic) LCD: This is the most common type of LCD, known for its low cost and wide viewing angles.

2. IPS (In-Plane Switching) LCD: This type of LCD offers better color accuracy and wider viewing angles than TN LCDs.

3. VA (Vertical Alignment) LCD: This type of LCD provides better contrast ratios and darker blacks than TN LCDs.

4. OLED (Organic Light-Emitting Diode) LCD: This type of LCD uses organic material to produce light, offering better contrast, wider viewing angles, and faster response times.

5. QLED (Quantum Dot Light-Emitting Diode) LCD: This type of LCD uses tiny crystals to create colors, offering better color accuracy and brightness.

6. Graphical LCD: This type of LCD can display graphics and images, often used in applications such as digital signage and gaming consoles.

7. Character LCD: This type of LCD can only display characters and numbers, often used in applications such as calculators and clocks.

8. Segment LCD: This type of LCD can display numeric and alphanumeric information, often used in applications such as digital clocks and timers.

9. Dot Matrix LCD: This type of LCD can display characters and graphics using a matrix of dots, often used in applications such as printers and plotters.

10. Passive LCD: This type of LCD does not have a built-in backlight, relying on external light sources to illuminate the display.

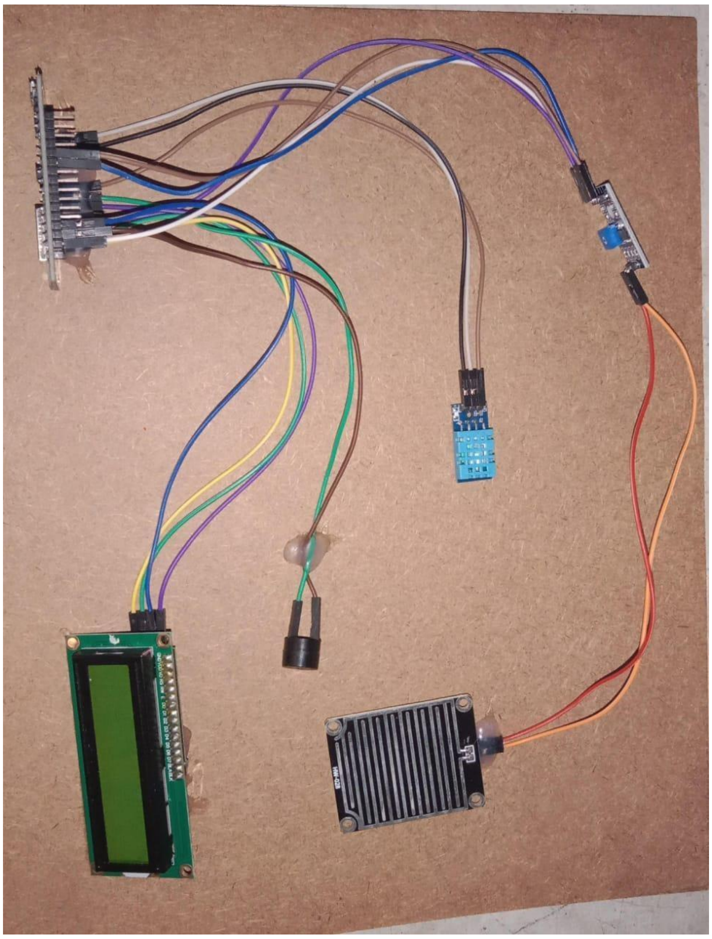
11. Active LCD: This type of LCD has a built-in backlight, allowing for brighter and more vibrant displays.

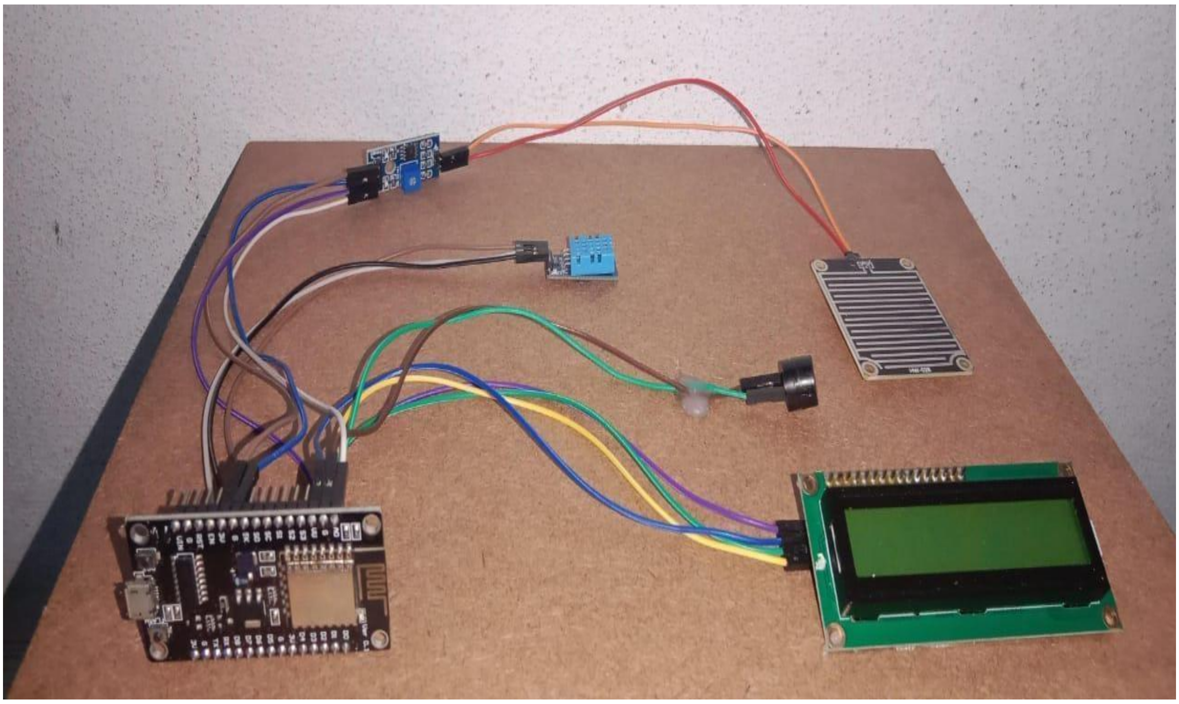
12. Transmissive LCD: This type of LCD uses a backlight to illuminate the display, often used in applications such as smartphones and TVs.

13. Reflective LCD: This type of LCD uses reflected light to illuminate the display, often used in applications such as digital watches and calculators.

14. Transflective LCD: This type of LCD uses both transmissive and reflective technologies to illuminate the display, often used in applications such as automotive displays.

Each type of LCD has its own strengths and weaknesses, and the choice of which to use depends on the specific application and requirements.





CHAPTER – 5

SOFTWARE IMPLEMENTATION

5.1 Arduino IDE

5.1.1 Introduction to Arduino IDE

IDE stands for Integrated Development Environment - An official software introduced by Arduino.cc that is mainly used for writing, compiling and uploading the code in almost all Arduino modules/boards. Arduino IDE is open-source software and is easily available to download & install from Arduino Official Site.

In this post, I'll take you through the brief Introduction of the Software, how you can install it, and make it ready for your required Arduino module.

Let's dive in and get down to the nitty-gritty of this Software.

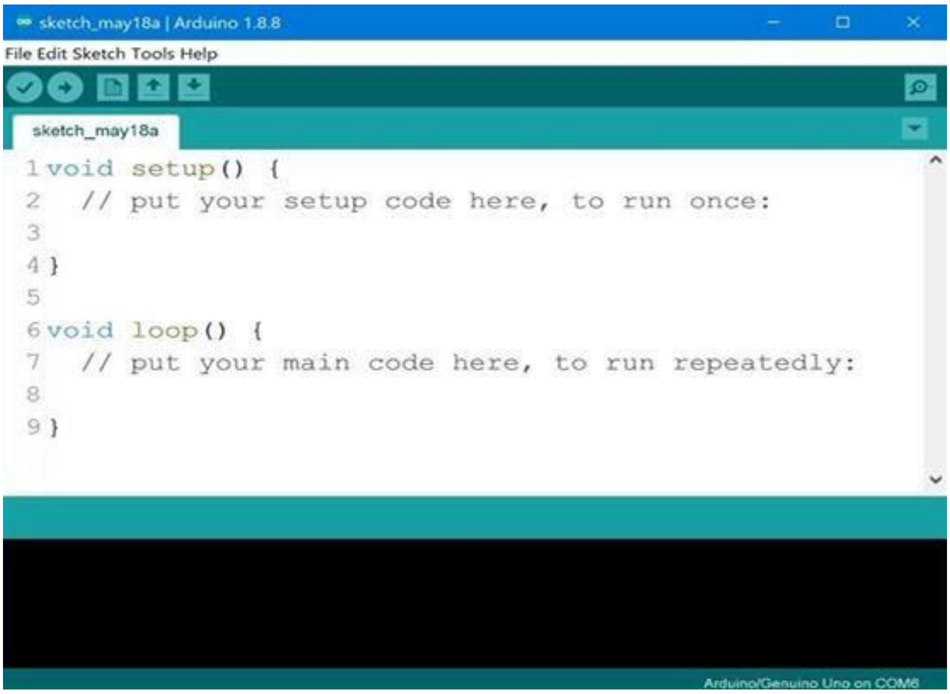


Fig 5.1: Arduino IDE Editor page

Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.

It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is available for all operating systems i.e., MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

This environment supports both C and C++ languages.

5.1.2 How to Download Arduino IDE

You can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.8.1 or Windows 10, as the app version is not compatible with Windows 7 or older version of this operating system.

You can download the latest version of Arduino IDE for Windows (Non admin standalone version), by clicking below button:

Arduino IDE Download

The IDE environment is mainly distributed into three sections.

1. Menu Bar

2. Text Editor

3. Output Pane

As you download and open the IDE software, it will appear like an image below:

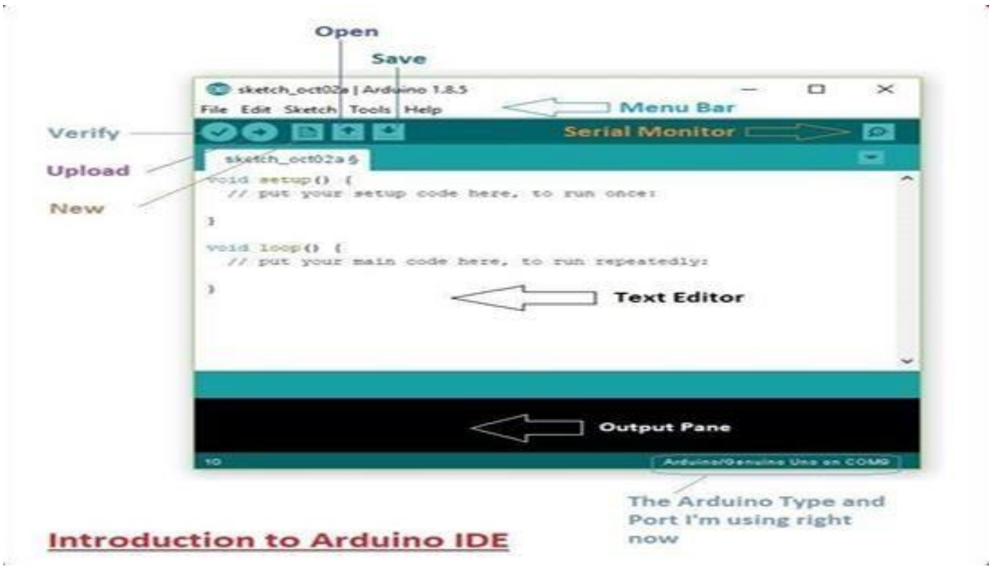


Fig 5.2: Introduction to Arduino IDE

The bar appearing on top is called Menu Bar that comes with five different options as File - You can open a new window for writing the code or open an existing one. The following table shows number of further subdivisions the file option is categorized into:

File	
New	This is used to open new text editor window to write your code
Open	Used for opening the existing written code
Open Recent	The option reserved for opening recently closed program
Sketchbook	It stores the list of codes you have written for your project
Examples	Default examples already stored in the IDE software
Close	Used for closing the main screen window of recent tab. If two tabs are open, it will ask you again as you aim to close the second tab
Save	It is used for saving the recent program
Save as	It will allow you to save the recent program in your desired folder
Page setup	Page setup is used for modifying the page with portrait and landscape options. Some default page options are already given from which you can select the page you intend to work on
Print	It is used for printing purpose and will send the command to the printer
Preferences	It is page with number of preferences you aim to setup for your text editor page
Quit	It will quit the whole software all at once

Fig 5.3: File subdivisions in Arduino IDE

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.

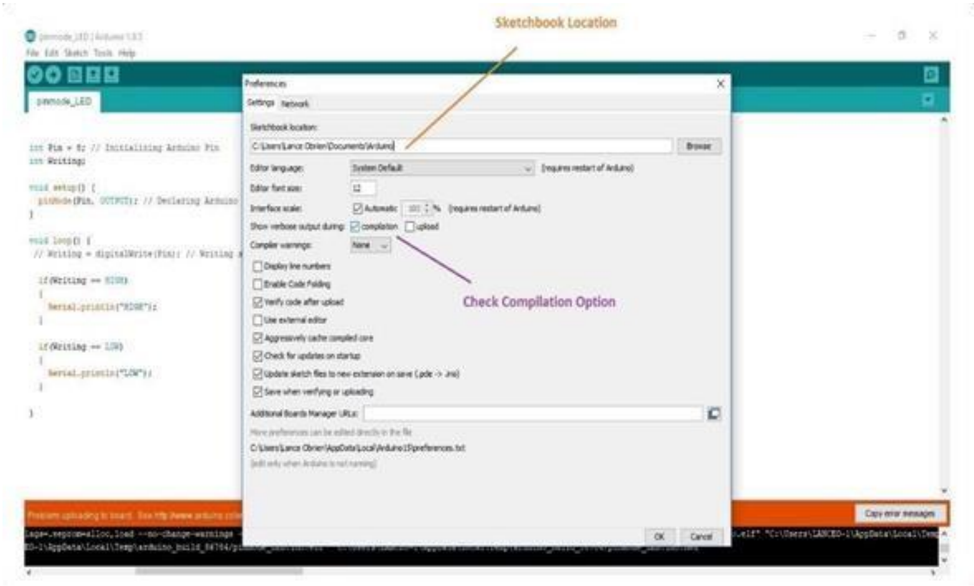


Fig 5.4: Selection of compilation

And at the end of the compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

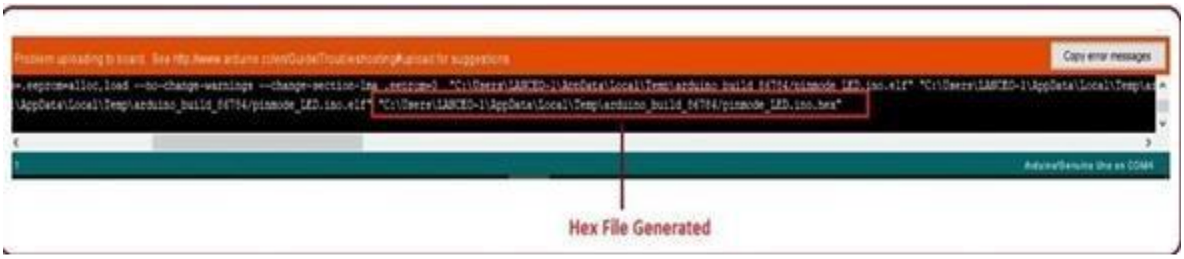


Fig 5.5: Hex file generation

Sketch - For compiling and programming.

• Tools - Mainly used for testing projects. The Programmer section in this panel is used for burning a boot loader to the new microcontroller.

• Help - In case you are feeling Edit - Used for copying and pasting the code with further modification for font.

• Sceptical about software, complete help is available from getting started to troubleshooting.

• The Six Buttons appearing under the Menu tab are connected with the running program as follows.

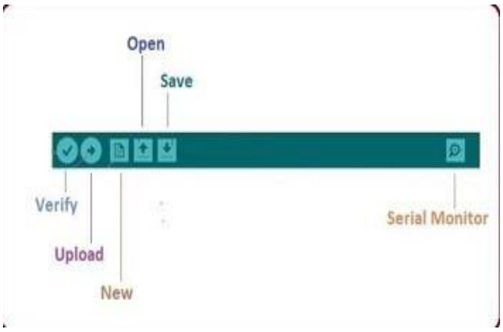


Fig 5.6: Serial monitor

- The check mark appearing in the circular button is used to verify the code. Click this once you have written your code.
- The arrow key will upload and transfer the required code to the Arduino board.
- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.
- The downward arrow is used to save the current running code.
- The button appearing on the top right corner is a Serial Monitor - A separate pop-up window that acts as an independent terminal and plays a vital role in sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
- You need to select the baud rate of the ESP8266 you are using right now. For my ESP8266 Baud Rate is 9600, Monitor, the output will show as the image below.

Fig 5.7: output of the serial monitor

The main screen below the Menu bard is known as a simple text editor used for writing the required code.

Fig 5.8: Text editor

Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors that occurred in the program. You need to fix the bottom of the main screen is described as those errors before you intend to upload the hex file into your Arduino Module.

Fig 5.9: output window

More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

5.1.3 Libraries

- Libraries are very useful for adding extra functionality into the ESP8266 Module.
- There is a list of libraries you can check by clicking the Sketch button in the menu bar and going to Include Library.

• As you click the Include Library and Add the respective library it will be on the top of the sketch with a #include sign. Suppose, I Include the Liquid Crystal library, it will appear on the text editor as

#include <LiquidCrystal_I2C.h>

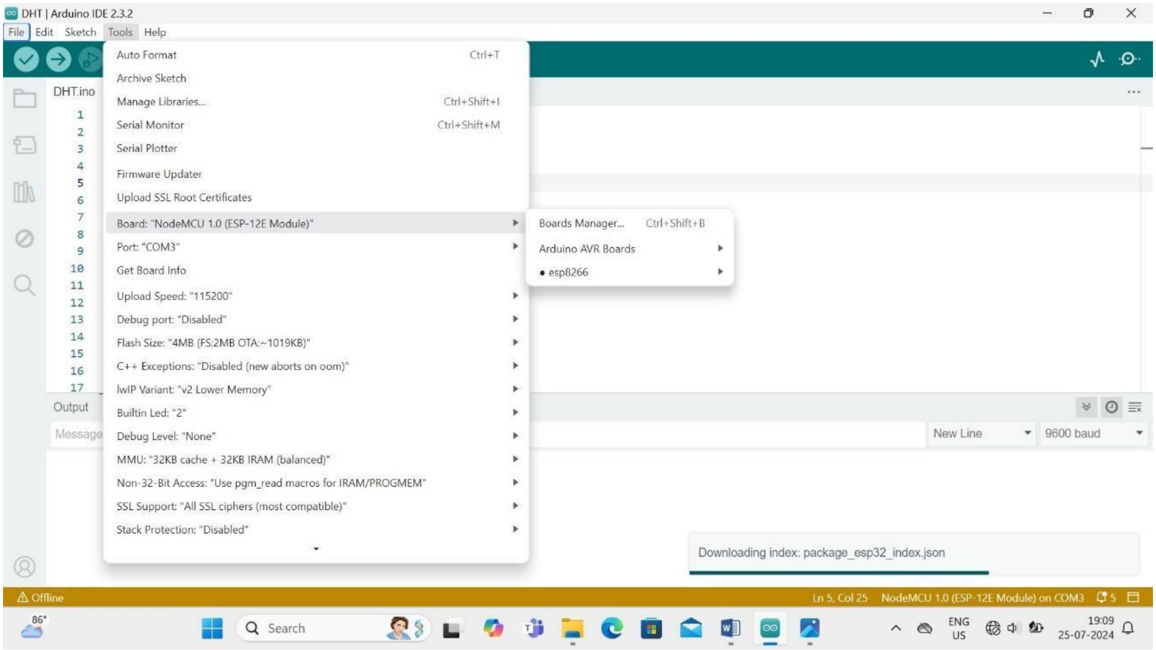


Fig 5.10: Selection of tools

• As you click the Include Library and Add the respective library it will be on the top of the sketch with a #include sign. Suppose, I Include the LiquidCrystal _I2C library, it will appear on the text editor as #include

• Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from external sources.

5.1.4 Making Pins Input or Output.

The digitalRead and digitalWrite commands are used for addressing and making the ESP8266 pins as an input and output respectively. These commands are text sensitive i.e., you need to write them down the exact way they are given like digitalWrite starting with small "d" and write with capital "W". Writing it down with DigitalWrite or digitalWrite won't be calling or addressing any function.

5.1.5 How to Select the Board

• In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system.

• As you click the Tools on the menu, it will open like the figure below:

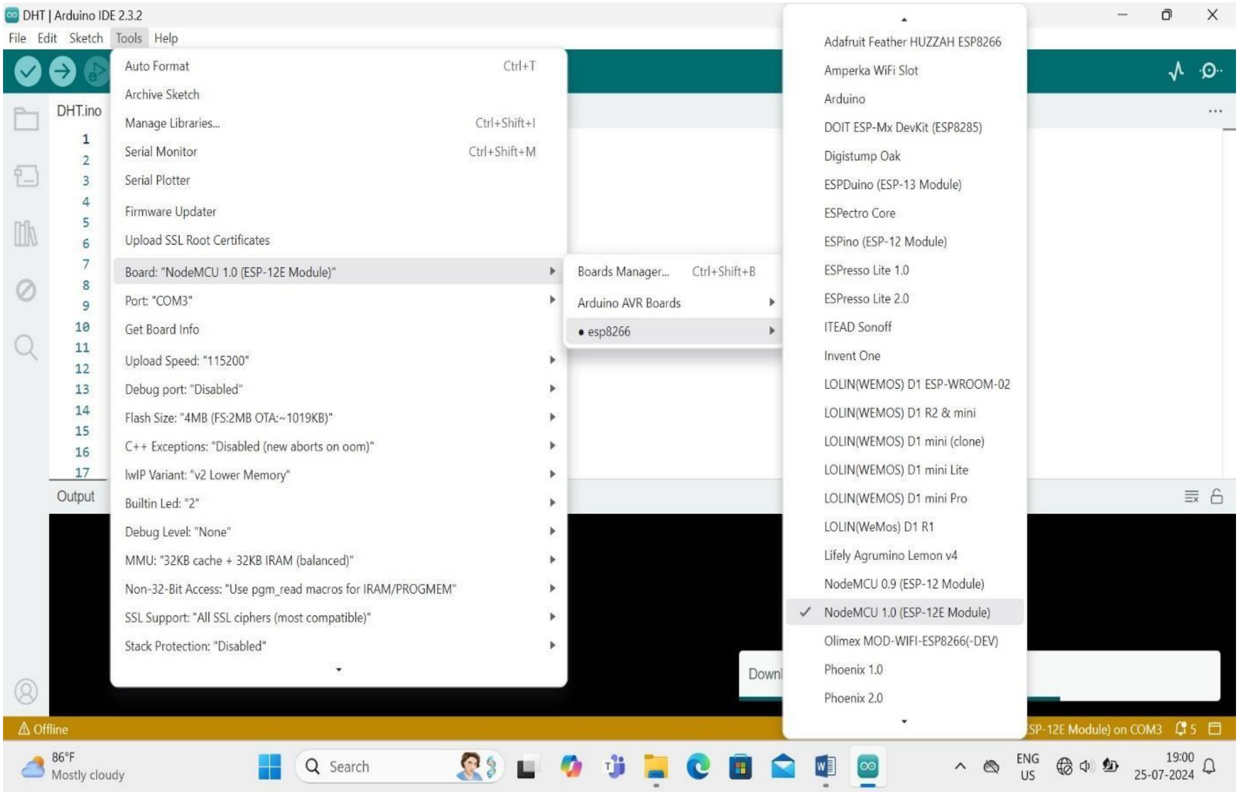


Fig 5.11: Selection of board manager

• Just go to the "Board" section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the port section of the Windows Device Manager.

• The following figure shows the COM3 that I have used for my project, indicating the Node MCU 1.0 with the COM3 port at the right bottom corner of the screen.

• After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six-button section or you can go to the Sketch section and press verify/compile and then upload.

• The sketch is written in the text editor and is then saved with the file extension into. It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, the older versions may require the physical reset on the board.

• Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

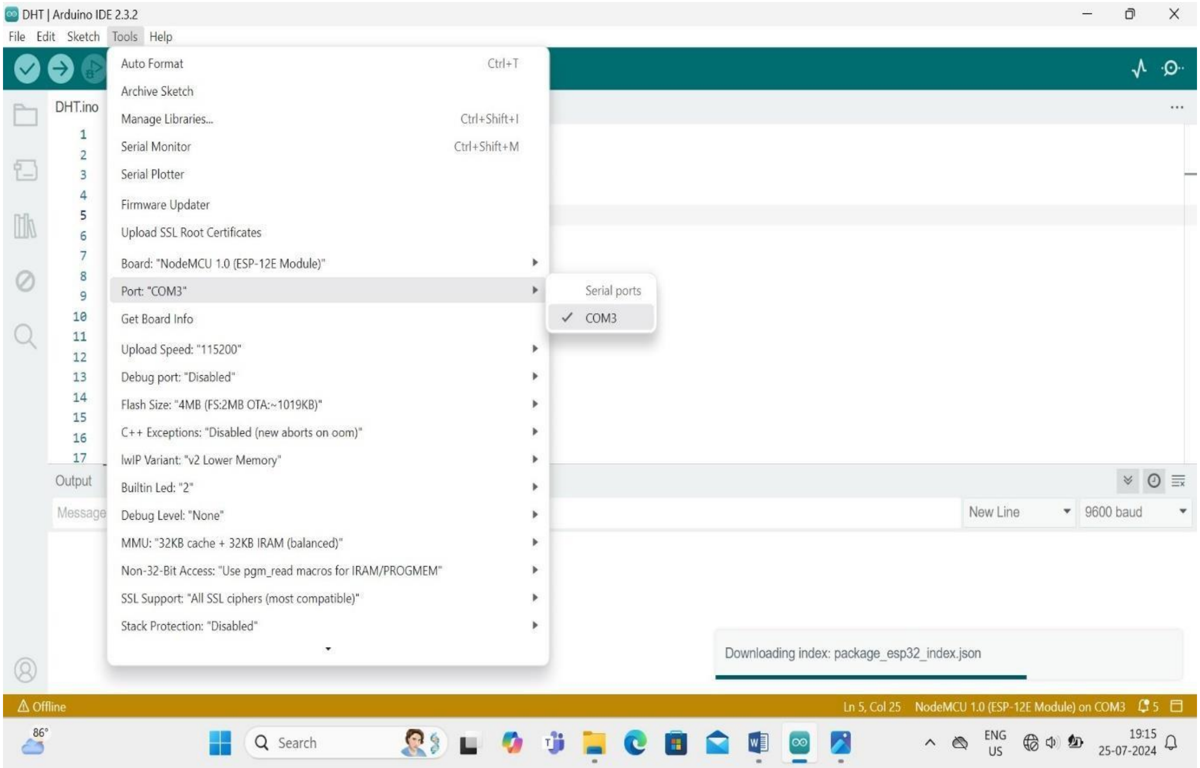


Fig 5.12: Selection of port

Note: The port selection criteria mentioned above are dedicated to Windows operating system only, you can check this Guide if you are using MAC or Linux.

The amazing thing about this software is that no prior arrangement or bulk of the mess is required to install this software, you will be writing your first program within 2 minutes after the installation of the IDE environment.

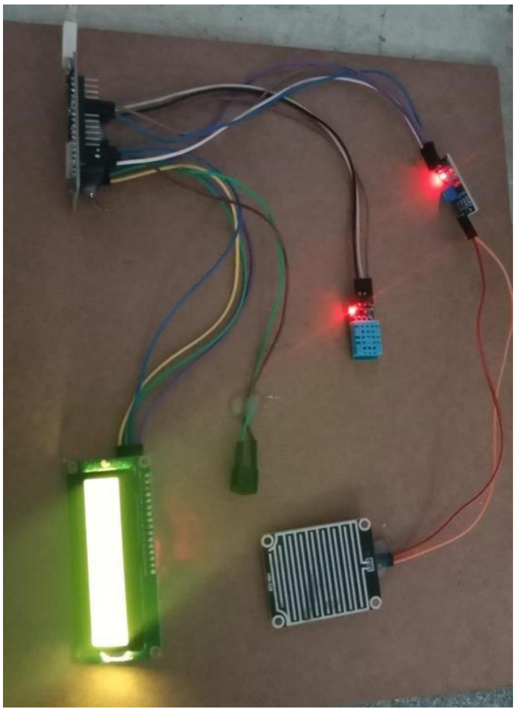
5.1.6 Uploading

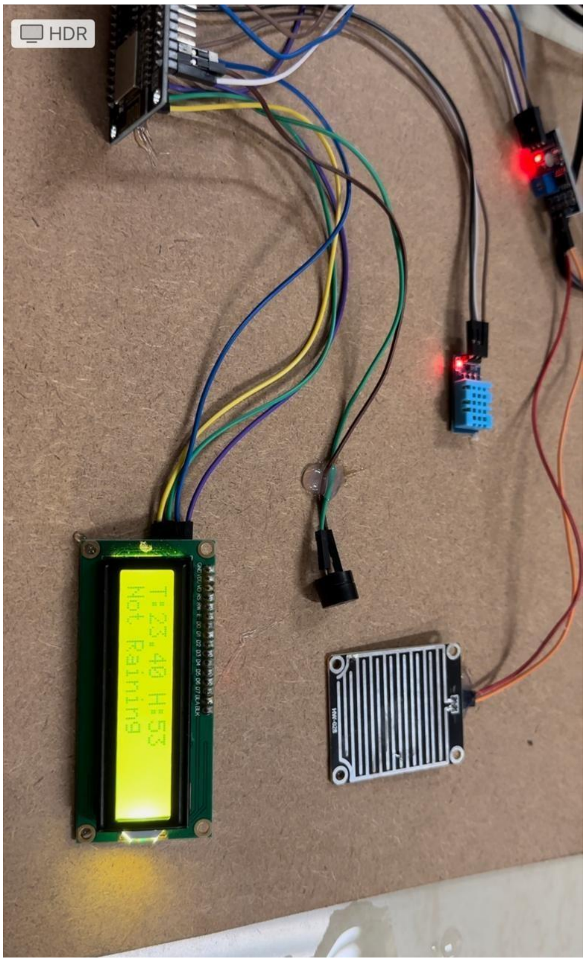
After writing your code, click on the upload button which is above the window and the code will be directly uploaded into the Node MCU with a cable wire connector.

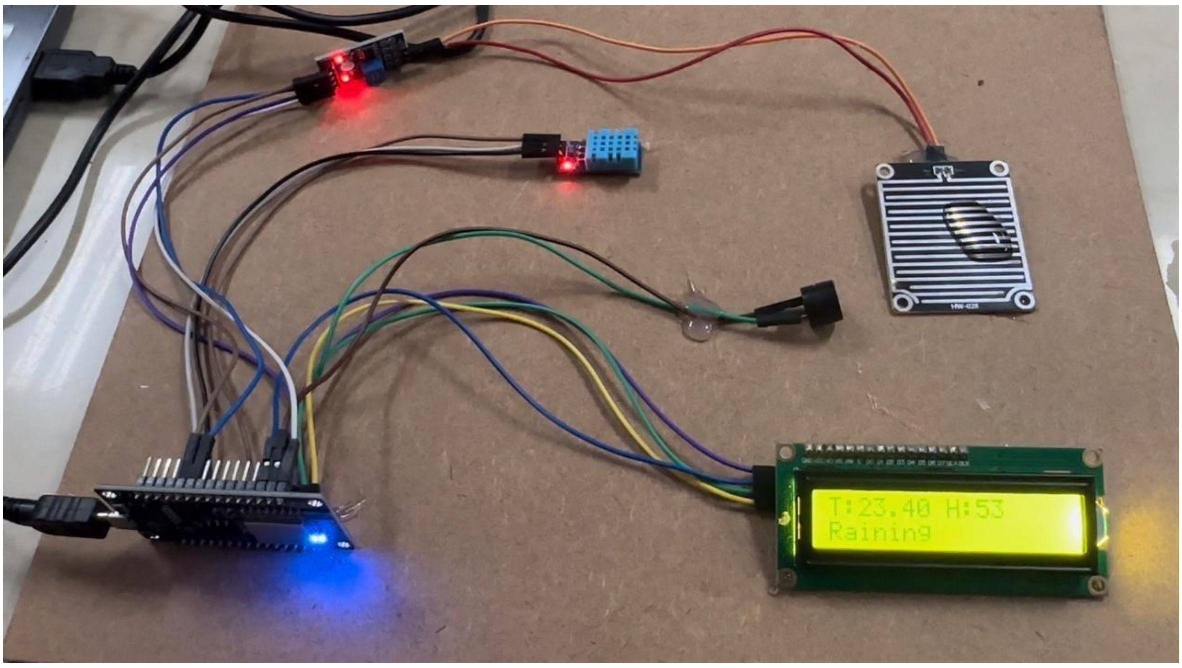
CHAPTER – 6

RESULT

The ‘DHT (Temperature and Humidity) sensor, the LDR sensor, and the rain sensor are all part of a control unit system circuit that the ESP8266 microcontroller controls. Then it is controlled by a USB link to transfer the coding portrayal to the ESP8266 microcontroller. The serial monitor displays the sensor data in the Arduino IDE software. The ESP8266 will associate with the Wi-Fi area of interest that has been relegated to this framework to develop a web server that will show all of the sensor information. Figure, which depicts the communication between the sensor station and the weather station over a Wi-Fi hotspot, will show the data received by the weather station on the OLED. Correspondence has been effectively settled. Figure shows the web server including html that might show sensor information utilizing fundamental code and association is expected to complete this capability. From that point onward, it will peruse all of the sensor esteems and impart them to the cloud through ThingSpeak. The sensor value will be saved and displayed on the channel you created by ThingSpeak. The climate boundary might be checked through ThingSpeak sites. Data was gathered from ESP8266 readings of all sensors and sent to ThingSpeak as a result of this projects objective.



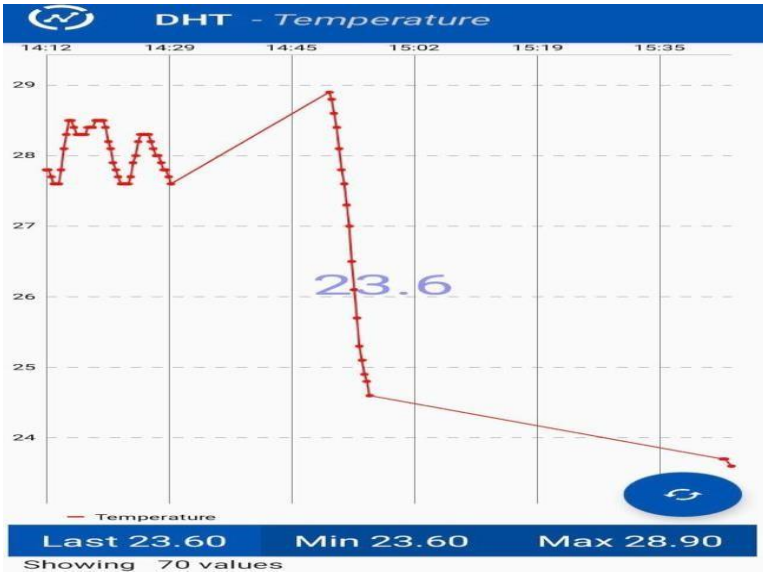




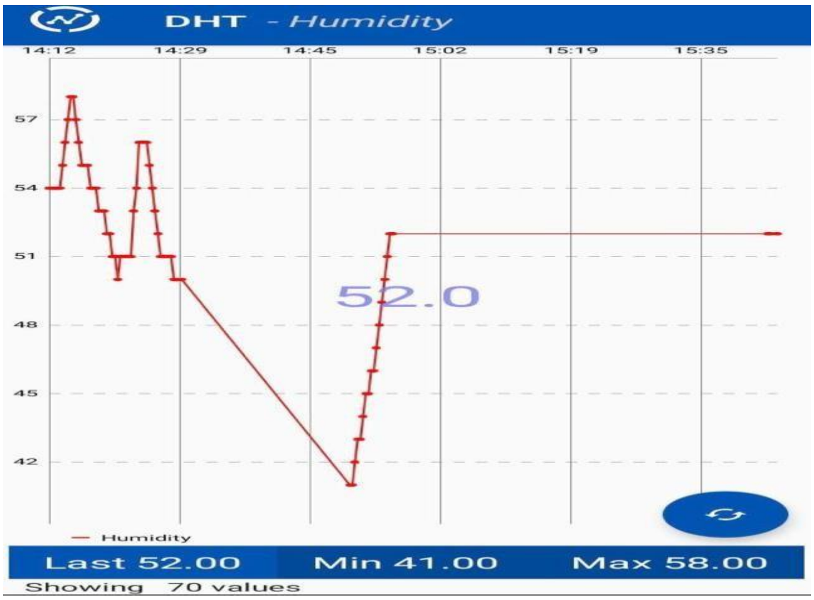
ThingSpeak display and data analysis:

Following testing with the web server, the ThingSpeak channel got similar sensor information as the web server however with a superior showcase of the diagram plot. The analysis of sensor data is made easier by this graph. ThingSpeak will plot analog data for each weather parameter on the graph to highlight its characteristics.

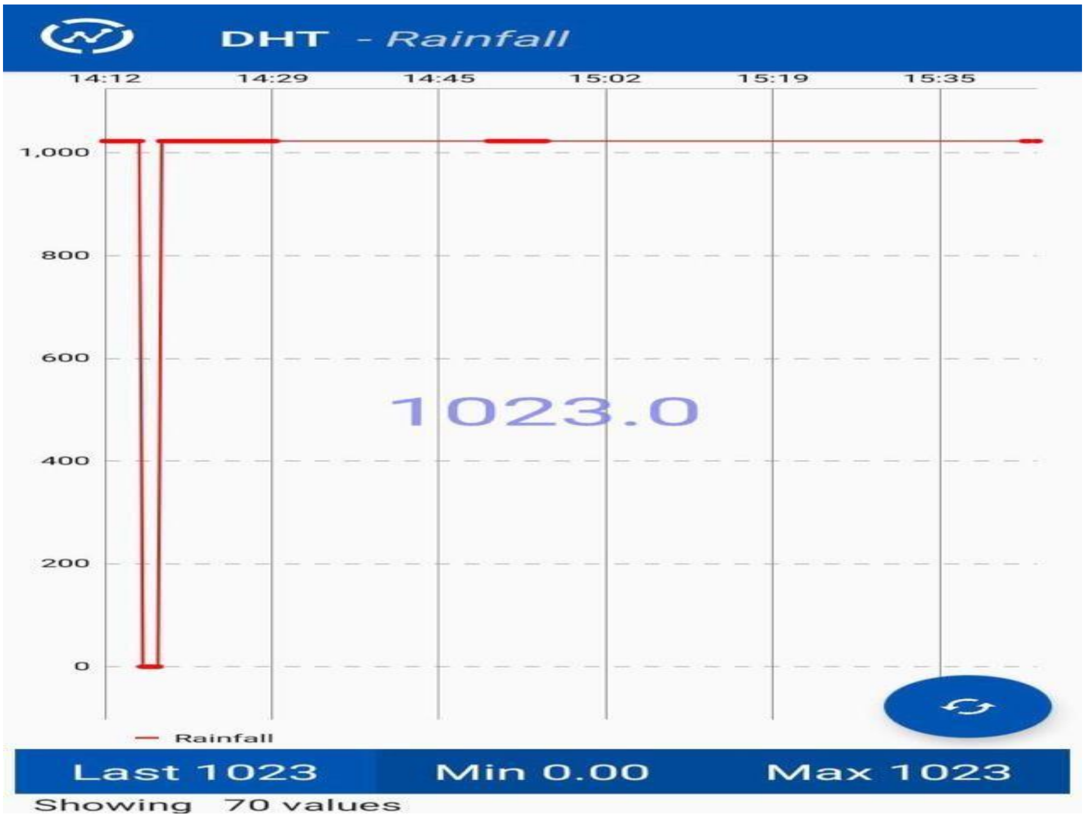
Temperature:



Humidity:



Rainfall:



CONCLUSION

Here's a conclusion for a weather monitoring system using DHT11 (not DHA), ESP8266, Rain Sensor, and LCD:

"In this project, we successfully designed and implemented a weather monitoring system using the ESP8266 microcontroller, DHT11 temperature and humidity sensor, rain sensor, and LCD display. The system accurately measures and displays temperature, humidity, and rainfall data in real-time.

The ESP8266 enables Wi-Fi connectivity, allowing for remote monitoring and data logging. The DHT11 sensor provides reliable temperature and humidity readings, while the rain sensor detects precipitation. The LCD display shows the current weather conditions and alerts users to changes in the weather.

This system has numerous applications, including:

1. Personal weather stations

2. Agricultural monitoring

3. Smart home automation

4. Environmental monitoring

The advantages of this system include:

1. Real-time weather data

2. Remote monitoring capabilities

3. Low cost and easy implementation

4. High accuracy and reliability

Overall, this project demonstrates the potential of IoT technology in weather monitoring and its applications in various fields.

Future Scope

1. IoT-based Weather Monitoring Networks: Establishing a network of weather monitoring systems for real-time monitoring and prediction of weather patterns.

2. Smart Agriculture: Using weather data to optimize crop management, irrigation, and harvesting.

3. Disaster Management: Utilizing weather monitoring systems for early warning systems and emergency response planning.

4. Environmental Monitoring: Expanding the system to monitor air and water quality, noise pollution, and other environmental parameters.

5. Commercial Applications: Developing weather monitoring systems for industries such as aviation, shipping, and construction.

By exploring these future enhancements and scope, we can further improve the system's capabilities and unlock new applications, ultimately contributing to a more sustainable and resilient future.

Future Enhancements:

1. Integrating additional sensors for wind speed, atmospheric pressure, and air quality monitoring

2. Developing a mobile app for remote monitoring and alerts

3. Implementing machine learning algorithms for weather forecasting

4. Expanding the system to monitor weather conditions over a larger geographical area 5. Integrating with other smart devices for automated decision-making

Source Code DHT

Code:

```
#include <LiquidCrystal_I2C.h>
#include<DHT.h>
LiquidCrystal_I2C lcd(0X27,16,2);
const int DHTPIN=D6; const int
DHTTYPE=11;
const int buzzer=D5; const int
rainSensorPin = D4;
DHT dht(DHTPIN,DHTTYPE);
void setup() {
  // put your setup code here, to run once:
lcd.init();
lcd.backlight(); dht.begin();
Serial.begin(115200);
pinMode(buzzer,OUTPUT); pinMode(rainSensorPin,
INPUT);
} void loop()
{
```

```
 // put your main code here, to run repeatedly:
float Temperature = dht.readTemperature();
int Humidity=dht.readHumidity(); int
sensorValue = analogRead(rainSensorPin); int
threshold = 500;
Serial.println("Temperature");
Serial.println("Humidity");
lcd.clear(); lcd.setCursor(0,0);
lcd.print("T:");
lcd.print(Temperature );
lcd.print(" H:");
lcd.print(Humidity);
if(sensorValue < threshold){
lcd.setCursor(0,1);
lcd.print("Raining");
digitalWrite(buzzer,HIGH);
delay(1000);
digitalWrite(buzzer,LOW);
delay(1000); }else{
lcd.setCursor(0,1);
lcd.print("Not Raining");
}
if(Temperature>=35||Humidity>=60){
digitalWrite(buzzer,HIGH);
delay(500);
digitalWrite(buzzer,LOW);
delay(1000); }else{
```

```
digitalWrite(buzzer,LOW);
delay(1000); } delay(5000);
}
```

ThingSpeak Code:

```
#define TS_ENABLE_SSL // For HTTPS SSL connection #include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include "secrets.h"
#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros char ssid[] = "PAVANI_DS";   // your network SSID (name)  char pass[] = "PAVANIM9";  // your network password int keyIndex = 0;          // your network key Index number (needed only for WEP) WiFiClientSecure  client; unsigned long myChannelNumber =  2600442; const char * myWriteAPIKey ="P1BHG4NI1TS0RQOB";
float Temperature; int
Humidity; int sensorValue; int
threshold; // Initialize our
values int number1 = 0; int
number2 = random(0,100); int
number3 = random(0,100); int
number4 = random(0,100);
String myStatus = "";
// Fingerprint check, make sure that the certificate has not expired.
const char * fingerprint = NULL; // use SECRET_SHA1_FINGERPRINT for fingerprint check
#include<DHT.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0X27,16,2); const
int DHTPIN=D6;
const int DHTTYPE=11;
```

```
const int buzzer=D5; const int
rainSensorPin = D4;
DHT dht(DHTPIN,DHTTYPE);
void setup() { Serial.begin(115200);
lcd.init(); lcd.backlight();
dht.begin();
pinMode(buzzer,OUTPUT);
pinMode(rainSensorPin, INPUT);  // Initialize serial   while
(!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo native USB port only
  }
  WiFi.mode(WIFI_STA);
if(fingerprint!=NULL){
    client.setFingerprint(fingerprint);
  }   else{     client.setInsecure(); // To perform a simple SSL
Encryption
  }
  ThingSpeak.begin(client);  // Initialize ThingSpeak
} void loop()
{
  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");     Serial.println(SECRET_SSID);
while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open
or WEP network
      Serial.print(".");
delay(5000);
```

```
    }
    Serial.println("\nConnected.");
  }
float Temperature = dht.readTemperature();
int Humidity=dht.readHumidity(); int
sensorValue = analogRead(rainSensorPin); int
threshold = 500;
Serial.println("Temperature");
Serial.println("Humidity");
lcd.clear(); lcd.setCursor(0,0);
lcd.print("T:");
lcd.print(Temperature );
lcd.print(" H:");
lcd.print(Humidity);
if(sensorValue < threshold){
lcd.setCursor(0,1);
lcd.print("Raining");
digitalWrite(buzzer,HIGH);
delay(1000);
digitalWrite(buzzer,LOW);
delay(1000); }else{
lcd.setCursor(0,1);
lcd.print("Not Raining");
}
if(Temperature>=35||Humidity>=60)
{
digitalWrite(buzzer,HIGH);
```

```
delay(500);
digitalWrite(buzzer,LOW);
delay(1000); } else{
digitalWrite(buzzer,LOW);
delay(1000);  } delay(5000);
 // set the fields with the values
 ThingSpeak.setField(1, Temperature);
 ThingSpeak.setField(2, Humidity);
 ThingSpeak.setField(3, sensorValue);
 ThingSpeak.setField(4,    number4);
// figure out the status message
 if(number1 > number2){
   myStatus = String("field1 is greater than field2");
 }
 else if(number1 < number2){
   myStatus = String("field1 is less than field2");
 }   else{     myStatus = String("field1
equals field2");
 }
 // set the status
 ThingSpeak.setStatus(myStatus);

 //   write   to   the   ThingSpeak   channel        int   x   =
ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
 if(x == 200){
   Serial.println("Channel update successful.");
 }
```

```
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
  // change the values
  number1++;
if(number1 > 99){
number1 = 0;
  }
  number2 = random(0,100);
number3 = random(0,100);
number4 = random(0,100);
  // Wait 20 seconds to update the channel again
}
```

REFERENCES

1. Math, R.K.M., & Dharwadkar, N.V. IoT Based Low-cost Weather Station and Monitoring System for Precision Agriculture in India. in the Proceedings of 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India., pp. 1-2, (2018) 9 E3S Web of Conferences 391, 01142 (2023) https://doi.org/10.1051/e3sconf/202339101142 ICMED-ICMPC 2023 2. Varghese, L., Deepak, G., & Santhanavijayan, A. An IoT Analytics Approach for Weather Forecasting using Raspberry Pi 3 Model B+. in the Proceedings of 2019 Fifteenth International Conference on Information Processing (ICINPRO). Kerala, India, pp. 1-5, (2019) 3. Dhawan, A., & Kumar, A. An Approach for Implementing Innovative Weather Monitoring System with DHT11 Sensor and Arduino Uno Tool based on IoT, in the Proceedings of International Conference on Computational Science and Computing, New Delhi, India, January 22-24, 2021.pp. 1-2, (2021) 4. Dabhi, A., & Jethva, D. Design and Development of E-Sense: IoT-based Environment Monitoring System. in the Proceedings of 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) Kharagpur, India.,pp. 1-5, (2020) 5. T. Sai, Y.-T., Kuo, Y.-W., Hwang, C., & Tsao, Y.-C. An implementation of IoT-based weather monitoring system. in the Proceedings of 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), Fukuoka, Japan pp. 12, (2019).