ISSN: **3107-6513**

Solar Powered Wi-Fi Weather Station using Node MCU and Thing Speak

Pandimadevi Ganesan[1], Anas Mohammed Mansoor Al-Anqoudi[2], Maher Masoud Khazam Alhinai[3]
[,2,31]*(College of Engineering and Technology/ University of Technology and Applied Sciences ,Oman)*

Abstract
The increasing need for continuous and reliable weather monitoring has highlighted the importance of developing autonomous, energy-efficient environmental monitoring systems. Conventional weather stations often depend on external power sources and may be unsuitable for remote and off-grid locations. Therefore, this project presents the design and development of a solar-powered Wi-Fi weather station capable of monitoring temperature, humidity, and atmospheric pressure in real time. The proposed system utilizes an IoT-based monitoring platform powered by a solar energy harvesting system. A solar panel, TP4056 charge controller, and lithium-ion battery provide a sustainable power supply for continuous operation. The DHT11 sensor measures temperature and humidity, while the BMP280 sensor monitors atmospheric pressure. A NodeMCU (ESP32) microcontroller processes sensor data and transmits the collected information wirelessly to the ThingSpeak cloud platform through its built-in Wi-Fi capability. Furthermore, the system incorporates deep sleep mode to reduce power consumption and extend battery operating time between charging cycles.The weather station is designed for outdoor use and is protected by a weatherproof enclosure. The collected data can be viewed remotely through the ThingSpeak cloud platform, allowing users to monitor weather conditions from anywhere. The system is suitable for agricultural fields, smart farms, homes, and remote areas.The proposed solar-powered Wi-Fi weather station provides a simple, reliable, and cost-effective solution for real-time weather monitoring. By using solar energy, Wi-Fi communication, and cloud-based data storage, the system reduces dependence on conventional power sources and supports smart agriculture and environmental monitoring.

Keywords: Solar Energy, Wi-Fi Weather Station, Internet of Things (IoT), NodeMCU ESP32, DHT11 Sensor, Renewable Energy, Environmental Monitoring.

I.INTRODUCTION

Weather monitoring is important in agriculture, environmental protection, climate studies, and daily life. Parameters such as temperature, humidity, atmospheric pressure, rainfall, and air quality provide useful information about weather conditions. Monitoring these parameters helps people understand environmental changes and make better decisions.Traditional weather monitoring systems may be expensive to install and maintain. They may also require a continuous electricity supply, which makes their use difficult in remote areas. Manual data collection takes time and does not always provide real-time information.Recent developments in microcontrollers, sensors, renewable energy, and Internet of Things (IoT) technology have made it possible to develop low-cost weather monitoring systems. Several existing systems use microcontrollers such as Arduino, NodeMCU, and ESP32 to collect environmental data. IoT technology allows the collected data to be sent to cloud platforms for remote monitoring. Solar energy can also be used to supply power to these systems, especially in areas where electricity is not easily available. However, combining sensor monitoring, wireless communication, local display, and solar power in a single system is still an important design consideration.

This work focuses on the design and development of a Solar-Powered Weather Monitoring System using an ESP32 microcontroller and IoT technology. The system uses different sensors to measure temperature, humidity, atmospheric pressure, rainfall, and air quality, based on the sensors included in the final design. The ESP32 collects and processes the sensor data. The readings are displayed on an LCD screen and sent to an IoT cloud platform through Wi-Fi for remote monitoring. A solar panel and rechargeable battery are used to provide power to the system.The main aim of this project is to develop a low-cost and energy-efficient weather monitoring system that can be used in agricultural fields, residential areas, and remote locations. The main contribution is the integration of environmental sensors, ESP32-based data processing, LCD display, IoT-based remote monitoring, and solar power supply into a single system. The developed system is tested to check sensor operation, data transmission, and overall performance.

1.2 Problem Statement

Weather information is important for agriculture, environmental monitoring, and daily decision-making. However, traditional weather monitoring systems can be expensive and may require complex equipment and a continuous electricity supply. These requirements make them difficult to use in small-scale applications and remote areas.In addition, some systems do not provide easy access to real-

ISSN: 3107-6513

time weather information from distant locations. Manual data collection also requires time and effort.Therefore, there is a need for a low-cost and energy-efficient weather monitoring system that can automatically collect environmental data and provide both local and remote access. This project aims to address this need by developing a solar-powered weather monitoring system using an ESP32 microcontroller, environmental sensors, and IoT technology.

1.3 Goal and Objectives

The main goal of this project is to design and develop a solar-powered weather monitoring system using an ESP32 microcontroller for real-time data collection and remote monitoring.

The specific objectives are:

To develop a weather monitoring system powered by solar energy.

To measure temperature, humidity, atmospheric pressure, rainfall, and air quality using suitable sensors.

To collect and process sensor data using the ESP32 microcontroller.

To display the measured values on an LCD screen.

To transmit weather data to an IoT cloud platform through Wi-Fi.

To reduce dependence on conventional electricity using a solar panel and rechargeable battery.

To assemble and test the system and evaluate its performance.

1.4 Significance of the Study

The proposed Solar-Powered Weather Monitoring System combines solar energy and IoT technology to monitor environmental conditions. Its main benefits are:

Energy efficiency: Solar energy is used to power the system and reduce dependence on conventional electricity.

Real-time monitoring: The system automatically collects weather data and helps users observe environmental changes.

Remote accessibility: Users can view the collected data through an IoT cloud platform using an internet connection.

Agricultural applications:Weather information helps farmers understand local conditions and plan their farming activities.

- Cost-effectiveness: The use of an ESP32 and low-cost sensors makes the system suitable for small-scale applications.

- Educational value: The project provides practical knowledge of sensors, microcontrollers, solar energy, and IoT technology.

- Environmental monitoring: The system helps collect and observe environmental data from different locations.

1.5 Scope and Limitations

This project covers the design, development, and testing of a solar-powered weather monitoring system using an ESP32 microcontroller. The system collects environmental data and provides local and remote monitoring. It measures temperature and humidity using a DHT11 sensor, atmospheric pressure using a suitable pressure sensor, and rainfall using a rain sensor. Air quality and light intensity can also be monitored if the corresponding sensors are included in the final system. The measured values are displayed on an LCD using I2C communication and transmitted to an IoT cloud platform through Wi-Fi. A solar panel and rechargeable battery are used to supply power to the system. The project also includes testing sensor readings, wireless communication, and overall system operation. The system is suitable for agricultural fields, residential areas, educational institutions, and environmental monitoring applications. However, its performance may be affected by several limitations. Solar power generation depends on sunlight, and long periods of cloudy weather may reduce battery charging. Remote monitoring requires a stable Wi-Fi connection and access to the IoT cloud platform. Sensor accuracy depends on the quality and calibration of the sensors, while dust, moisture, and high temperatures may affect their performance and the electronic components. The operating time also depends on the solar panel capacity, battery capacity, and total power consumption. Furthermore, the system is designed to monitor current weather conditions and does not provide professional weather forecasting. Regular inspection and maintenance of the sensors, solar panel, battery, and protective enclosure are also required to ensure reliable operation.

1.6 Organization of the Paper

The remaining sections of this paper are organized as follows. Section 2 reviews previous research related to IoT-based weather monitoring systems, environmental sensors, ESP32 microcontrollers, and solar-powered systems. Section 3 explains the proposed system design and methodology, including the system architecture, hardware components, sensor connections, power supply, and software development. Section 4 describes the system implementation and experimental setup, including hardware assembly, programming, sensor interfacing, IoT cloud connection, and testing procedures. Section 5 presents and discusses experimental results, including sensor readings, LCD display, cloud monitoring, and overall system performance. Finally, Section 6 concludes the project by summarizing the main findings, discussing the limitations, and suggesting possible improvements for future development.

II. LITERATURE REVIEW

Several studies were conducted in literature to examine (FIV) for non-intrusive flow measurement. These studies use various sensors, including accelerometers, laser Doppler vibrometers, Distributed Acoustic Sensing, piezoelectric and other devices, to capture vibration and fluctuation produced by flow in pipelines. By analysing these signals, researchers have identified clear relationships between flow rate and vibration. Each study examines different factors such as pipe diameter, material, geometry, fluid properties and flow velocity, offering insights into how FIV can accurately monitor water flow rates.

Microcontroller-Based Weather Monitoring System

ISSN: 3107-6513

Kumar et al. (2019) described a microcontroller-based weather monitoring system using an Arduino Uno and a DHT11 sensor to measure temperature and humidity. The measured values were displayed on an LCD, allowing users to observe the environmental conditions locally.

This approach demonstrates the usefulness of microcontrollers in automating basic weather measurements. However, a system that relies only on a local display does not provide convenient remote access to the collected information. Wireless communication and cloud-based data storage could improve its usefulness by allowing users to monitor readings from different locations.

IoT-Based Weather Monitoring System

Patel et al. (2020) presented an IoT-based weather monitoring approach using a NodeMCU ESP32, DHT11 sensor, and BMP280 pressure sensor. The system used Wi-Fi communication to transmit environmental measurements to the ThingSpeak cloud platform.

The use of a wireless microcontroller and an IoT platform supports remote monitoring and online visualization of weather data. Users can access recorded measurements without being physically present near the monitoring unit. However, the suitability of such a system for locations without a dependable electrical supply depends on its power arrangement. Integrating a solar panel and rechargeable battery can help address this requirement.

Solar-Powered IoT Weather Monitoring System

Sharma et al. (2021) described a solar-powered IoT weather monitoring system that combined a solar panel and rechargeable battery with environmental sensors. The system was intended to monitor parameters such as temperature, humidity, and rainfall, with data made accessible through cloud-based or mobile applications.

This approach is relevant to agricultural and remote applications because solar energy can reduce dependence on grid electricity. The combination of renewable energy and wireless data communication offers a practical direction for developing autonomous environmental monitoring equipment. Nevertheless, the system's operating duration depends on factors such as solar energy availability, battery capacity, sensor consumption, and the power requirements of the controller and communication module.

Rainfall Monitoring System

Rao et al. (2018) described a rainfall monitoring system using an Arduino controller and a rain sensor. The system detected rainfall conditions and presented information through an LCD or an alert mechanism.

Rainfall monitoring can provide useful information for agricultural planning and local environmental observation. Integrating a rain sensor with temperature, humidity, and pressure sensors can provide a broader picture of local weather conditions than monitoring rainfall alone. For the proposed project, rainfall measurement can be incorporated into the ESP32-based system if the selected sensor and its interface are included in the final hardware design.

Smart Weather Monitoring Using Multiple Sensors

Singh et al. (2022) described a weather monitoring approach using a NodeMCU controller and multiple environmental sensors, including sensors for temperature, humidity, light, and rainfall. The collected information was transmitted to a cloud platform, where it could be presented through graphs or a dashboard for access using a smartphone or computer.

The use of multiple sensors increases the range of environmental information available to users. Cloud-based visualization also makes it easier to observe changes in measurements over time. However, the number of monitored parameters depends on the sensors installed, the controller's interface configuration, and the software developed for the system.

Wireless Environmental Monitoring Using Cloud Computing

Gupta et al. (2021) described a wireless environmental monitoring approach using an ESP32-based controller, cloud data storage, and a web dashboard. Such systems demonstrate how wireless communication can connect field-level sensors with online data management platforms.

Cloud integration can support remote access, data recording, and visualization of environmental measurements. These capabilities are useful when monitoring equipment is installed in locations that are difficult to access regularly. However, remote monitoring depends on the availability of a suitable wireless network and an operational cloud service. Local data display can remain useful when remote access is temporarily unavailable. **IoT Weather Monitoring with Data Analytics**

Joshi et al. (2023) described an IoT-based weather monitoring approach involving cloud data collection and analytical methods for examining weather patterns. The use of stored measurements and data analytics can help users identify trends and changes in environmental conditions.

This approach indicates the potential for extending basic weather monitoring systems with additional analytical features. However, collecting and displaying sensor measurements is different from providing reliable weather forecasts. Forecasting requires appropriate historical data, validated analytical methods, and performance evaluation. The primary focus of the proposed project is environmental data collection and remote monitoring rather than professional-grade weather prediction.

Problem Identification

Despite the availability of modern weather monitoring systems, many existing solutions are still expensive and require continuous power supply and complex infrastructure. These systems may not be suitable for small-scale applications or remote locations. Many rural and agricultural areas lack access to reliable electricity and advanced monitoring systems. As a result, farmers often rely on manual observations, which may not provide accurate or timely weather information. To address these challenges, a Solar Powered Weather Monitoring System using NodeMCU is proposed. The system integrates environmental sensors with an IoT-enabled microcontroller to collect and transmit real-time weather data.

ISSN: 3107-6513

The proposed system offers the following advantages:

- Low-cost implementation using easily available components
- Solar-powered operation suitable for remote areas
- Real-time monitoring using IoT cloud platforms
- Wireless data transmission using Wi-Fi
- Easy access to weather data through mobile or web applications

By combining solar energy and IoT technology, the proposed system provides an efficient, sustainable, and affordable solution for weather monitoring applications

III. METHODOLOGY

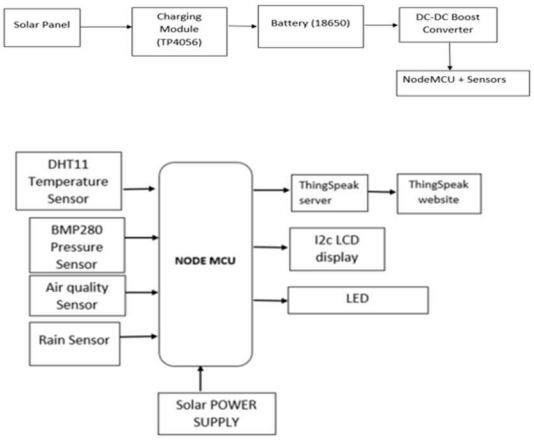

Figure 3.1: Overall System Architecture

The given diagram (figure 3.1) illustrates a solar-powered IoT-based weather monitoring system designed to operate independently using renewable energy. The system begins with a solar panel that captures sunlight and converts it into electrical energy. This energy is managed by a TP4056 charging module, which ensures safe and efficient charging of a rechargeable 18650 lithium-ion battery by protecting it from overcharging and over-discharging. The stored energy in the battery is then supplied to a DC-DC boost converter, which steps up the voltage to a stable level (typically 5V) required for proper operation of the NodeMCU and associated components. This ensures continuous functioning of the system even during low sunlight conditions or at night.At the core of the system is the NodeMCU microcontroller, which acts as the brain and control unit. It receives input data from multiple environmental sensors connected to it. The DHT11 sensor measures temperature and humidity, providing basic atmospheric conditions. The BMP280 sensor measures atmospheric pressure and can also estimate altitude, making it useful for weather prediction. The air quality sensor detects the presence of harmful gases or pollutants in the air, helping to monitor environmental pollution levels. The rain sensor identifies the presence of rainfall by detecting water droplets on its surface. All these sensors continuously send real-time data to the NodeMCU for processing. Once the data is collected and processed, the NodeMCU performs multiple output functions. It displays the real-time sensor readings on an I2C LCD display, allowing users nearby to monitor the conditions instantly. An LED is also included in the system to provide visual indications, such as showing system status or alerting when certain conditions (like rain detection) occur. In addition to local display, the NodeMCU uses its built-in Wi-Fi capability to transmit the sensor data to the ThingSpeak server, an IoT cloud platform. This data is then stored and visualized on the ThingSpeak website, where users can access it remotely in the form of graphs, charts, and analytics over time. Overall, this system integrates renewable energy, embedded systems, and IoT technology to create an efficient and autonomous weather monitoring solution. It is highly useful in applications such as smart agriculture, environmental monitoring, and remote weather stations, where continuous data collection and remote accessibility are essential. The use of solar power makes the system eco-friendly, cost-effective, and suitable for deployment in remote or off-grid locations.

IV. RESULTS AND DISCUSSION

This section discusses the complete implementation and working process of the proposed project in a detailed manner. Initially, the testing and verification of individual hardware and software components used in the system are presented to ensure proper functionality and reliability. After confirming the successful operation of the individual modules, the components were assembled and integrated to form the complete system.

NodeMCU ESP32

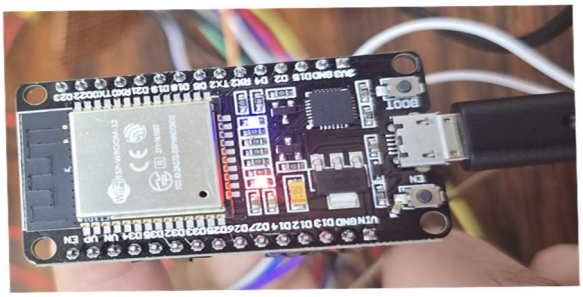

Figure Node MCU ESP32 Testing

Figure illustrates the procedure followed to test the NodeMCU-32. Initially, the ESP32 board was connected to a laptop using a USB cable. After establishing the connection, the Arduino IDE software was opened, and the Blink example program was selected from File → Examples → ESP32 → Blink. The program was then verified to check for errors and uploaded to the ESP32 board successfully. Once the uploading process was completed, the built-in LED available on the ESP32 board started blinking continuously, confirming the proper functioning of the board and successful program execution

LCD I2C 16x2

Figure shows the interfacing of the I2C LCD display with the NodeMCU-32. The LCD connections are made as follows: Vin to 5V, GND to GND, SDA to D2, and SCL to D1 of the ESP32 board. After completing the circuit connections, the ESP32 is connected to a laptop using a USB cable.

ISSN: **3107-6513**

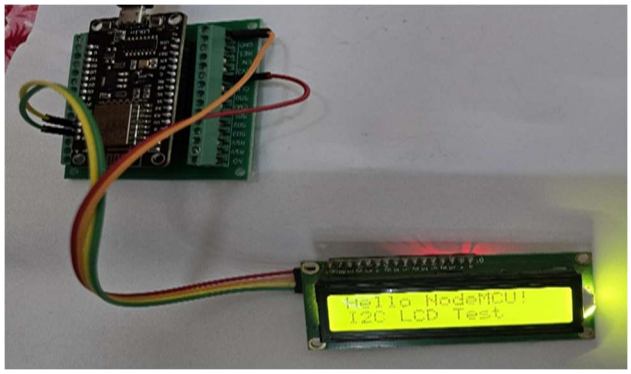

Figure 6.2 LCD I2C Testing

The Arduino IDE software is then opened, and the example program is selected through File → Examples → LiquidCrystal_I2C → HelloWorld. After verifying the program successfully, the code is uploaded to the ESP32 board. Once the uploading process is completed, the LCD screen displays the messages "Hello NODEMCU" and "I2C LCD TEST," confirming the successful interfacing and operation of the LCD module.

DHT11 Sensor

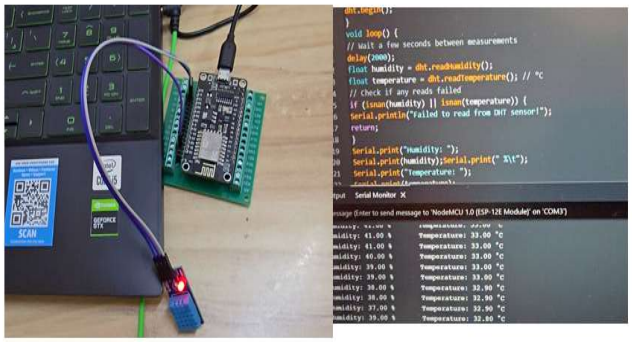

Figure DHT11 Sensor Testing and output

Figure 6.3 shows the interfacing of the DHT11 temperature and humidity sensor with the NodeMCU-32. The sensor connections are made as follows: the VCC pin is connected to 5V, the data pin is connected to D1, and the ground pin is connected to GND of the ESP32 board. After completing the hardware connections, the ESP32 is connected to a laptop using a USB cable. The Arduino IDE software is then opened, and the required program for reading temperature and humidity values from the DHT11 sensor is uploaded to the ESP32 board. After successful verification and uploading of the program, the Serial Monitor in the Arduino IDE is opened to observe the sensor readings. The Serial Monitor displays the real-time temperature and humidity values measured by the DHT11 sensor.

Air quality sensor

Figure shows the interfacing of the MQ135 air quality sensor with the NodeMCU-32. The sensor connections are made as follows: Vin is connected to 5V, GND is connected to GND, and the analog output pin A0 is connected to GPIO 34 of the ESP32 board. After completing the circuit connections, the ESP32 is connected to a laptop using a USB cable. The Arduino IDE software is then opened, and the program for reading air quality data from the MQ135 sensor is uploaded to the ESP32 board. After successful verification and uploading of the program, the Serial Monitor in the Arduino IDE is opened to observe the sensor readings. The Serial Monitor displays the raw sensor values along with the corresponding air quality condition detected by the MQ135 sensor. The obtained Serial Monitor output is shown in Figure

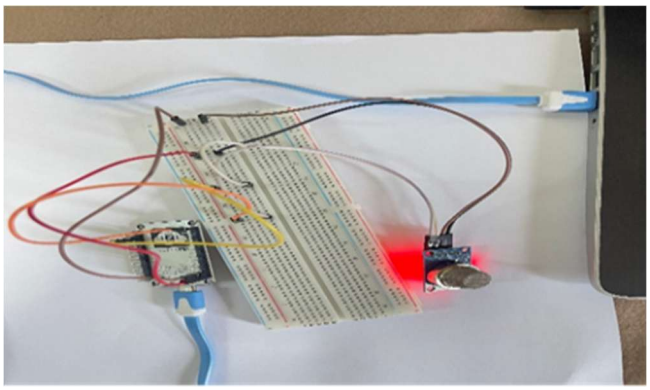

Figure 6.5 Air Quality Sensor

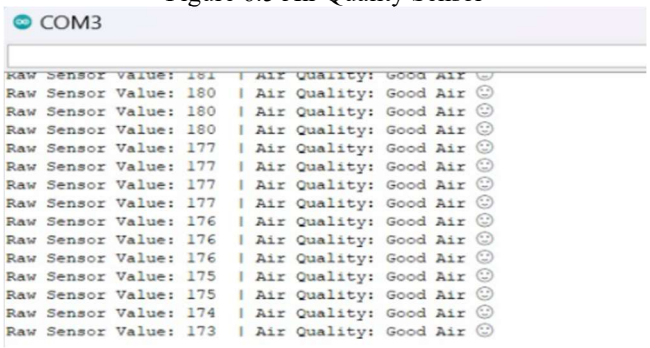

Figure 6.6 Air quality sensor output

BMP280 sensor

Figure illustrates how to connect the BME280 sensor with the ESP32 (VIN to 3.3V, GND to GND, SCL to D2, and SDA to D1), and then connect the ESP32 to a laptop. After completing the connections, the Arduino software is opened and the program is uploaded. Once the program is verified and uploaded successfully, the Serial Monitor is opened to observe the output, which displays the pressure readings.

ISSN: 3107-6513



Figure 6.7 BMP 280

Rain sensor

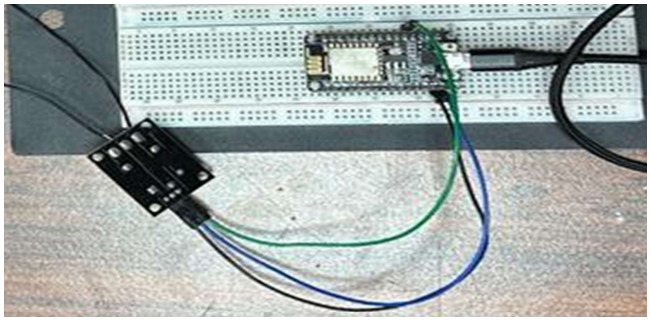

Figure Rain Sensor Testing

Figure illustrates how to connect the rain sensor module with the ESP32 (VCC to 5V, GND to GND, and D0 to D2). After completing the connections, the ESP32 is connected to a laptop, and the Arduino software is opened to upload the program. Once the program is verified and successfully uploaded, the Serial Monitor is opened, where the rain detection status is displayed.

Final Circuit Assembly

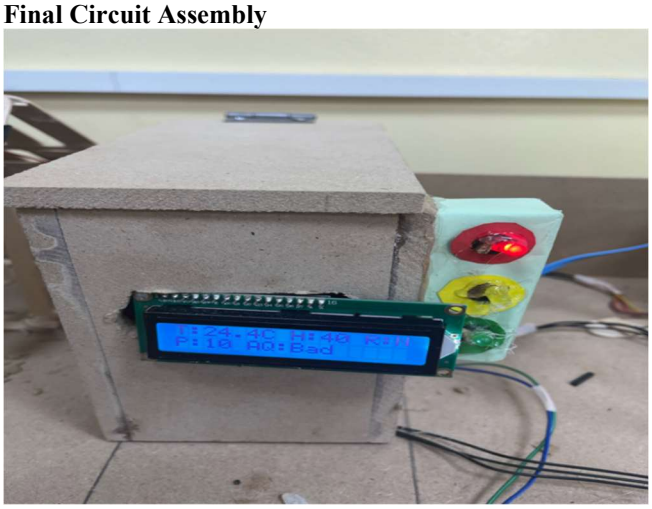

Figure Final Circuit

After the successful completion of individual component testing, the final circuit was constructed by connecting all components as illustrated in Figure A comprehensive program was developed by integrating the code segments required for each sensor and actuator. This unified program was thoroughly verified for logical accuracy and uploaded to the NodeMCU microcontroller. The system's performance was observed via the Serial Monitor, as shown in Figure , to validate the correct functioning of all components. Upon execution, the system began real-time acquisition of environmental parameters including pressure, liquid level, temperature, and air quality. These sensor readings were displayed on the I2C LCD module, providing a continuous visual update of system conditions. Based on the sensor data and predefined threshold values, the NodeMCU triggered the relay module accordingly. This, in turn, activated the appropriate output device such as the LED indicators and LCD display, enabling automatic environmental monitoring and real-time air quality, weather, and rain condition updates based on sensor readings. The successful integration and operation of all components confirmed the system's readiness for practical deployment. The LCD outputs are shown in figure.

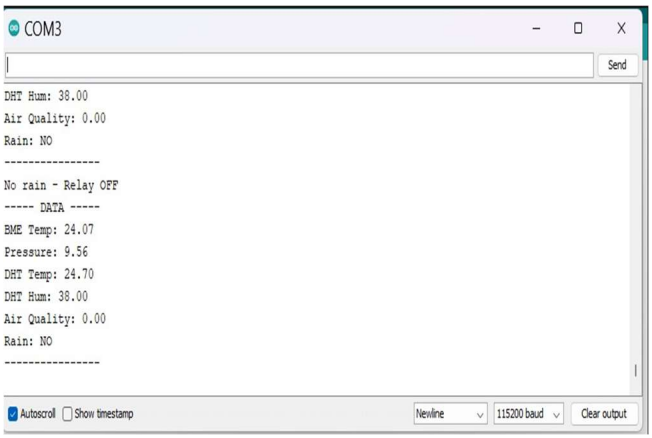

Figure Serial monitor output



Figure LCD output

ISSN: 3107-6513

Thingspeak Results



Figure Thingspeak output

V. CONCLUSION

The Solar Powered Weather Monitoring System using NodeMCU successfully demonstrates the implementation of an IoT-based environmental monitoring solution powered by renewable energy. The system is capable of measuring important weather parameters such as temperature, humidity, atmospheric pressure, rainfall, light intensity, and air quality using various sensors integrated with the ESP32 NodeMCU controller. The collected data is displayed locally through the LCD display and transmitted wirelessly to an IoT cloud platform for remote monitoring and analysis. The use of solar energy makes the system energy-efficient, eco-friendly, and suitable for remote locations where conventional power supply may not be available. The integration of IoT technology enables real-time monitoring and improves accessibility for users through mobile phones or computers. The project also provides a low-cost alternative to traditional weather stations while maintaining acceptable accuracy and reliability for small-scale applications. Overall, the project achieved its objectives of developing a portable, wireless, and sustainable weather monitoring system. The system can be effectively used in agriculture, environmental monitoring, educational research, and smart city applications.

REFERENCES

[1] Kumar, A., Sharma, R., & Verma, P. (2019). Design and development of a low-cost weather monitoring system using Arduino Uno. International Journal of Advanced Research in Electronics and Communication Engineering, 8(4), 120–125.

[2] Patel, R., Shah, D., & Joshi, M. (2020). IoT-based weather monitoring system using NodeMCU (ESP32). International Journal of Innovative Technology and Exploring Engineering, 9(6), 210–216.

[3] Sharma, P., Gupta, N., & Singh, R. (2021). Solar powered IoT weather monitoring system for remote agricultural applications. International Journal of Scientific Research in Computer Science and Engineering, 9(3), 55–61.

[4] Rao, S., Kulkarni, P., & Nair, V. (2018). Rainfall detection and monitoring system using Arduino microcontroller. International Journal of Electronics and Communication Engineering, 7(2), 78–84.

ISSN: **3107-6513**

[5] Singh, V., Patel, K., & Yadav, A. (2022). Smart weather monitoring system using NodeMCU and multiple environmental sensors. International Journal of Internet of Things and Embedded Systems, 11(1), 98–105.
[6] Gupta, R., Meena, S., & Agarwal, T. (2021). Wireless environmental monitoring system using ESP32 and cloud computing technologies. Journal of Cloud Computing and Smart Systems, 6(4), 140–147.
[7] Joshi, M., Desai, R., & Kulshrestha, A. (2023). Advanced IoT weather station integrated with data analytics and predictive models. International Journal of Intelligent Systems and Applications, 12(2), 200–208.
[8]https://www.theengineeringprojects.com/2019/03/introduction-to-dht11.html
[9]https://circuitdigest.com/microcontroller-projects/interfacing-bmp280-sensor-with-arduino.
[10]https://lastminuteengineers.com/rain-sensor-arduino-tutorial/
[11] Arduino Smoke Level Detector using MQ-135 Sensor with Alarm