Weather Reporting and Monitoring System with Wireless Communication

[1]Nway Nway Aung, [2]Yi Mon Htun, [3]Theint Theint Tun, [4]Khin Htwe Myint, [5]Mya Moe Lwin
[1]Demonstrator, [2]Demonstrator, [3]Demonstrator, [4]Demonstrator, [5]Demonstrator
[1]Department of Physics
[1]Meikhtila University, Meikhtila, Myanmar
[1]nwaynwayaung261996@gmail.com

Abstract — **Due to seasonal changes, agricultural activities are affected by irregular rainfall and temperature fluctuations, which leads to crop damage. To reduce these problems, have built a system using IoT technology and ThingSpeak Web Server to monitor and record weather data in real-time and manage agricultural activities more effectively. The main components are NodeMCU (microcontroller), rain sensor, humidity sensor, and barometric pressure sensor. The sensors collect rainfall, humidity, pressure, and temperature data. The data is sent to the ThingSpeak Web Server via Wi-Fi and can be viewed in real-time. Animals may damage, tear, or rest on the cables connected to the sensors, causing the sensor data to be read incorrectly or the system to stop working. If the sensors are moved due to animal movements or dirt falls on the sensors, errors may occur in the weather data sent to the ThingSpeak Web Server. In cases where the internet or Wi-Fi connection is unstable, data transmission from NodeMCU to the ThingSpeak Web Server may be interrupted and the data may not be available in real-time. Therefore, agricultural operations can be managed more efficiently and help improve crop yields. Expand the use of technologies and mobile application chats to improve data accuracy. It is a suitable weather reporting system for farmers, which is low-cost and easy to use. In addition, it should be widely used across the country to further develop the agricultural sector.**

Key words — **BMP180 Pressure Sensor, Humidity Sensor, Internet of Things, NodeMCU Board, Rain Sensor, ThingSpeak Web Server.**

I. INTRODUCTION

Nowadays, IoT technology is becoming increasingly important for automatic data collection and remote monitoring and control. In particular, real-time weather data is very important in sectors such as agriculture and natural disaster prevention. Accurate understanding of weather is important for people's lives and agricultural activities. By using IoT technology and online platforms, weather data can be collected in real-time and practical information can be easily obtained. In previous studies, weather systems were created using Arduino, NodeMCU and various sensors. In addition, data was displayed online using cloud platforms such as ThingSpeak, Blynk, etc. In this work, a system was created based on NodeMCU to collect weather data such as temperature, humidity, pressure from sensors and automatically send it to the ThingSpeak web server. Nowadays, there is a need for systems that can collect weather data in real-time and monitor it online, but the current weather systems in use are expensive, difficult to build and maintain, and have limited space, which limits their practical applications. To solve these limitations, it is necessary to design and build a weather monitoring system that can automatically transmit weather data obtained from sensors to the ThingSpeak web server and monitor them in real-time based on IoT technology [1]. The highlight of this thesis is that a weather system that can automatically transmit data obtained from sensors to the ThingSpeak web server and monitor them in real-time has been designed and implemented. In addition, the use of low-cost hardware makes it convenient for practical use. In addition, by building a system that can display weather data as visualization and widget on the ThingSpeak platform, it is shown that weather data monitoring can be easily studied and analyzed online. The system can serve as a reference model that can be used not only in education and research, but also in agriculture and natural disaster monitoring. The work is systematically structured and begins with an introduction that introduces the background, objectives, and problems of the thesis and a review of previous thesis. Then, the design of the IoT-based weather monitoring system, the hardware and software used, and the operation of the system are described. Finally, the experimental results are discussed, and a summary of the thesis and future work are presented. This structure helps to present the data for the work in a systematic manner and to facilitate the reader to understand the main points of the paper [2].

II. BACKGROUND THEORY

Basically, the design and development of this system are divided into two main parts: hardware architecture and software details. In the hardware architecture, the design of the circuit was constructed and the prototype of the system was built. In the software development, the whole complete prototype was operated via programming codes.

Hardware Architecture

A barometric pressure sensor is a sensor that detects atmospheric pressure. Sensors that detect atmospheric pressure are called barometric pressure sensors. This sensor is used for indoor navigation, elevator/floor change detection and weather monitoring. Murata's barometric pressure sensors are compact, highly accurate and consume low power. Therefore, they are being used in a wide range of applications including wearable devices, the wellness market and industrial equipment. The products with waterproof functionality are suitable for devices that require higher levels of waterproof performance, like underwater applications, in addition to environments such as those with sweat and rain. Using barometric pressure for altitude measurement must reflect the fact that air pressure is not static over time and fluctuates based on current atmospheric conditions and weather. The key to solving this issue is to simultaneously measure weather induced atmospheric pressure changes at multiple fixed locations, and then to use this information as a real-time reference for the actual pressure readings measured using the sensor in the smartphone. Figure 1 shows the BMP 180 pressure sensor before connecting to the NodeMCU board.

Humidity is a very important feature in detecting a fire. In case fire the air will be dry thus decreasing the humidity [3]. This decrease in humidity can give us indication of fire. DHT-11 uses resistive type humidity measurement component. Humidity sensing component is used, of course to measure humidity, which has two electrodes with moisture holding substrate (usually a salt or conductive plastic polymer) sandwiched between them. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes. In the Arduino IDE, enter and will print the temperature and relative humidity values on the serial monitor. The DHT-11 sensor can be used to detect humidity in the range of 20 - 90% RH with the accuracy of ±5% RH. Figure 2 shows the humidity sensor before connecting to the NodeMCU board.

The term IoT, or Internet of Things, refers to the collective network of connected devices and the technology that facilitates communication between devices and the cloud, as well as between the devices themselves. The Internet of Things integrates everyday “things” with the internet. As computing devices shrank in size, these chips also became smaller, faster, and smarter over time. The cost of integrating computing power into small objects has now dropped considerably. A whole industry has sprung up with a focus on filling our homes, businesses, and offices with IoT devices. These smart objects can automatically transmit data to and from the internet. All these “invisible computing devices” and the technology associated with them are collectively referred to as the Internet of Things. Figure 3 shows the Internet of Things which is connected to wireless communication.

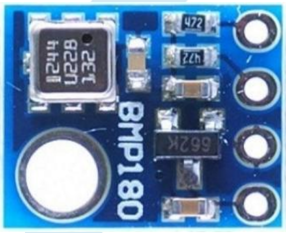

Fig 1: BMP 180 pressure sensor

Fig 2: Humidity sensor

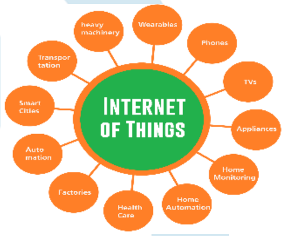

Fig 3: Internet of Things

NodeMCU is an open-source development board and firmware based in the widely used ESP8266-12E Wi-Fi module. It allows user to program the ESP8266 Wi-Fi module with the simple and powerful Lua programming language or Arduino IDE. With just a few lines of code user can establish a Wi-Fi connection and define input/output pins according to user’s needs exactly like Arduino, turning user ESP8266 into a web server and a lot more. It is Wi-Fi equivalent of ethernet module. With its USB-TTL, the NodeMCU development board supports directly flashing from USB port. It can be used as access point and /or station, host a web server or connect to internet to fetch or upload data. Figure 4 shows the NodeMCU board that is used throughout the system.

Rain sensors are devices designed to detect the presence and intensity of rainfall. They are widely used in weather monitoring systems, automatic irrigation controllers, automotive applications, and smart home devices. The primary function of a rain sensor is to convert the physical phenomenon of rain into an electrical signal that can be interpreted by a microcontroller or data acquisition system. The basic working principle of a rain sensor relies on the change in electrical properties caused by water. Most rain sensors consist of a sensing surface with conductive traces, typically arranged in a grid pattern, which can detect the presence of water droplets. When raindrops fall onto the sensor, they create conductive bridges between the traces, allowing a current to flow. This change in conductivity is then converted into an analog voltage or digital signal, which indicates the intensity or presence of rainfall. Figure 5 shows the humidity sensor before connecting to the NodeMCU board.

ThingSpeak is IoT cloud platform where user can send sensor data to the cloud. User can also analyze and visualize his data with MATLAB or other software, including making his own applications. The ThingSpeak service is operated by MathWorks. ThingSpeak is free for small non-commercial projects. It includes a Web Service (REST API) that lets user collect and store sensor data in the cloud and develop Internet of Things applications. It works with Arduino, Raspberry Pi -and MATLAB (premade libraries and APIs exists) But it should work with all kind of programming languages, since it uses a REST API and HTTP. Figure 6 shows ThingSpeak web server which is displayed the condition of the air quality via sensors. Figure 7 shows the schematic diagram of the system.

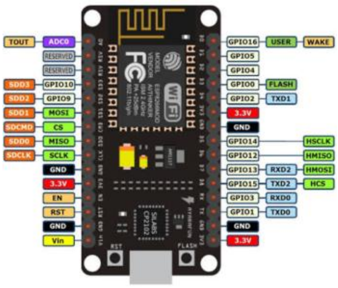

Fig 4: NodeMCU board

Fig 5: Rain sensor

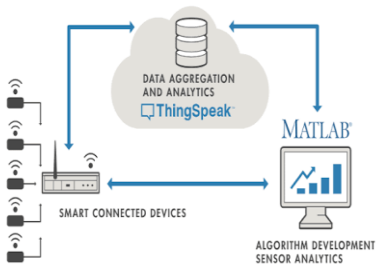

Fig 6: ThingSpeak web server

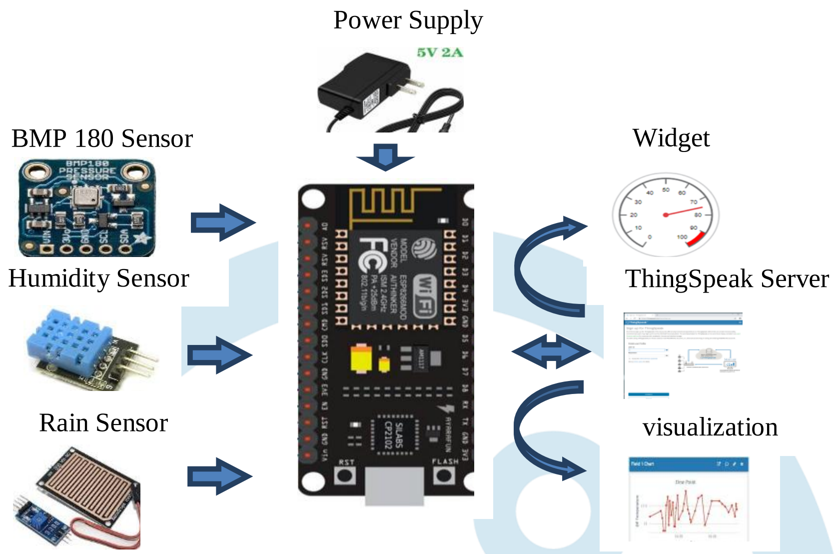

Fig 7: Schematic diagram of the weather reporting and monitoring system

Software Architecture

In terms of software, it includes Arduino IDE software, Proteus software, Microsoft office, programming language, and ThingSpeak web server. First of all, Arduino IDE software can be easily downloaded for free from Google using the internet. The library files that will be used for sensors also need to be added to Arduino IDE. The required library files can be easily obtained from Google. The Proteus software is used to draw the circuit diagram. Since the system is built using the internet, simulation cannot be done in Proteus software. As for the programming language, the entire system will be written in MATHLAB language. To monitor the data sent by the sensors, ThingSpeak cloud server is used. ThingSpeak cloud server needs to be registered with Gmail. Since it is freely available from the MathWorks group, it can be easily used for many IoT projects. Once user register, user can create user's own channel and create the projects user want. Finally, after writing the code in Arduino IDE, upload it to the NodeMCU board. Then power on the system and run it. Once done, the system user has created will start running. Figure 8 shows the flowchart of the system.

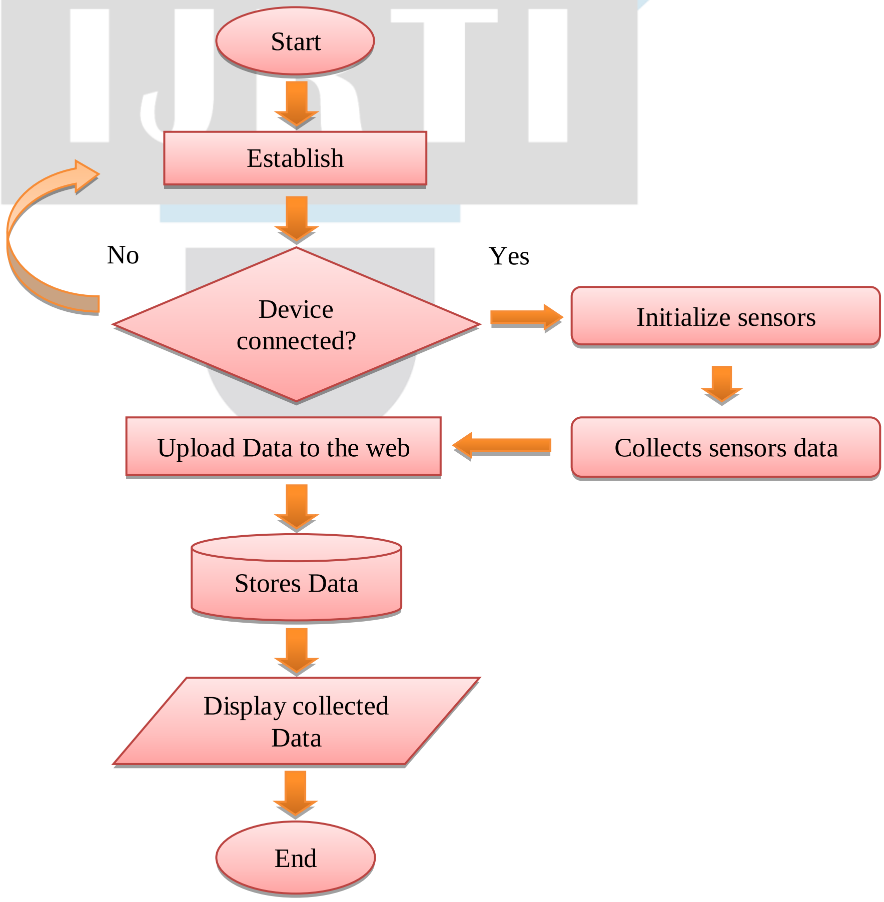

Fig 8: Flowchart of the system

III. Working System

This system includes NodeMCU ESP8266 Wi-Fi module, humidity sensor, barometric pressure sensor, rain sensor, 5V DC power supply, Arduino IDE, and ThingSpeak web server. The NodeMCU ESP8266 acts as the central processing unit that collects environmental data from various weather sensors via its GPIO pins. It collects data from connected sensors, process the signals, and converts them into digital information. Then, it transmits the processed data to the ThingSpeak web server through a Wi-Fi network for real-time monitoring. The humidity sensor measures the amount of water vapor present in the air and detects changes in environmental humidity. When the surrounding humidity level changes, the capacitance of the sensor element changes accordingly. The internal circuit converts this change into digital data representing relative humidity (RH or %). The barometric pressure sensor uses a piezoresistive diaphragm to measure atmospheric pressure. When the atmospheric pressure changes, the diaphragm deforms slightly and alters the electrical resistance. The sensor converts this resistance change into digital pressure data expressed in hectopascal (hPa). The rain sensor detects rainfall using a conductive sensing plate with parallel copper tracks. When raindrops fall on the plate, the resistance between the tracks decreases due to water conductivity. The microcontroller reads the voltage change and determines the presence and intensity of rainfall. The proposed IoT-based weather reporting and monitoring system collects environmental data using different sensors such as temperature, humidity, rain, and pressure sensors. These sensors are connected to the NodeMCU ESP8266 microcontroller, which reads and processes the sensor data. The NodeMCU then transmits the collected data to the ThingSpeak web server through a Wi-Fi network. The ThingSpeak platform stores, analyzes, and displays the data in graphical form. As a result, users can monitor weather conditions in real time through the internet. The sensors send their data to the NodeMCU (ESP8266), which acts as the main controller to process and convert the signals into digital values. The NodeMCU then connects to a Wi-Fi network and transmits the collected data to the ThingSpeak cloud platform. The uploaded data is stored, analyzed, displayed in graphical form on the web server, allowing users to monitor weather conditions remotely in real time. The IoT weather monitoring system has been fully built, with all sensors and the NodeMCU properly connected and programmed. The system has been tested and is now fully operational, ready to collect and transmit real-time weather data to the ThingSpeak web server for remote monitoring in fields, gardens. Figure 9 and 10 show the circuit diagram and completed system.

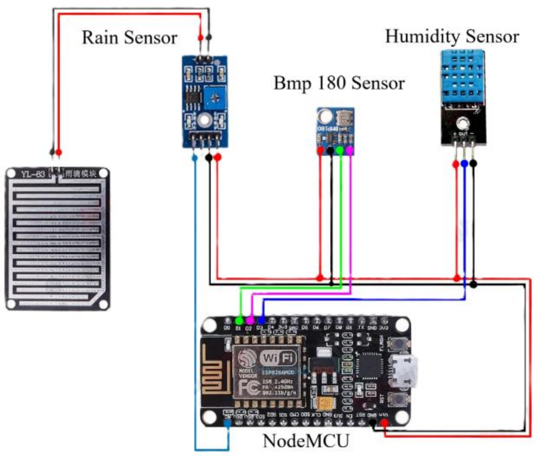

Fig 9: Circuit diagram

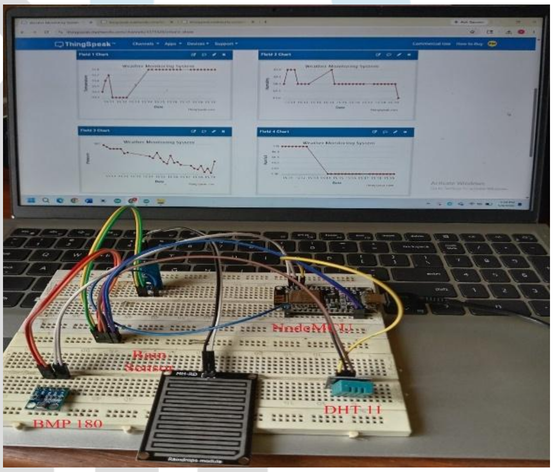

Fig 10: Completed system

IV. Results and Discussion

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads—the template will do that for you. An IoT-based weather monitoring system was developed to measure and upload environmental parameters such as temperature, humidity, atmospheric pressure, and rainfall to the cloud using the ThingSpeak web server. The experimental setup consisted of sensors connected to a microcontroller with Wi-Fi capability such as ESP8266, which continuously transmitted sensor data to the cloud platform for visualization and analysis. The system included the humidity sensor (DHT11), atmospheric pressure sensor (BMP180), rain sensor module, NodeMCU board based on Wi-Fi enabled microcontroller and ThingSpeak official website cloud server for data storage and monitoring. The sensors collected environmental data every 3–5 seconds and uploaded the readings to the cloud channel created in ThingSpeak. The data was displayed in graphical format and could be monitored remotely through the internet. The system successfully monitored real-time environmental conditions and transmitted the readings to the cloud server without significant delay. Temperature increased gradually from morning to afternoon and decreased during rainfall conditions, which validates proper sensor operation. Humidity values increased during rainy periods, indicating an inverse relationship between temperature and humidity. Atmospheric pressure showed slight variation throughout the day, which is consistent with changing weather conditions. The rain sensor accurately detected rainfall events and updated the cloud channel immediately. ThingSpeak dashboard displayed real-time graphs that helped visualize weather trends effectively. Figures 11, 12, 13 and 14 show the displaying the temperature, humidity, pressure and rainfall on ThingSpeak web server. Table 1 shows the experimental results of the sensors data.

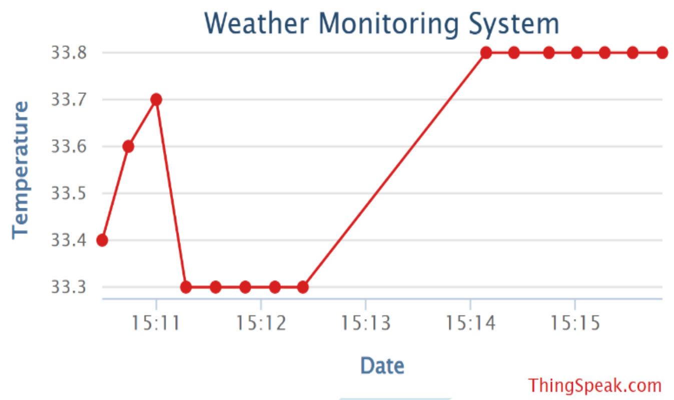

Fig 11: Displaying the temperature on ThingSpeak web server

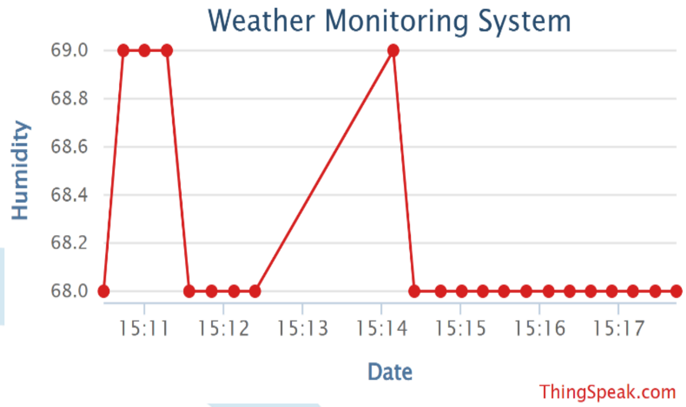

Fig 12: Displaying the humidity on ThingSpeak web server

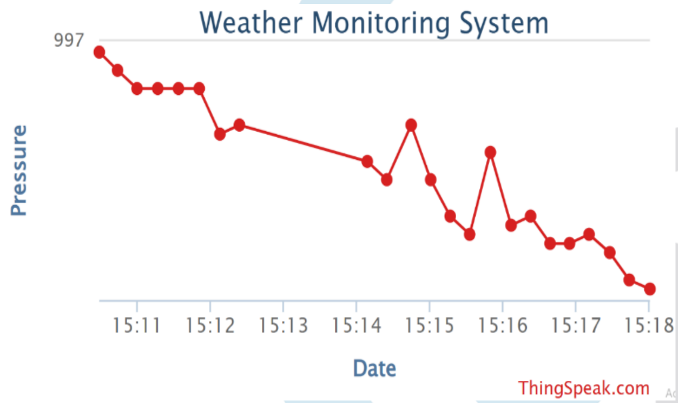

Fig 13: Displaying the pressure on ThingSpeak web server

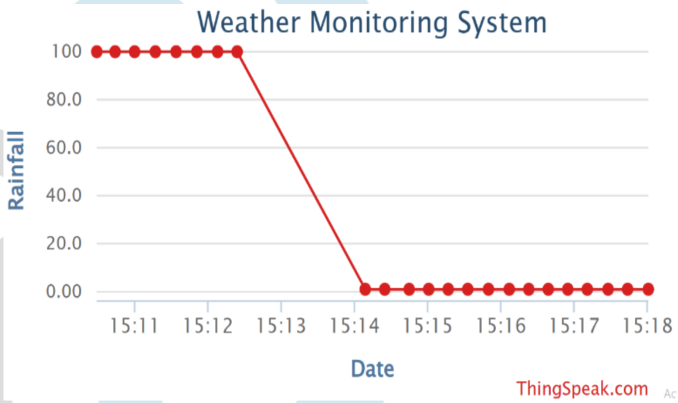

Fig 14: Displaying the rainfall on ThingSpeak web server

Table 1 Experimental Results of the sensors data

No:	Time (hr)	Pressure (hPa)	Temperature (°C)	Humidity (%)	Rainfall (mm)	Weather condition
1	7:00 AM	1030	32	22	0	cloudy
2	8:00 AM	1032	31	23	0	cloudy
3	9:00 AM	1010	30	24	0	cloudy
4	10:00 AM	1000	26	32	0	cloudy
5	11:00 AM	995	25	41	1	rainy
6	12:00 AM	990	24	43	3	rainy
7	13:00 PM	981	23	44	5	rainy
8	14:00 PM	950	22	49	7	rainy
9	15:00 PM	1070	33	20	0	sunny
10	16:00 PM	1065	35	18	0	sunny
11	17:00 PM	1050	31	21	0	sunny
12	18:00 PM	1044	28	26	0	sunny

The IoT-based weather monitoring system demonstrated efficient real-time acquisition and transmission of environmental data using the ThingSpeak cloud platform. The integration of sensors with a Wi-Fi-enabled microcontroller enabled continuous monitoring of weather parameters such as temperature, humidity, atmospheric pressure, and rainfall. During experimentation, the system successfully uploaded sensor readings to the cloud server at regular intervals without major communication failure. The graphical visualization provided by ThingSpeak made it easy to observe environmental changes over time and analyze weather trends remotely through internet connected devices. The temperature readings showed a gradual increase from morning to afternoon due to solar heating and later decreased during rainfall conditions. At the same time, humidity values increased when rain was detected, confirming the expected atmospheric relationship between temperature and moisture content. Atmospheric pressure readings exhibited slight fluctuations, which are common in changing weather conditions. These observations indicate that the sensors functioned accurately and responded properly to environmental variations. One of the major advantages of the proposed system is its ability to provide remote monitoring and cloud-based data storage. Unlike conventional weather monitoring systems that require manual observation, the IoT-based approach automates data collection and allows users to access information from any location. This improves efficiency, reduces human effort, and supports continuous monitoring. The use of the ThingSpeak web server also enhanced the analytical capability of the system [3]. The platform enabled real-time graph plotting, data logging, and online visualization, making the system suitable for applications such as smart agriculture, environmental monitoring, climate observation, and disaster management. However, some limitations were observed during experimentation. Sensor accuracy may vary due to environmental noise, improper calibration, or exposure to extreme conditions. Network interruptions can also delay data transmission to the cloud server. Additionally, low-cost sensors such as DHT11 may provide less precise readings compared to industrial-grade devices. Power consumption and internet dependency are further challenges for long-term outdoor deployment. To improve the system, future work may include the integration of more accurate sensors, the addition of solar-powered energy systems, the implementation of machine learning for weather prediction, the mobile application support for notifications and alerts and GPS-based location tracking for distributed monitoring stations. Overall, the developed IoT-based weather monitoring system proved to be reliable, cost-effective, and practical for real-time environmental monitoring applications. The experimental results and observations confirm that cloud-connected IoT technology can significantly improve the efficiency and accessibility of weather monitoring systems [4].

V. Conclusion

An IoT-based weather monitoring system using the ThingSpeak web server demonstrates how modern technology can make environmental data collection more efficient, accurate, and user-friendly. By integrating sensors (for temperature, humidity, pressure) with a microcontroller and internet connectivity, the system can send data to the cloud in real-time. Using ThingSpeak as a platform offers several advantages, to easy visualization of data through graphs, to support for data analysis with built-in tools, and to enables remote monitoring from anywhere via web. This makes the system highly practical for applications such as agriculture, climate research, disaster management, and smart cities. Overall, this thesis highlights the power of IoT in transforming traditional weather monitoring into a smart, automated, and scalable solution. Despite some limitations, such as dependence on internet connectivity and sensor inaccuracies, the system is still cost-effective and adaptable. Future developments could include adding more advanced sensors, integrating machine learning for weather forecasting, and improving system security. Ultimately, this IoT-based approach using ThingSpeak is a reliable and innovative way to continuously monitor the environment and support decision-making.

VI. Acknowledgments

Special thanks are extended to the Board of Editors, International Journal for Innovative Research in Technology (IJRTI), for allowing me to submit the research article and invaluable suggestions. I would like to express my deepest gratitude to family for their encouragement to submit this paper. Further, I would like to thank researchers, teachers and students for my research paper.

References

[1] K. S. Win, K. K. Aung, P. L. Htun, N. S. L. K. Min and P. D. Htun, “Development of IoT Based Weather Reporting System using ThingSpeak Web Sever”, International Journal for Research Trends and Innovation, 2024.

[2] I. A. Ndakara and A. T. Ojotule, “An IoT-Based Weather Monitoring System”, International Conference on Engineering and Applied Natural Sciences, 2022.

[3] S. Masanta, A. Biswas, S. Santra, P. Mondal, M. Chanda and S. Ghosh, “IoT-based Smart Weather Monitoring System”, International Journal on Science and Technology, 2025.

[4] A. Choudhary and R. Pol, “IoT Based Weather Monitoring System”, International Journal of Innovative Research in Science, Engineering and Technology, 2023.