IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Real-Time Air Quality Monitoring System Through IoT and ThingSpeak Cloud Server

[1]May Myo Lwin, [2]Mya Khin Zan, [3]Theint War War Khaing, [4]Yadanar Win Maung, [5]Moh Moh Aung

[1]Demonstrator, [2]Associate Professor, [3]Lecturer, [4]Associate Professor, [5]Demonstrator

[1]Department of Physics

[1]Taunggyi University, Taunggyi, Myanmar

Abstract: Air pollution has become a significant environmental and public health concern, especially in densely populated areas such as university campuses. Continuous monitoring of air quality is essential to understand pollution levels and take preventive measures. This work presents a real-time air quality monitoring system for campus environments using IoT technology and the ThingSpeak cloud server platform. The proposed system uses air quality sensors to measure environmental parameters such as harmful gases, temperature, humidity, and particulate matter in the ambient air. A Wi-Fi-enabled microcontroller collects data from the sensors and transmits it to the cloud in real-time. The data is then uploaded to the ThingSpeak platform, where it is stored, processed, and displayed in graphical form for easy monitoring and analysis. This system allows continuous monitoring of air quality conditions through an online dashboard that can be accessed remotely via computers or mobile devices. The collected data can help campus authorities and researchers identify pollution trends, maintain a healthier environment, and support smart campus operations. Overall, the proposed IoT-based air quality monitoring system provides a cost-effective, reliable, and scalable solution for real-time environmental monitoring and data analysis.

Keywords - Gas Sensor, Humidity Sensor, Internet of Things (IoT), Liquid Crystal Display (LCD), NodeMCU ESP8266, and ThingSpeak Cloud Server.

I. INTRODUCTION

Air pollution is one of the most serious environmental problems affecting human health and the quality of life. Rapid urbanization, increasing transportation, and industrial activities have led to a significant rise in air pollutants such as carbon dioxide, carbon monoxide, and particulate matter. These pollutants can cause respiratory diseases, allergies, and other health issues. Therefore, continuous monitoring of air quality has become essential, especially in places with large populations such as university campuses. Traditional air quality monitoring systems are often expensive, complex, and limited to specific locations. With the advancement of the Internet of Things, it is now possible to develop low-cost and efficient monitoring systems that can collect and transmit environmental data in real time. IoT technology allows sensors, microcontrollers, and communication modules to work together to gather environmental information and send it to cloud platforms for analysis and visualization. This system proposes a real-time air quality monitoring system for a campus using IoT and the ThingSpeak cloud server platform. The system uses air quality sensors to detect pollutants and environmental parameters such as temperature, humidity, and gas concentration. A Wi-Fi enabled microcontroller processes the sensor data and uploads it to the ThingSpeak cloud server, where the data can be stored, analyzed, and displayed through graphical dashboards. The implementation of this system enables campus authorities, researchers, and students to monitor environmental conditions remotely through the internet. Real-time data visualization helps in identifying pollution trends and taking necessary actions to maintain a healthy and sustainable campus environment. This project also demonstrates how IoT technology can be applied to environmental monitoring systems in smart campuses and smart cities.

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. While in the software development, the whole complete prototype was operated via programming codes.

2.1 Hardware Architecture

NodeMCU is an open-source development board and firmware based in the widely used ESP8266- 12E Wi-Fi module [1]. It allows user to program the ESP8266 Wi-Fi module with the simple and powerful Lua programming language or Arduino IDE [2]. 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 1 shows the NodeMCU board that is used throughout the system.

The MQ135 is one of the popular gas sensors from the MQ series of sensors that are commonly used in air quality control equipment. It operates at 5V and consumes around 150mA. It can detect gases like Ammonia, sulphur, Benzene, CO_2, and other harmful gases and smoke. When there is polluted gas in the environment where the sensor is located, the conductivity of the sensor increases with the increase of the concentration of polluted gas in the air. The MQ135 gas sensor has a high sensitivity to ammonia, sulphide, and benzene-based vapours, and is ideal for monitoring smoke and other harmful gases. This sensor can detect a variety of harmful gases and is a low-cost sensor suitable for a variety of applications. Figure 2 shows the gas sensor. Figure 2 shows the humidity 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. The DHT11 and DHT22 sensors are fairly easy to connect. Although supply voltage ranges from 3.3V to 5.5V, 5V supply is recommended. 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 3 shows the humidity sensor before connecting to the NodeMCU board.

Fig. 1. NodeMCU ESP8266

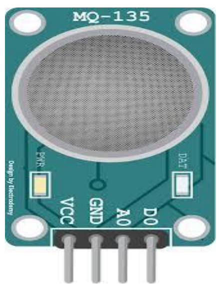

Fig. 2. Gas sensor

Fig. 3. Humidity sensor

This is I2C interface 16+2 LCD display module, a high-quality 2 line 16-character LCD module with on board contrast control adjustment, backlight communication interface. For Arduino beginners, no more cumbersome and complex LCD driver circuit connection. The real significance advantages of this I2C Serial LCD module will simplify the circuit connection, save some I/O pins on Arduino board, simplified firmware development with widely available Arduino library. Figure 4 shows the liquid crystal display before connecting 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 [4]. 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 5 shows the internet of things which is connected to wireless communication.

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 cloud server which is displayed the condition of the air quality via sensors.

Fig. 4. Liquid crystal display

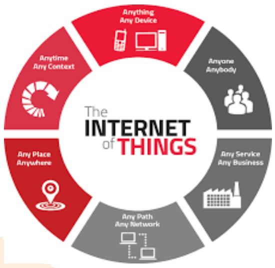

Fig. 5. Internet of things

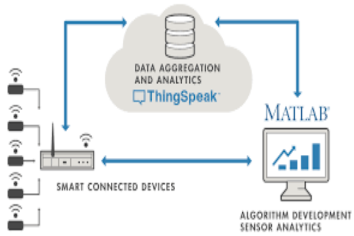

Fig. 6. ThingSpeak cloud server

2.2 Software Architecture

In this research, the MATLAB programming language is written on Arduino IDE which is an open-source software. The code will be written on this software and it will be uploaded to NodeMCU board. The program of microcontroller reads the program and sends the data to the indicator LCD display and ThingSpeak cloud server. ThingSpeak is a platform with cloud server to display analyzed data from sensors. After connecting microcontroller USB via computer, it will choose port. Then, it will complete and upload the program toward the NodeMCU board. Figure 7 and 8 show the block diagram and the flowchart of the system.

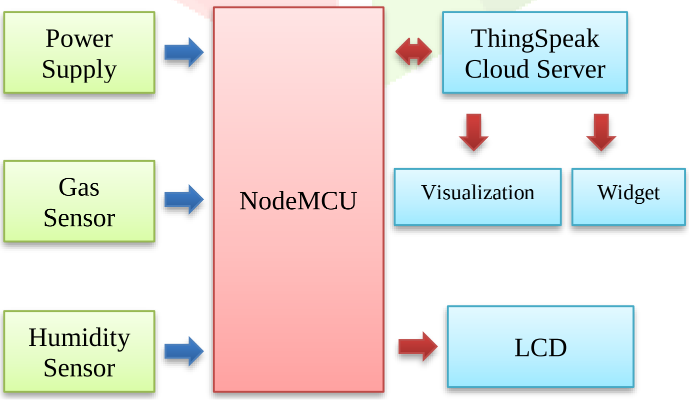

Fig. 7. Block diagram of the system

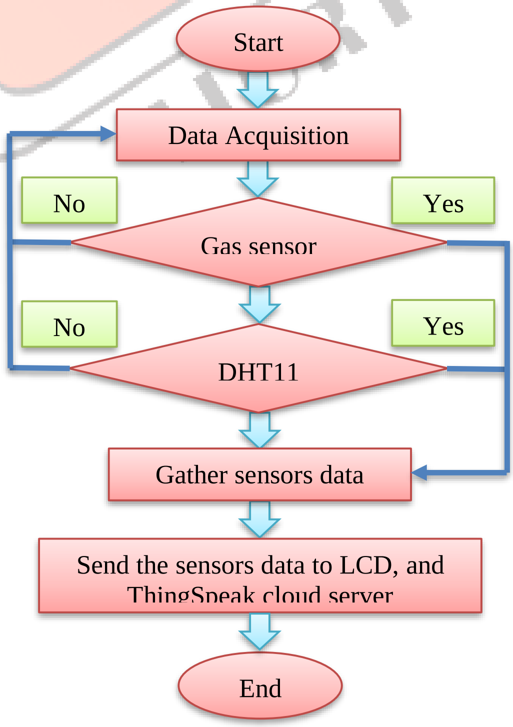

Fig. 8. Flowchart of the system

III. Working System

Based on IoT, the Real-Time Air Quality Monitoring System is integrated with hardware and software through Wi-Fi communication. In terms of hardware components, NodeMCU ESP8266 Wi-Fi module, gas sensor, humidity sensor, LCD, buzzer, and power supply installed. As for the software application, the main thing used is ThingSpeak cloud server. The operating voltage of the NodeMCU board is 3.3V, but the input voltage must be provided between 5 V and 9 V. The microcontroller included in the NodeMCU board is ordered to be given to command in the Mathlab programming language. The Mathlab programming language is written in the Arduino IDE. In this system, the use of the gas sensor to detect the air quality, a humidity sensor is used to detect the temperature and humidity around the environment. ThingSpeak cloud server is used to monitor the data sent by the sensors in real time during dangerous air quality conditions. At the same time, LCD screen is used to display the air quality and weather condition. The circuit diagram is drawn with proteus software. The data sent by the sensors will be recognized by the microcontroller, and through it, thingSpeak cloud server, and LCD screen will be operated as an alarm system according to the instructions of the programming language. In the system, the dangerous air quality condition and the normal condition are controlled by the language. When the dangerous condition is reached, the person who controls the system warns the nearby students. In addition, the conditions occurring are displayed on LCD displays in places where the students can see them. After that, a signal sound system has also been installed. After all the systems were tested, the complete system was built and tested in parallel. Figures 9 and 10 show the circuit diagram and the completed circuit of the system.

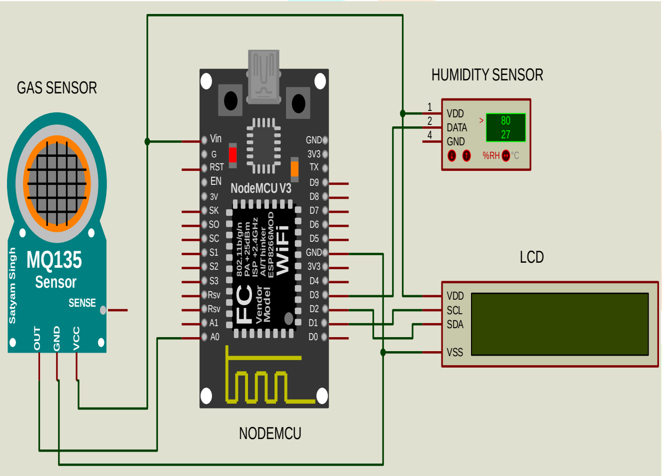

Fig. 9. Circuit diagram of the system

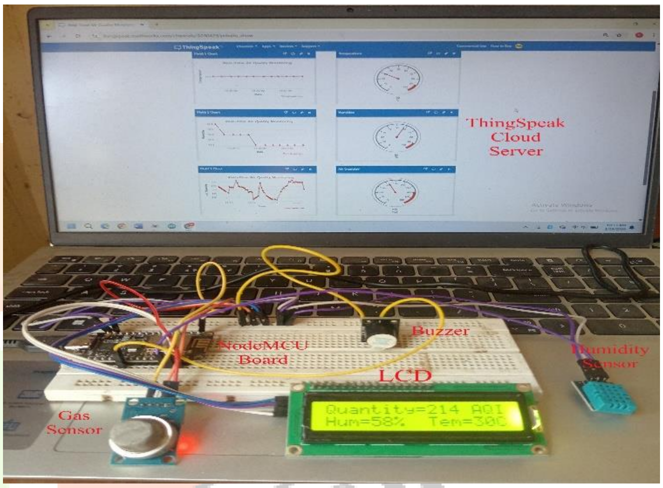

Fig. 10. Completed system

IV. Experimental Results

The components included in the system were tested in various ways. Firsty, the humidity sensor was tested with an ice cube, soldering, and hair dryer. During the testing, a small amount of data error was found. It was found that the probability error obtained from the experiments was small. The error that occurred was not significant, so it was used for the system. The gas sensor was tested with nitrogen oxides (NO_x), sulfur dioxide (SO_2), and carbon monoxide (CO), which are air pollutants. According to the tests, it was found that the values read by the sensor were different. The sensor was connected to an analog pin and used. When connecting the LCD to the NodeMCU board, the text did not appear on the LCD screen, so had to reduce the version of the library file in the Arduino IDE and use it in various ways. When using the ThingSpeak cloud server, had to log in with a gmail account. It was smooth and easy to use when logging in and connecting to Wi-Fi. One thing to note is that if the Wi-Fi connection is poor, there is a slight delay when displaying real-time data sent by sensors. Figure 11 and 12 show the displaying the temperature and humidity on ThingSpeak cloud server. Figure 13 and 14 show the displaying the air quality on ThingSpeak cloud server and sensors value on LCD. Table 1 shows the experimental results of various gases. Table 2 shows the experimental results of humidity sensor.

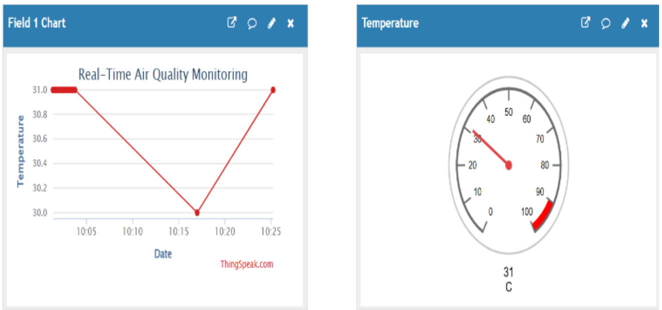

Fig. 11. Displaying the temperature on ThingSpeak cloud server

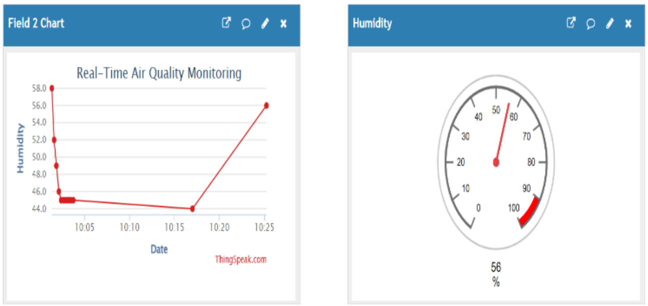

Fig. 12. Displaying the humidity on ThingSpeak cloud server

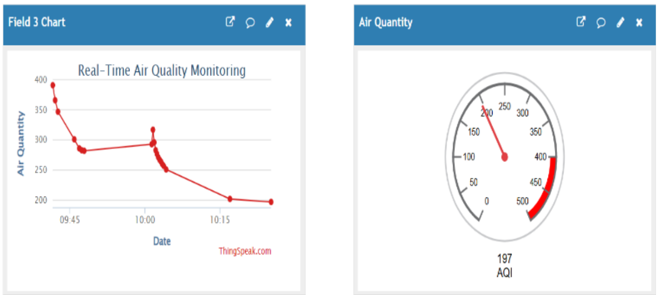

Fig. 13. Displaying the air quality on ThingSpeak cloud server

Fig. 14. Displaying the sensors value on LCD

Table 1: Experimental Results of Various Gases

No:	**Gases**	**Trial No:**	**Time (s)**	**Values (AQI)**	**Level**
1	Nitrogen Oxides (NO_x)	1	1	29	Moderate
		2	5	41	Unhealthy for sensitive groups
		3	10	56	Unhealthy
		4	15	72	Very unhealthy
		5	20	89	Hazardous
2	sulfur dioxide (SO_2)	1	1	26	Moderate
		2	5	39	Unhealthy for sensitive groups
		3	10	52	Unhealthy
		4	15	68	Very unhealthy
		5	20	83	Hazardous
3	carbon monoxide (CO)	1	1	27	Moderate
		2	5	35	Unhealthy for sensitive groups
		3	10	49	Unhealthy
		4	15	64	Very unhealthy
		5	20	78	Hazardous

Table 2: Experimental Results of Humidity Sensor

No:	**Sensor**	**Trial No:**	**Time (hour)**	**Temperature (°C)**	**Humidity (%)**
1	Humidity (DHT11)	1	9:00 AM	30	43
		2	10:30 AM	32	38
		3	12:00 PM	35	32
		4	13:30 PM	36	29
		5	15:00 PM	31	39

V. Conclusion

The development of a Real-Time Air Quality Monitoring System using IoT and the ThingSpeak cloud server demonstrates an effective and scalable approach to monitoring environmental conditions on a campus. By integrating sensors with microcontrollers and cloud-based analytics, the system successfully collects, processes, and visualizes air quality data in real time. This work highlights the importance of continuous air quality monitoring in ensuring a healthy and safe environment for students, faculty, and staff. The use of IoT technology enables remote access to data, timely alerts, and informed decision-making, which are crucial for addressing pollution-related issues. Furthermore, the implementation of the ThingSpeak platform provides a user-friendly interface for data storage, analysis, and graphical representation, enhancing the system's usability and efficiency. This research also demonstrates cost-effectiveness, flexibility, and the potential for future expansion, such as integrating additional environmental parameters or implementing automated control systems. This system serves as a practical solution for smart campus initiatives and contributes to building sustainable and environmentally conscious communities.

VI. Acknowledgements

Special thanks are extended to the Board of Editors, International Journal of Creative Research Thoughts (IJCRT), 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] Zeroday, "A lua based firmware for wifi-soc esp8266," Github. Retrieved 2 April 2015. [online]. Avaiable: https://github.com/nodemcu/nodemcu-firmware

[2] Benchoff, "An SDK for the ESP8266 Wi-Fi chip," Hackaday. Retrieved 2 April 2015. [online]. Avaiable: https://hackaday.com/2014/10/25/an-sdk-for-the-esp8266-wifi-chip

[3] Babin, M. Steven, "Relative Humidity & Saturation Vapor Pressure: A Brief Tutorial," Johns Hopkins University. Retrieved 28 November 2022. [online]. Avaiable: https://fermi.jhuapl.edu/people/babin/vapor/vapor.html

[4] Gillis, Alexander, "What is internet of things (IoT)?," IOT Agenda. Retrieved 17 August 2021. [online]. Avaiable: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT