Environmental Monitoring System

Sri Harsha Mittapalli
Computer Science
Cleveland State University
Cleveland Ohio United States
s.mittapalli@vikes.csuohio.edu

Pranay Kumar Gattu
Computer Science
Cleveland State University
Cleveland Ohio United States
p.gattu26@vikes.csuohio.edu

Sai Charan Reddy Kumbham
Computer Science
Cleveland State University
Cleveland Ohio United States
s.kumbham17@vikes.csuohio.edu

Swathi Takkellapati
Computer Science
Cleveland State University
Cleveland Ohio United States
s.takkellapati@vikes.csuohio.edu

ABSTRACT

Internet of Things would play one of the major roles in today's era. These systems are specially designed for monitoring and the data generated would be transmitted and stored in the cloud. This report talks about the design and implementation of the environmental monitoring system web application which has been developed using IOT components such as ESP 8266, MQ 2, DHT 11, and MQ 135. The application which is deployed would utilize the sensors for the collection of data based on temperature, Humidity, Smoke etc. The collected data would be transmitted to the cloud server and the data stored in the cloud server can be accessed by using the Environmental Monitoring system web application. The experimental results would show us the usefulness of the Environmental monitoring system web application which is used in monitoring the environmental conditions and collecting the data for our future analysis. Finally, we can say that the system of our web application would be demonstrated as a practical or real-time web application which is based on the IOT technology.

CCS CONCEPTS

• IOT • hardware • software • Wireless communication
• programming

KEYWORDS

Internet of things, Web server, Cloud, environmental monitoring system, sensors, Arduino, Things speak, Streamlit, Web application, DHT-11, MQ 2, MQ 135, ESP 8266.

ACM Reference format:

Sri Harsha Mittapalli, Pranay Kumar Gattu, Sai Charan Reddy Kumbham and Swathi Takkellapati. 2023. Environmental Monitoring System. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1 Introduction

The IOT is one of the transformative technologies which can revolutionize the world that enables monitoring and controlling of some of the vital phenomena of the environment even though, the use of devices would have capable of sensing, processing and transmitting data wirelessly to the cloud. The cloud would store and analyze the data and it will present to us in a useful form. The information which is stored in the cloud can be retrieved or accessed by the end users using the web or the mobile which depends upon the requirements and the suitability. Here the internet plays a major role which is efficient and reliable. It would shift the data communication from the device to the cloud and again from the cloud to the end users. The IOT is nothing but the network of physical objects and vehicles and some of the other items which are embedded with electronics, sensors, network connectivity and software. These "things" can collect and transmit data over the internet which allows them in monitoring, control, and optimize remotely. Some of these things would be capable of transmitting and sensing the data such as humidity, smoke, temperature and air quality. The Internet of Things has a revolutionary way in which it would interact with the real world. The Environmental Monitoring System is one of the important IOT web applications that involve monitoring the environment's surroundings. This report would represent the implementation and the results of the Environmental Monitoring System web application. The web application is comprised of a central Arduino UNO board which would interface at the input with the temperature and humidity monitoring sensor called a DHT 11 sensor and output through the ESP8266 Wi-Fi module which is used to transmit the sensor data through the internet to the cloud via remotely using the things speak. Through the things speak the data would be analyzed and stored. The Things Speak server analytics would be carried out on the data and a trigger would be generated. A web application is developed using the stream lit and the data retrieved from the Things Speak is displayed to the user from anywhere in the world. The developed system is a low-cost web application which gives a brief introduction to the design and the implementation of the entire IOT web application which involves all the aspects from the sensors and the wireless transmission into the cloud and retrieving of the data from the cloud using the web application. Here it would involve the study and the deployment of the Arduino board and its interfacing with input and output modules for example the WIFI module and the sensors and Things speak usage which is an open-source API and lastly the development of the web application which is developed using the stream lit. The project results show us the real-time monitoring of the temperature, humidity levels, air quality levels, and smoke from anywhere in the world and it can be analyzed statistically.

1.1 Problem Statement

Environmental pollution is a growing concern in the world. It is important to monitor the environmental conditions to recognize and address the potential hazards and threats. There are some issues with the traditional monitoring systems. There are some devices which cannot produce accurate data for us. It is very difficult for us to monitor all the environmental factors individually. With the traditional monitoring systems, it is a time-consuming process. In the traditional method, we need to depend on multiple sources of information. With the present traditional system, it would be difficult for us to integrate with the other traditional systems. We require special knowledge or understanding for using the existing monitoring systems. In the existing systems, the data is not displayed in the raw format.

2 Evaluation

2.1 Existing System

Earlier versions of the environmental monitoring system use wired sensors to transmit the data to the implemented standalone application workstation and it has limited connectivity options and data storage capacity where the data is stored in a local server based on the designed architecture.
Some of the challenges of using the existing environmental monitoring system are expensive to purchase, and install, requires regular maintenance to function properly, is complex to implement and is vulnerable to malicious actors.

3 Design & Analysis

Arduino UNO is the microcomputer which is the single processing unit for our environmental monitoring system. Multiple sensors interface with the microcontroller for processing and reading the data and the WIFI module transmit the processed data over the network to the cloud server.

I. Microcontroller

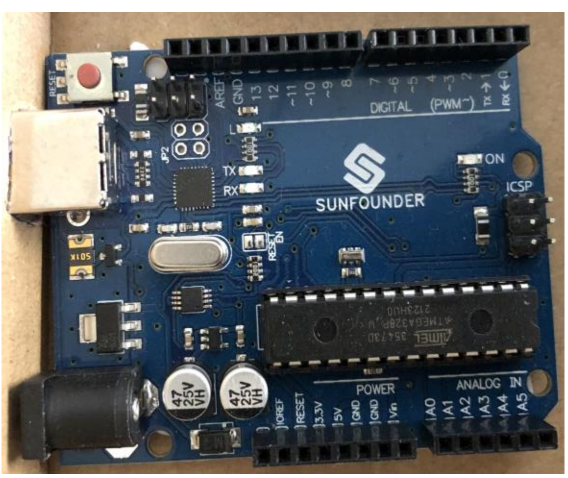

Fig 1. Arduino microcontroller board

One of the hardware components of the environmental monitoring system is the Arduino UNO microcontroller interfacing with the sensors to read the temperature, humidity, air quality and smoke and our system does not require any local storage which makes our selection for the Arduino UNO microcontroller. It is simple, low cost and robust. the below figure 1 depicts the picture of Arduino UNO microcontroller [1].

II. Sensors

DHT 11 is commonly used for sensing temperature and humidity it has a dedicated NTC to measure temperature and the output generated in serial data by an 8-bit microcontroller. The DHT11 sensor by default calibrated and easy to assemble with Arduino.

The DHT11 has long-term stability and high reliability, low cost and good quality with compact size and immediate response and output of the digital signal. Figure 2 depicts the picture of the DHT11 sensor [2].

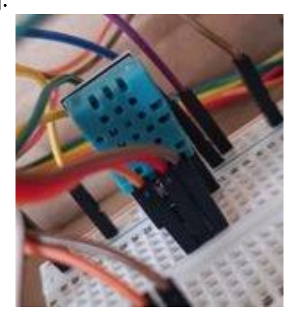

Fig 2. DHT11 sensor.

To measure the air quality MQ-135 gas sensor is used which will sense the wide range of gases such as carbon monoxide, Sulphur dioxide, ammonia and nitrogen oxides. The special features of MQ-135 are high sensitivity and fast response, long life and easy to implement. It has the operational voltage up to +5V and analog output voltage in the range of 0V to 5V [3]. The MQ-135 can be used for analog or digital sensor. The figure 3 depicts the picture of MQ-135 gas sensor.

Fig 3. MQ-135 gas sensor

MQ-2 sensor is widely used for detecting the gases such as methane, LPG, Butane and smoke and this sensor comes with the digital pin which makes operable without the microcontroller. The analog pin is used to measure the gas in PPM and analog pin is TTL driven which works with 5V [4]. it has the interface that makes it easy to assemble with the microcontroller. The figure 4 depicts the picture of MQ-2 sensor.

Fig 4. MQ-2 gas sensor.

III. WIFI Module

The information, which was extracted from the sensors such as DHT11, MQ-135, and MQ-2 needs to be uploaded over the network. ESP8266 help us to connect the microcontroller to the internet and the ThingSpeak cloud. The WIFI module has a low-cost microchip with a complete TCP/IP stack, compact size and powerful. The ESP8266 can be used as both an access point and a station and it can be programmed using Arduino IDE or through AT commands [5]. It generates an IP upon being connected to the WIFI through which it communicates to the internet. Figure 5 depicts the picture of the ESP8266 WIFI module.

Fig 5. ESP8266 WIFI module

IV. ThingSpeak Cloud Server

The ThingSpeak cloud server collects the data from ESP 8266 Wi-Fi module. This data is then transmitted to a web application using API. Figure 6 depicts the picture of the ThingSpeak.

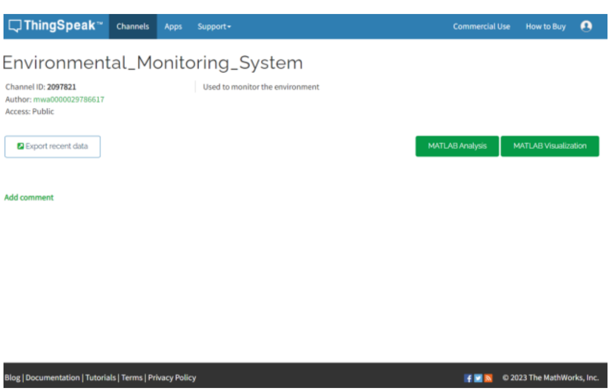

Fig 6. ThingSpeak

V. Hardware Block Diagram

Figure 7 depicts the block diagram for our environmental monitoring system which will show the system functionality where MQ-135, MQ-2, and DHT11 generates updated reading of air quality, smoke, humidity and temperature parallelly to the microcontroller and updates the data to the ThingSpeak cloud and Streamlit web application.

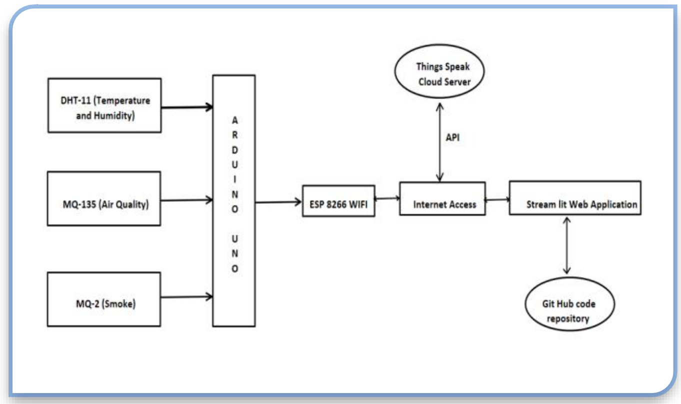

Fig 7. Block diagram of environmental monitoring system

4 Proposed System

The proposed system is a Real-time IOT - based Environmental Monitoring System that uses Arduino and multiple sensors to monitor different environmental parameters such as temperature, humidity, air quality and smoke. We are using DTH11 to detect temperature and humidity, MQ-135 to detect air quality and MQ-2 sensor to detect indoor smoke, while an ESP8266 Wi-Fi module transmits the collected data over the internet. This data is then uploaded to the ThingSpeak cloud server through an API every 15 seconds.

The data is then transmitted from ThingSpeak to the Ems-anytime web application using API with the help of Streamlit library and shown in interactive graphs and tiles. This system allows users to download the past 7 days data from the application itself and provides them with information on the last updated date and time. Alert messages will be shown in the application user interface when the smoke in the room exceeds a predefined threshold.

5 Implementation

Implementation is foremost important to achieve the success of advance or latest system, designing the project plan and concrete architecture helped us to finish the complete environmental monitoring system. The below points will describe how the project has been implemented.

1. Arduino UNO mini microcontroller and Arduino IDE for sensor data retrieval and establishing a successful connection

between the local machine and to upload the environmental monitoring system Arduino programming to the Arduino UNO.

2. Bread board acts as a medium for the integration of Arduino UNO microcontroller to the sensors and ESP module.
3. Python 3.7 - 3.11, install the python IDLE using the official python website. https://www.python.org/downloads/ and choose the appropriate version. we have chosen the python version 3.11. once it has been downloaded continue the installation process.
4. Download and install the visual studio code IDE using the official website. https://code.visualstudio.com/ and choose the latest version.
5. Download the Streamlit from the official website once the prerequisites have been installed from the following link. https://streamlit.io/
6. download and include the Arduino libraries: Software Serial and DTH11 in order to work with temperature and humidity values.
7. Download and import the Altair, pandas for graphs and data frames and datetime, pytz is used for timestamp conversion and time zone conversions
8. Download and import requests library for HTTP requests and responses.
9. Configuration of ESP module will be initiated by executing the commands such as "AT" to make the microcontroller a client-based workstation.
10. Software has been developed to read the data from DHT11, MQ-135 and MQ-2 and simultaneously uploaded to the ThingSpeak cloud analytics service.
11. With the help of the ThingSpeak channel API key we receive data from microcontroller and upload the data to the developed Streamlit web application.
12. Our web application communicates with the microcontroller with the help of ThingSpeak cloud. The live feed will be updated using one of the REST API method known as GET.
13. Web application will retrieve the data by sending JSON (Java Script Object Notation) GET request to the ThingSpeak along with the configured fields and channel ID parameters.
14. The JSON format will be stored in a data frame which then input to the Altair library to visualize the data over the web application.

6 Experimental Results

Fig 8 displays the complete design of environmental monitoring system which shows the integration and assembly of hardware components in a operating conductions. MQ-135, MQ-2, DHT11 and ESP8266 modules are connected to the Arduino UNO microcontroller. The web application connects to the ThingSpeak and visualize the generated data in the form of graphs. The fig 9 displays the hardware implementation of EMS system.

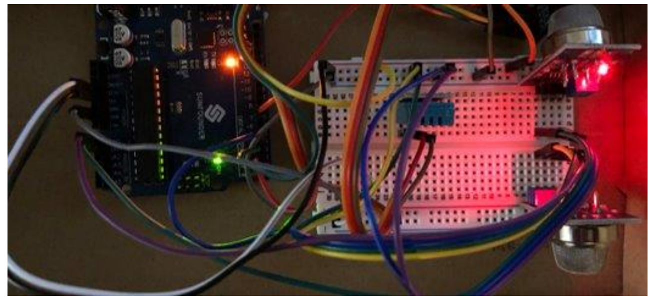

Fig 8. Hardware Implementation of EMS System

Table 1. will provide the details of the system hardware, display and cloud service.

System Component	Details
Sensors	DHT11, MQ-135, MQ-2
Network Connectivity	ESP8266 Module
Microcontroller	Arduino UNO
Cloud	ThingSpeak Cloud
User Interface	Web Application

Table 1. System Component Details

This section majorly concerns about the experimental outcomes of our environmental monitoring system.

A. Web Application User Interface

Figure 9 displays the sample screenshot for the environmental monitoring system web application.

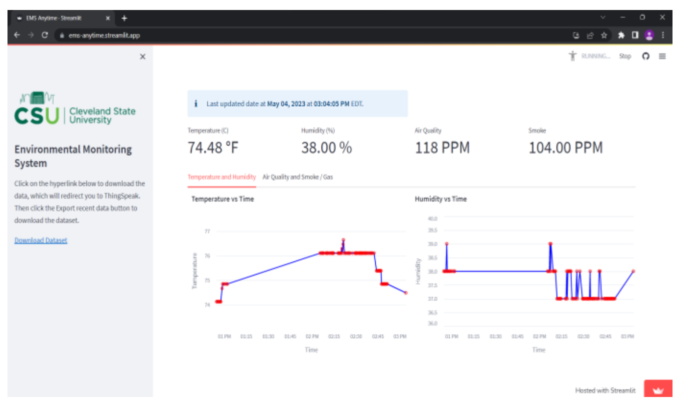

Fig 9. Environmental Monitoring System

In the EMS (Environmental monitoring system) user be able to download the updated data in the CSV, JSON, XML formats just by clicking on hyperlink which navigates to ThingSpeak. From there, user can download data. Also, user has the capability to download the separate field level data.

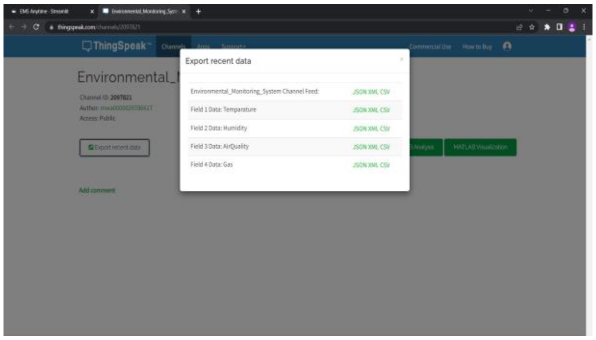

Fig 10. Download live data using ThingSpeak cloud.

B. Graphical record of Temperature monitoring

Figure 11 depicts the temperature monitoring over a time period. the graph is designed with temperature in y-axis and time in x-axis. due to the temperature drop there is a gradual decrement of the temperature value on the graph.

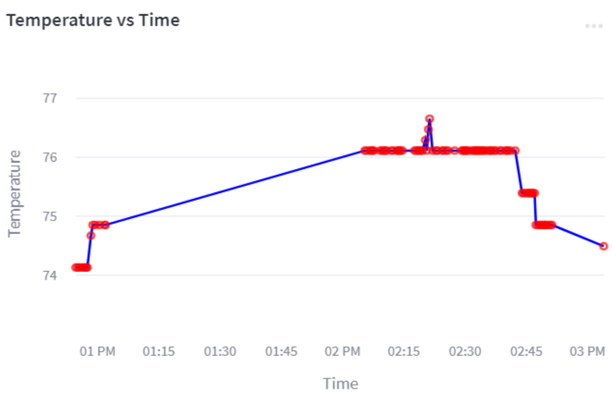

Fig 11. Temperature monitoring graph

Figure 12 depicts the humidity monitoring by having a humidity on y-axis and time period on x-axis. the live data displays that humidly were not constant on a given time period.

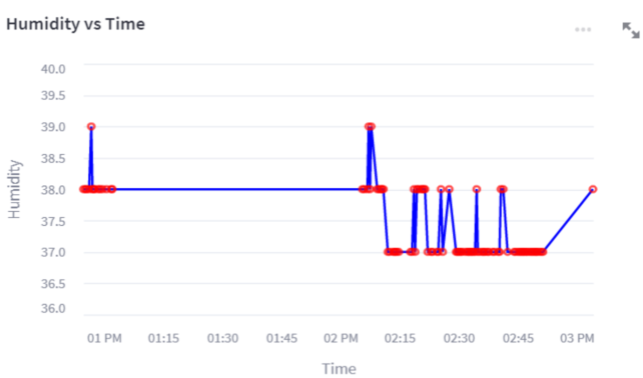

Fig 12. Humidity monitoring graph

Figure 13 depicts the air quality monitoring by having a AQ parameter on y-axis and time on x-axis. As the system were continuously monitoring the condition the air quality gradually increased and a depleting graph is shown.

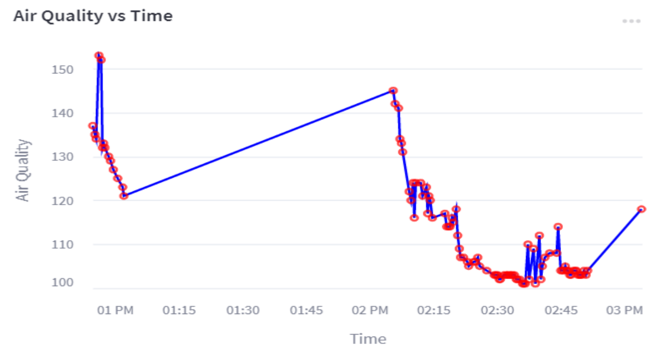

Fig 13. Air Quality (AQ) monitoring graph

Figure 14 depicts the smoke detection monitoring by having the smoke parameter on y-axis and time on x-axis. As the air quality has been restored in the previous graph the smoke has been decreased and a smoke depleting graph is shown.

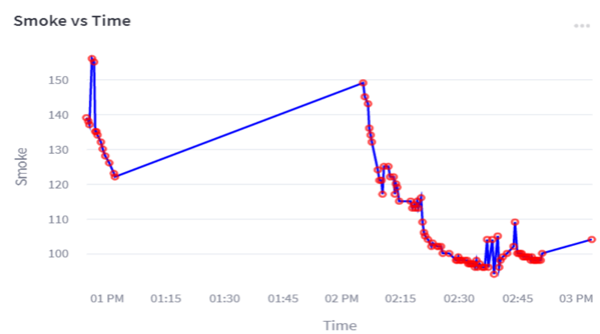

Fig 14. Smoke monitoring graph

TEAM MEMBERS ROLE

- Sri Harsha Mittapalli: Implementation of the web application’s entire user interface and coordinated with the team regarding integration of cloud server and hardware components.
- Pranay Kumar Gattu: Worked on ESP 8266 Wi-Fi sensor and Arduino coding. Also configured entire ThingSpeak configuration and connected cloud server to microcontroller using Wi-Fi sensor. In addition, helped the team in assembling the sensors together.
- Sai Charan Reddy Kumbham: Worked on MQ-2, MQ-135 sensors and Arduino coding. Assembled all the sensors and connected them together.
- Swathi Takkellapati: Worked on DHT-11 sensor and Arduino coding. Testing of the entire EMS system.

CONCLUSION

We encountered difficulties in coding and assembling the sensors, especially since using the Arduino IDE to program the sensors was a new experience for our team. Additionally, connecting the system to the ThingSpeak server and creating a real-time application using Streamlit that does not crash while ensuring code optimization was challenging. However, thanks to the overall contribution of our team, we divided the work into subtasks and integrated them, which allowed us to successfully complete the system. Finally, we hosted the web application on the internet using Streamlit community cloud.

ACKNOWLEDGMENT

The primary focus is to monitor and collect data on different environment conditions in real-time, ability to download and analyze data, display alert messages, and show interactive graphs in a user-friendly application. It is a low-cost, efficient, and reliable solution for monitoring and analyzing IOT sensor data. By using this system, you can get good safety measures and wellness.

REFERENCES

[1] "UNO R3." *Docs.Arduino.Cc*, docs.arduino.cc/hardware/uno-rev3.

[2] "DHT11–Temperature and Humidity Sensor." *Components101.Com*, 16 Jul. 2021, components101.com/sensors/dht11-temperature-sensor.

[3] "MQ-135 - Gas Sensor for Air Quality." *Components101.Com*, 27 Feb. 2018, components101.com/sensors/mq135-gas-sensor-for-air-quality.

[4] "MQ2 Gas Sensor." *Components101.Com*, 4 Jan. 2018, components101.com/sensors/mq2-gas-sensor.

[5] "ESP8266 - WiFi Module." *Components101.Com*, 20 Mar. 2018, components101.com/wireless/esp8266-pinout-configuration-features-datasheet.

[6] Tastan, M., & Gokozan, H. (2019, August 20). Real-Time Monitoring of Indoor Air Quality with Internet of Things-Based E-Nose. Www.Mdpi.com. https://www.mdpi.com/2076-3417/9/16/3435