Environmental Monitoring System Project Proposal

Final Project Title

Environmental Monitoring System

Team Members

Name	Contact
Zara Moinuddin	zmoinuddin@hawk.iit.edu
Sofia Yang	syang78@hawk.iit.edu

Problem Statement

Develop a monitoring and notification system for monitoring environmental air quality and pollution metrics.

Components of this project:

1. Creating a system to record sensor metrics of the temperature, humidity, and ppm levels of toxic gasses.
2. Develop a web application to visualize the trends of the logged metrics.
3. Create a notification system to send emails to team members when environmental conditions are lower than a predefined set of threshold values.

Extra Components (Dependent on time restrictions)

4. Develop a regression-based machine learning model to forecast future air quality based on sensor values.

Research/Investigation

Project Motivation and Design

The motivation behind the project is to develop a solution for monitoring and analyzing the effects of climate change and pollution in the air quality of an environment. Due to most toxic air pollutants appearing as fine particles, it is difficult to evaluate the air quality and level of pollutants in an area. The goal of this project is designed to collect valuable metrics, and alert when the level of pollutants in an environment is unsafe. Based on the metrics collected, it is possible to use analysis and machine-learning to forecast future trends, develop solutions or implement public policy to mitigate pollution.

Sensors Used

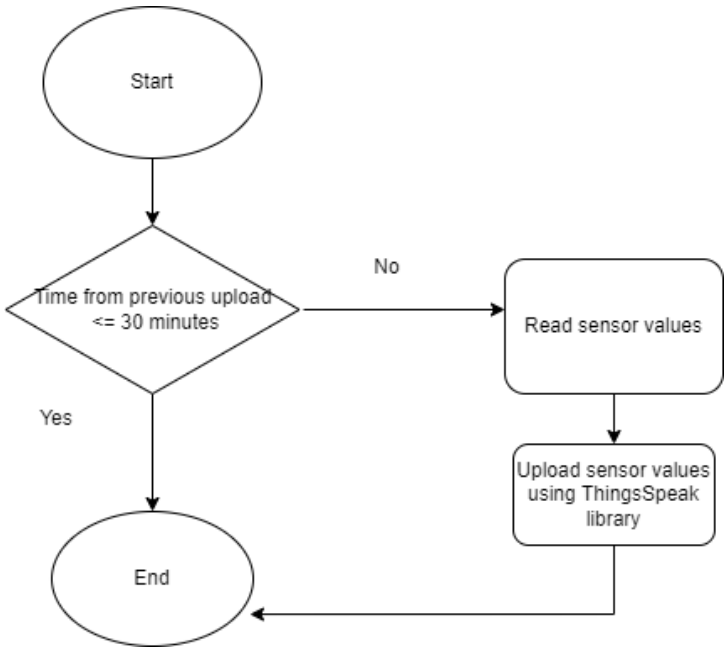
Due to scope limitations on the usable sensors for the project, it was decided to use the following sensors:

1. Air Quality Gas Sensor
   The air quality gas sensor will be used to read the ppm levels of gasses (such as ammonia, sulfides, nitrogen oxide, etc.) The sensor values recorded will be used to determine the amount of pollutants.
2. Temperature and Humidity Sensor Module
   The sensor module will be used to record the temperature and humidity levels of the environment. At high humidity levels, soluble air pollutants will dissolve, or undergo 'dry deposition', causing the air quality to decrease. At higher temperatures, the frequency of chemical reactions between pollutants and oxygen particles increases, causing the air quality to decrease as well.

Potential Solutions:

Collecting data with sensors

An Arduino sketch will be made with a loop to read the values from each sensor every 30 minutes, and upload the data to the ThingSpeak platform.
The flowchart is shown below.



Data Visualization and Notification System

Using the ThingsSpeak platform, a MATLAB script will be created to use a React condition. Based on a set of predefined sensor value thresholds, when the sensor values are in a certain range, the React will send an email alert.

Graphs and charts will be automatically generated by the ThingsSpeak platform.

Web Application Design

A Bootstrap website will be created with HTML/CSS/JS. The data visualizations from the ThingsSpeak platform will be embedded on the website with use of iframes.

Creating a Regression-Based ML Model

An Azure Data Factory will be provisioned to import JSON data from the ThingSpeak, as well as location air quality data. A regression-based machine learning model will be created in Azure ML Studio to predict air quality data based on sensor values, and the model will be trained using actual air quality data. A Logic App will be used to trigger the model to automatically run on receiving new data.

List of Tasks:

1. Breadboard the wire connections between the Arduino and sensors.
2. Write Arduino sketch to read sensor data and upload sensor values to ThingsSpeak
3. Setup ThingsSpeak for data visualization and notification system
4. Create webapp with graphs from ThingsSpeak

Extra Tasks:

1. Connect ThingsSpeak JSON data to automatically port to Azure Data Factory
2. Use Azure ML Studio to create a regression model based on data from Azure Data Factory
3. Create a logic app to automatically train and run the model.
4. Use Power BI to visualize data from the model (not sure)
5. Embed the Power BI graphs in the webapp.

List of parts:

Material/Equipment	**Obtained**	**Notes**
Breadboard	YES	
+5V Power supply for breadboard	NO	Maybe could use one of the labs on campus for power supply

Jumper wires	YES	Not sure what quantity will be needed
Resistors	YES	Need to see what resistance resistors will be needed
Capacitors	NO	
Temperature and humidity sensor module	NO	(From spreadsheet of available sensors)
Air Quality Detection Sensor (i.e MQ135)	NO	Not on spreadsheet of available sensors

Deliverables and Timeline

The guaranteed deliverables will be a notification system and web application containing the visualized sensor data. Depending on time constraints, it may be possible to develop a machine learning model to predict air quality based on sensor data.

The following table shows the project timeline:

Date	Expected Completion Goals
10/24	Go over proposal and software/hardware used. Stretch goal: Get started on Arduino sketch for reading sensor values.
10/31	(A) If sensors arrive: Start with the breadboarding process. (B) If sensors do not arrive: Complete rough Arduino sketch.
11/7	Continue/Start breadboarding process. Stretch goal: Test/debug Arduino sketch.
11/14	Setup ThingSpeak and notification system; create web application. Test system.
11/21 (No Lab)	Continue testing system. After testing the system, work on ML components.
11/28	Deliver final result.

Conclusion

The goal of this project is to develop a monitoring and notification system for monitoring environmental air quality and pollution metrics. By doing so, we hope to collect valuable insights for analysis and future solutions/public policies.
The project will be built on C code, and use the MATLAB platform ThingsSpeak.
An additional goal for the project would be to create a machine learning model to analyze the collected data. The machine learning model will be built on Python's pandas framework, and the infrastructure for deploying/running the model will be Microsoft's Azure platform.

References

1. *Arduino - humidity sensor*. Arduino Project Hub. (n.d.). Retrieved October 29, 2022, from https://create.arduino.cc/projecthub/sachendra003/arduino-humidity-sensor-288146
2. *MQ135 Air Quality Gas Sensor Module*. QuartzComponents. (n.d.). Retrieved October 29, 2022, from https://quartzcomponents.com/products/mq-135-air-quality-gas-sensor-module
3. Mathworks. (n.d.). *Mathworks/thingspeak-arduino: Thingspeak Communication Library for Arduino, ESP8266 and ESP32*. GitHub. Retrieved October 29, 2022, from https://github.com/mathworks/thingspeak-arduino
4. *IOT based air pollution monitoring system using Arduino*. IoT Based Air Pollution Monitoring System using Arduino & MQ135 Sensor. (n.d.). Retrieved October 29, 2022, from https://circuitdigest.com/microcontroller-projects/iot-air-pollution-monitoring-using-arduino
5. *Humidity sensors*. element14. (n.d.). Retrieved October 29, 2022, from https://in.element14.com/sensor-humidity-sensor-technology
6. Lavaa, A. (2021, October 23). *Temperature and humidity sensors: Ultimate Guide*. linquip. Retrieved October 29, 2022, from https://www.linquip.com/blog/temperature-and-humidity-sensors-an-ultimate/