University of the Philippines Manila

College of Arts and Sciences

Department of Physical Sciences and Mathematics

Low-Cost Air Quality Monitoring System for the Local Government Units of the Province of Rizal

A special problem in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Computer Science

Submitted by:

John Michael C. Pacis

June 2023

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled “Low-Cost Air Quality Monitoring System for the Local Government Units of the Province of Rizal” prepared and submitted by John Michael C. Pacis in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Avegail D. Carpio, M.Sc.
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Richard Bryann L. Chua, Ph. D.(*cand.*)		
2. Perlita E. Gasmen, M.Sc.(*cand.*)		
3. Ma. Sheila A. Magboo, Ph.D.(*cand.*)		
4. Vincent Peter C. Magboo, M.D., M.Sc.		
5. Marbert John C. Marasigan, M.Sc.(*cand.*)		
6. Geoffrey A. Solano, Ph.D.		

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Vio Jianu C. Mojica, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences
and Mathematics

Marie Josephine M. De Luna, Ph.D.
Chair
Department of Physical Sciences
and Mathematics

Maria Constancia O. Carrillo, Ph.D.
Dean
College of Arts and Sciences

Abstract

Every year, air pollution from fossil fuels causes millions of deaths, raises the risk of stroke, lung cancer, and asthma, and causes the nation's economy to suffer enormous losses. Along with the physical and economic effects of polluted air, Filipinos have long borne the burden of climate change's effects. The province of Rizal does not have any equipment that can be used to monitor, detect, and analyze Air Quality. The main goal is to design and develop a "Low-Cost Air Quality Monitoring System" that will track the air quality index in specific indoor and outdoor locations. The study successfully implemented an Air Quality Monitoring System prototype that incorporated gas sensors such as MQ-5, MQ-7, DHT11, and Dust Particle Sensor to detect, measure, and analyze pollutants like temperature, humidity, methane, carbon dioxide, and particulate matter.

Keywords: Climate change, Air Quality, Monitoring System, Methane, Particulate Matter

Contents

List of Figures

List of Tables

I. Introduction

A. Background of the Study

Our breathing environment has grown to be a crucial topic of research in the twenty-first century due to the continuous advances in quality of life. [1]

Benguet, Rizal, and Metro Manila are at the bottom of the air quality standings, with over 97 percent of their population exposed to some of the highest particulate matter (PM) concentrations in the entire country. [2] The data included in the study was crowdsourced from a range of continuous governmental monitoring sources, as well as outdoor air visual air quality monitors operated by private individuals and organizations.

Power plants, food manufacturing, small-scale chemical manufacturing, paper and packaging product production, and food manufacturing all contribute to the emissions of PM10, PM2.5, SOx, NOx, CO, and VOCs in Metro Manila. A further 2.3 million motor vehicles are registered, accounting for 27 percent of the nation's total in 2012. However, it is important to note that certain gases, such as SOx, NOx, and VOCs could not be directly detected and measured in this study due to the unavailability of specific sensors locally.

The presence of particulates, or microscopic solid or liquid aerosol particles, in the atmosphere might raise concerns about pollution. Particle pollution in the air can originate from both natural and man-made sources, including smoke from forest fires and other recreational activities, volcanic eruptions, car exhaust emissions, industrial emissions, soil, and road dust. It may move from hundreds to thousands of kilometers while hanging in the air, depending on its particular size, features, and environmental factors, and it may do so for a short period of time or forever. [3]

All airborne particles with dimensions of 100 micrometers or less are referred to as total suspended particles (TSP). Carbon monoxide (CO), an odorless, colorless gas, is extremely harmful when ingested because it slows down the body's ability to transfer oxygen. CO has the potential to be fatal at very high amounts. Particularly in metropolitan areas, vehicle emissions are the main source of this pollutant.

There is a complicated correlation between exposure and air pollution levels. But precise and straightforward information about their negative impact on health must be provided to the general public. The Air Quality Index (AQI), a set of environmental standards for measuring air quality, was developed to safeguard human health, safety, and general welfare.

Electric vehicles particularly tricycles and jeepneys have been introduced and promoted in cities in the Philippines in order to minimize mobile emissions. These cities include Pasig, Makati, and Manila. Separate traffic lanes have also been set aside for electric jeepneys (Makati). The city of Pasig encouraged residents to walk or bike to work and other places and provided pedestrian and cycling pathways in an effort to promote non-motorized transportation.

The creation of alliances between the Metro Manila city administration and various stakeholders, such as the national government, donors, and NGOs, has been essential in raising funds and carrying out important initiatives. A ”no exposed soil” guideline was put forth by the Department of Environment and Natural Resources (DENR), which employs best practices to reduce fugitive dust from building sites and unpaved roadways. The use of enclosures and coverings was mandated in order to reduce dust migration, as well as the use of bagged cement and on-site mixing of concrete and mortar [4].

The results of three scientific studies on the effects of air pollution on health in the Philippines have yielded very little proof. Nearly 600,000 cases of pneumonia, bronchitis, acute lower respiratory tract infections, and heart disorders were brought on by air pollution [4]. The Philippine Cancer Society attributes more than 9,000 premature deaths from all medical causes to air pollution. The cost of the effects of air pollution in Metro Manila was estimated to be USD 15.2 million for the years 2004 and 2007 combined.

B. Statement of the Problem

Every year, air pollution from fossil fuels causes millions of deaths, raises the risk of stroke, lung cancer, and asthma, and causes the nation's economy to suffer enormous losses. Along with the physical and economic effects of polluted air, Filipinos have long borne the burden of climate change's effects. According to a Greenpeace Southeast Asia and CREA report published in February 2020, air pollution from fossil fuels, primarily coal, oil, and gas, is thought to contribute to an estimated 27,000 premature deaths per year in the Philippines and can cost the nation up to 1.9 percent of GDP in economic losses each year.

At present, the Philippines only has a fleet of 55 government-owned PM2.5 monitoring stations. Twenty two are stationed in Metro Manila, while some are not strategically located in areas with high PM2.5 concentrations, such as major highways and coal plants. Bataan, which houses major megawatt-coal plants, has a lone PM2.5 monitor for the whole province.

And currently, there is no web-based system that the Municipality of Binangonan, Rizal is using. In addition, the current air monitoring equipment is large and difficult to maintain. There is also the need of an operator to make use of the machine. The province of Rizal does not have any equipment that can be used to monitor, detect, and analyze Air Quality.

C. Objectives of the Study

The study's main goal is to design and develop a ”Low-Cost Air Quality Monitoring System” that will track the air quality index in specific indoor and outdoor locations. To be more precise, it seeks to: (1) Track the air quality in a few chosen areas of the province of Rizal (2) Build a low-cost prototype using a microcontroller and develop a website that will compile data from the prototype; (3) Detect air pollutants by incorporating gas sensors like MQ7 and MQ5. The device will have the following functionalities:

1. Display the amount of methane
2. Display the amount of carbon dioxide
3. Display the amount of particulate matter (ppm)
4. Display the amount of humidity
5. Display the temperature

The user of the device can only:

1. Gather the results
2. View the results via dashboard
3. Generate graphs using the data
4. View weekly and monthly graphs

D. Significance of the Project

The study aims to develop a low-cost, portable air quality monitoring system for the Local Government Units since the current devices and systems are not yet available locally. Also, the study aims to address some of the problems which are limited resources and lack of funds that are evident in the municipalities. The monitoring system will make it simpler to gather data in real time, examine, record, and compile the information on the air quality. Additionally, giving municipalities or government organizations access to affordable air quality monitoring devices will enhance their quality of life and enable them to analyze, plan, and react to improve the quality of life.

The government's goal of stopping the growth of air pollution in the nation can be furthered by the findings of this study. This research might aid future studies in better understanding how to solve the nation's air pollution problem and can serve as a basis on how they will conduct their project or experiment in the future.

E. Scope and Limitations

1. The research study will focus on detecting Air Quality.
    (a) This study will use sensors such as MQ 5, MQ 7 that will only detect certain gasses: methane and carbon monoxide
    (b) Dust Particle Sensor that will only sense particulate matter.
2. The device will be incapable of purifying and cleaning the smoke
    (a) It will only detect the dangerous smoke like carbon gasses, hydrogen and methane present in the air.
3. Securing the device while deployed will be outside the scope of the study

F. Assumptions

1. The web application's user has received the necessary training to properly understand the results of the model.

2. The system will only be utilized as a decision support tool for the monitoring of Air Quality in the area

II. Review of Related Literature

Recent technological developments have given rise to innovations and opportunities like the Internet of Things (IoT), which enables embedded equipment linked to the Internet to communicate and be controlled remotely. For instance, researchers used IoT technology to keep an eye on a lake's water quality. [5]. It has been the subject of numerous research attempts to detect and monitor the AQI using wireless sensor networks or IoT. [6] These concluded projects measured various air pollutants with the use of mobile or stationary sensors.

There are various forms of pollution. According to Read and Digest, they are grouped based on the environmental component that they have an impact on [7]. It was also underlined that environmental destruction can be categorized according to the type of pollutant or the characteristics of the environment it affects. In addition, he divided pollution into the three (3) categories of air, water, and soil pollution.

Air pollution is the final of the three major types of pollution. Air pollution increased as a result of the rapid industrialization and urbanization of developing nations. Air pollution is described by the World Health Organization [8] as any chemical, physical, or biological substance that alters the atmospheric properties naturally and contaminates either the indoor or outdoor environment. Gasses including carbon monoxide, methane, and chlorofluorocarbons are the principal air pollutants. These gasses are frequently released by automobiles, commercial buildings, and home equipment like the refrigerator and air conditioner. Therefore, suburbanization causes an increase in traffic density, which in turn becomes a significant factor in air quality decline.

The level of pollution has increased with time due to a lot of factors like the increase in population, increased vehicle use, industrialization and urbanization which results in harmful effects on human wellbeing by directly affecting the health of the population exposed to it.

Additionally, for a sizable amount of time, experts have connected air pollution to a wide range of negative health outcomes. They include respiratory conditions including asthma and lung issues. Humans may develop cancer as a result of air pollution. The National Institute of Health [9] and World Health Organization [8] came to the same conclusion that airborne pollutants cause human cancer.

In 2019, Harsh Gupta et al. [10] In order to continuously monitor the levels of temperature, humidity, carbon monoxide, smoke, LPG, PM2.5, and PM10 in the atmosphere, harsh gupta created an IOT-based air pollution monitoring system. Their efforts have resulted in the development of a one-way link between an Android application and the free and open-source cloud platform Thing Speak. The Raspberry Pi and physical system are linked by a gateway. Once the firebase API was connected into an Android or iOS app, Firebase features like Analytics, Authentication, Storage, Messaging, Hosting, Crash reporting, Real-time Database, etc. were employed. The graphs were drawn in Thing Speak based on the sensor data that was received, and the same data was shown in a tabular fashion in an Android app.

In August 2019, Monika Singh et al. [11] suggested an Air Pollution Monitoring System. This system senses the many types of gasses present in the environment using an Arduino microcontroller linked with MQ135 and MQ6 gas sensors. It was then connected to the Wi-Fi module, which connects to the internet. An LCD is used to show the user the output, and a buzzer sounds a warning when the ppm exceeds a particular threshold. Industrial perimeter monitoring, indoor air quality monitoring, site selection for reference monitoring stations, and data dissemination were some of its applications.

According to [12], humanity has been the true culprit for the rise in air pollution among all other living things since it has forgotten the importance of sustainability as it transitions to an era focused on improvement. At some locations, the levels of pollution have gotten so bad that they are starting to harm our own physical and mental health. An IoT-based system for monitoring air pollution uses a MQ Series sensor connected to a Node MCU with an ESP8266 WLAN connection to transmit sensor readings to a Thing Speak cloud. Additional components of this work include a proper AI model to predict the level of air pollution and an anticipating model, which is essentially a subset of prescient showing. It becomes difficult to stop the dangerous gasses from polluting the pure air as the age of poisonous gasses from activities, cars, and other sources grows exponentially over time. In this research, a useful framework for air pollution monitoring is suggested. This system can be used to monitor pollution levels in a given area, find unusual air conditions, or do property inspections. The required framework will focus on monitoring the concentration of air pollutants with the use of a combination of wireless sensor networks and the Internet of Things. The investigation of air quality should be possible by figuring the air quality index (AQI)

According to [13], a number of reasons including population growth, increased automobile use, industrialization, and urbanization have all contributed to rising pollution levels, which have a negative impact on people's health and welfare. We will create an IOT-based air pollution monitoring system for this project in order to keep track of the air quality. This system will use the internet to monitor the air quality, and it will sound an alarm when the level of harmful gasses such as CO2, smoke, alcohol, benzene, and NH3 in the air exceeds a certain threshold. It will display the air components in PPM on the LCD and on the website so that we can easily monitor it. With this IOT project, you may use a PC or a mobile device to check the pollution level from anywhere.

According to [14] the ongoing rise in air and noise pollution is a grave issue in the modern world. Controlling and properly monitoring the situation is now essential in order to take the necessary action to bring it under control. An IOT-based system to track a region's Air Quality Index and Noise Intensity has been proposed in this research. The four elements that make up the suggested technology are the Anomaly Notification Module, the Sound Intensity Detection Module, the Cloud-based Monitoring Module, and the Air Quality Index Monitoring Module. First, the existence of the five criterion air pollutants is taken into account while calculating the Air Quality Index. The sound intensity is then measured using the appropriate sensor. The Raspberry Pi's Wi-fi module then facilitates the process of data acquisition, achieving the goal of periodic information analysis, while the Cloud-based Monitoring Module ensures data acquisition. In case of an undesirable state, the Anomaly Notification Module warns the user.

According to [10], the amount of pollution is rising quickly as a result of variables that include industry, urbanization, population growth, and automobile use, all of which can have an impact on human health. A web server and the Internet are utilized to monitor the air quality using an IOT-based air pollution monitoring system. When the amount of dangerous chemicals including CO2, smoking, alcohol, benzene, NH3, and NOx is high enough, it will sound an alarm when the air quality drops below a specified threshold. It will display the air quality in PPM on the LCD and on the website so that air pollution can be easily monitored.

In November 2018, [15] used the Internet of Things to measure the gas concentration using a variety of sensors that were seen through an Arduino serial monitor. This information is gathered in thing talk channels using an Ethernet shield, which is available in real time for processing. These analyzed data were presented graphically through thing speak. Next, an android app was used to examine the time-controlled findings after matlab analysis to determine the average pollution level. The air quality index value was also obtained using the location using the android app. Additionally, the health consequences were also shown in this app so that users could always be aware of the pollution levels.

In October 2017, [16] created a device for monitoring the air using an Arduino microcontroller. They utilized an Arduino to manage the entire operation and a MQ135 gas sensor to detect several types of harmful gasses. The output from the MQ135 gas sensor is provided in the form of voltage levels, which must be translated into PPM. The entire process was connected to the internet via a Wi-Fi module, and an LCD was utilized to display the results visually. The buzzer starts blaring and the LCD and webpage display ”Poor Air, Open Windows” when the PPM exceeds the limit. When the number is less than 1000 PPM, the LCD and webpage display ”Fresh Air.” If it increases to 2000 then the buzzer will keep beeping and the LCD and webpage will display “Danger! Move to Fresh Air”.

In 2017, [17] used a carbon sensor to find air pollutants or carbon particles. The sensor calculates the quantity of air contaminants that are present as well and outputs the information as an analog signal. Since the microcontroller only accepts digital input, the analog output of the sensor was transformed into digital form using the ADC and sent as an input to the device. These values are continuously displayed on the LCD. A switch pad was used to enter the crucial value. If the amount of pollutants in the air exceeds the critical value entered, the buzzer will beep and a notification will be sent by the microcontroller via the GPRS module to the webpage on the mobile phone. A notification was also issued to the website when the pollutant level exceeded the threshold. The mobile phone receives the signal from the modem and sends it to the server and internet. Analysis is done on the data that the server receives from the smartphone. It makes inferences from the data it has gathered and then disseminates those inferences online.

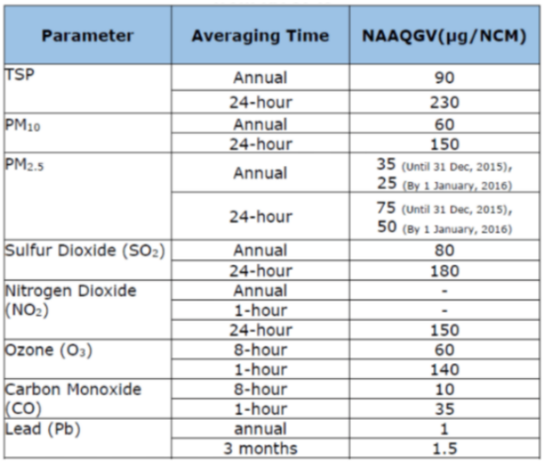
A system with Beaglebone interfaced with air pollution measuring sensors like carbon dioxide, carbon monoxide , and noise sensor was proposed by [18] in 2017. The Analog pin on the Beagle Bone Black, which receives input signals in the 0 v to 1.8 v range, was used to read the analog output from the sensor. Python SQL was used to upload sensor data to the Azure Cloud. The same data from the.CSV file is uploaded into the cloud database at the end of each day. An automated shell script was used to remove outdated data from the beagle bone. Data was stored in the Azure database from various sensors. This database data was downloaded in order to provide input for the machine learning service. With the aid of historical data, a machine learning service was employed to train the module. Sensor data that was retrieved by the BeagleBone Black was represented using Power BI.

III. Theoretical Framework

A. Air Quality Guidelines

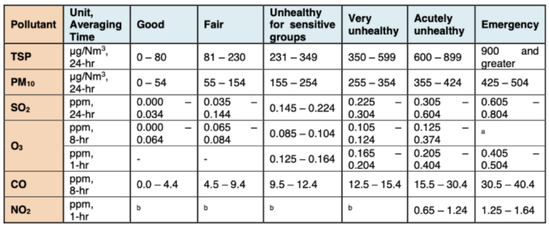
This project contributes in the direction of monitoring the quality of air. According to an interview from the department head of the Binangonan Municipal Environment and Natural Resources Office (2023), there is no current equipment, devices, and systems that are being used in monitoring and detecting the Air Quality in the municipality of Binangonan.

With the use of the Air Quality Monitoring System, this will determine whether the smoke emitted contains dangerous gasses that can contribute to further increase of air pollution. It also determines if the gas emitted is in excellent, passed, dangerous, and extremely dangerous form.



Parameter	Averaging Time	NAAQGV(μg/NCM)
TSP	Annual	90
	24-hour	230
PM_{10}	Annual	60
	24-hour	150
$PM_{2.5}$	Annual	35 (Until 31 Dec, 2015), 25 (By 1 January, 2016)
	24-hour	75 (Until 31 Dec, 2015), 50 (By 1 January, 2016)
Sulfur Dioxide (SO_2)	Annual	80
	24-hour	180
Nitrogen Dioxide (NO_2)	Annual	-
	1-hour	-
	24-hour	150
Ozone (O_3)	8-hour	60
	1-hour	140
Carbon Monoxide (CO)	8-hour	10
	1-hour	35
Lead (Pb)	annual	1
	3 months	1.5

Table 1: National Air Quality Guidelines from RA 8749



Pollutant	Unit, Averaging Time	Good	Fair	Unhealthy for sensitive groups	Very unhealthy	Acutely unhealthy	Emergency
TSP	μg/Nm³, 24-hr	0 – 80	81 – 230	231 – 349	350 – 599	600 – 899	900 and greater
PM_{10}	μg/Nm³, 24-hr	0 – 54	55 – 154	155 – 254	255 – 354	355 – 424	425 – 504
SO_2	ppm, 24-hr	0.000 – 0.034	0.035 – 0.144	0.145 – 0.224	0.225 – 0.304	0.305 – 0.604	0.605 – 0.804
O_3	ppm, 8-hr	0.000 – 0.064	0.065 – 0.084	0.085 – 0.104	0.105 – 0.124	0.125 – 0.374	a
	ppm, 1-hr	-	-	0.125 – 0.164	0.165 – 0.204	0.205 – 0.404	0.405 – 0.504
CO	ppm, 8-hr	0.0 – 4.4	4.5 – 9.4	9.5 – 12.4	12.5 – 15.4	15.5 – 30.4	30.5 – 40.4
NO_2	ppm, 1-hr	b	b	b	b	0.65 – 1.24	1.25 – 1.64

Table 2: Air Quality Breakpoints from RA 8749

The system proposed here is an air quality monitoring system that measures the surrounding air quality, dust particles and presence of suspended items in the air using sensors. These sensed parameters are shown and transmitted to a website so that proper action can be taken by the people in outdoor activities to reduce the possible health problems that could occur. The diagram describing the proposed system and its interconnection along with the stages involved is as described below:

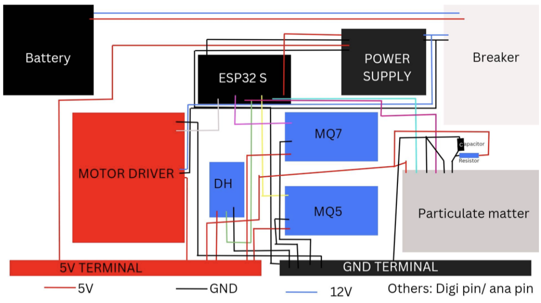


Figure 1: Diagram of the prototype

B. ESP32

This ESP-32 Development Board with 2.4 GHz ultra low power dual mode Wi-Fi and Bluetooth devkit is ideal for prototyping IoT or other devices which require wireless and internet connectivity, utilizing the powerful ESP32 microcontroller.

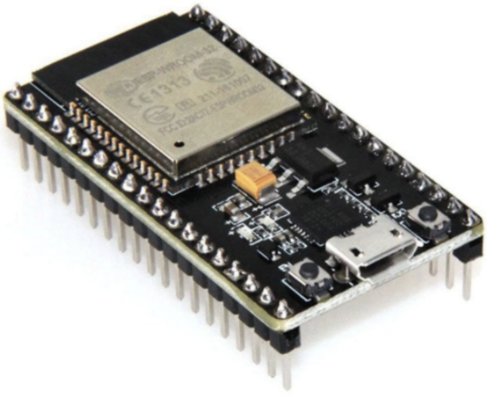


Figure 2: ESP32

C. MQ-7

The MQ-7 gas sensor has high sensitity to Carbon Monoxide. The sensor could be used to detect different gases containing CO, it is low cost and suitable for different applications.

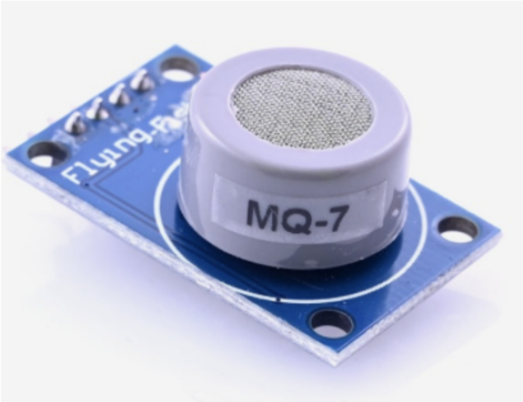


Figure 3: MQ-7

D. MQ-5

Commonly used in gas leak detection. These sensors are suitable for the detection of LPG, Natural Gas, Town Gas, Avoid Alcohol and Cooking Fumes and Cigarette Smoke.

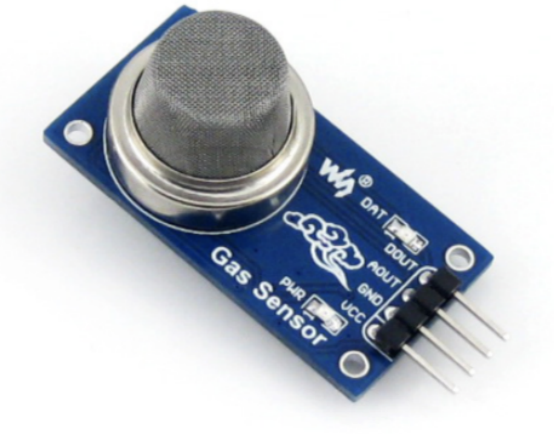


Figure 4: MQ-5

E. DHT-11

The DHT11 is a commonly used Temperature and humidity sensor that comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

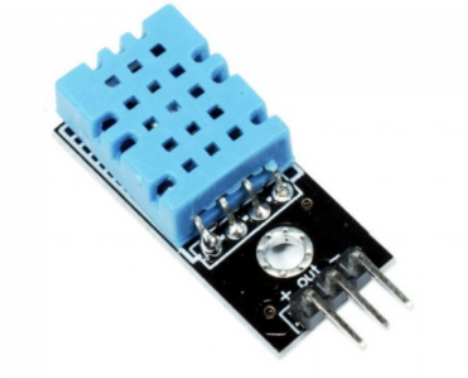


Figure 5: DHT-11

F. Motor Driver

This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator.

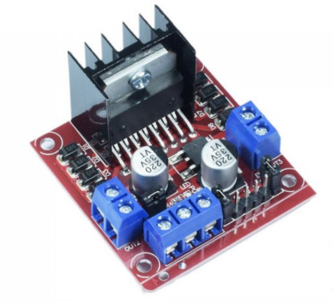


Figure 6: SMPS

G. Dust Particle Sensor

An infrared emitting diode and a phototransistor are diagonally arranged into this device, to allow it to detect the reflected light of dust in air. It is especially effective in detecting very fine particles like cigarette smoke, and is commonly used in air purifier systems.

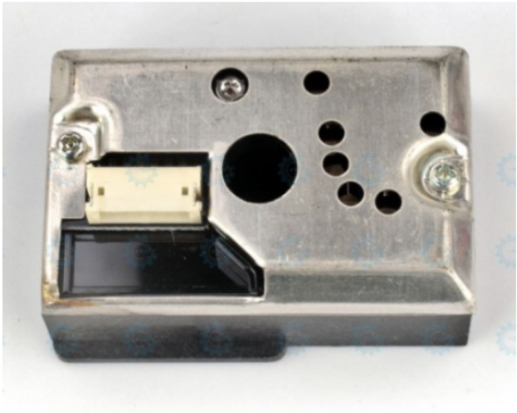


Figure 7: Dust Particle Sensor

H. Ambient Air Quality Model

Nowadays, air quality data can be easily accumulated by sensors around the world [19]. The concentration of pollutants monitored at monitoring stations changes with meteorological conditions, policies, pollution sources, human factors, etc. Evidence theory can well address the ambiguity of air quality and the uncertainty of environmental systems. For air quality evaluation, the main air pollutants affecting air quality are CO, PM10, NO2, PM2.5, SO2, O3. Air quality is not determined by a single air pollutant, but a combination of multiple air pollutants. Through the fusion pollution information through the improved evidence theory, a more accurate assessment of air quality can be obtained.

IV. Design and Implementation

The designed virtual system acquires, processes, archives and transmits the required signals, and visualizes reports that can be used by any person for easy observing of air quality. The developed system can be observed as consisting of three main parts – a set of detectors (integrated and gas sensors), data acquisition system and graphical application development environment.

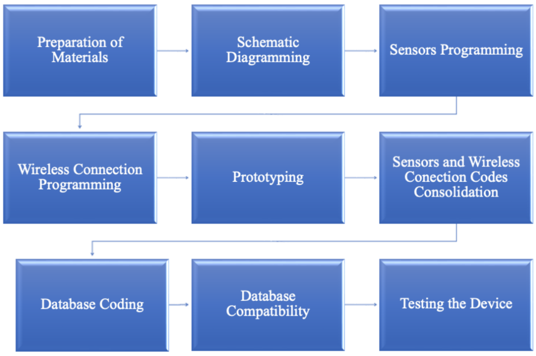


Figure 8: Design and Implementation Flowchart

A. Preparation of the Materials

The sensors, MQ-7, MQ-5, SMPS, and dust particle sensor was bought from e-Gizmo along with the micro controller unit, ESP32. After the collection of the different sensors, the researcher will assemble the materials that made the testing of the air quality possible.

B. Planning of codes

The researcher used the sample codes for MQ-5 Gas Sensor, MQ-7 Gas Sensor and Dust Particle Sensor in checking how the different sensors work and in planning the consolidation of the codes for the final output.

C. Prototyping of the device

The prototyping phase began with the selection and integration of necessary hardware components. Gas sensors such as MQ7 and MQ5 were carefully selected and incorporated into the prototype to detect pollutants like methane, carbon dioxide, and other harmful gases. These sensors were connected to the microcontroller board using appropriate wiring and connectors to establish the necessary electrical connections.

D. Database Coding

An account was created on the Firebase Google platform. The necessary Firebase libraries and dependencies were then installed in the Arduino IDE to establish a connection between the Arduino board and the Firebase database. The coding involved configuring the microcontroller to collect air quality data from the sensors and transmit it to the Firebase database. The collected data, including pollutant measurements, humidity, temperature, and timestamps, were formatted and packaged for transmission.

In addition to the Arduino integration, a web-based application was developed to interact with the Firebase database and display the air quality data in a user-friendly manner. This involved coding the frontend using web development technologies such as HTML, CSS, and JavaScript.

E. Database Compatibility

The researcher monitored the data and test the compatibility of the database and the device.

F. Context Free Diagram

From the diagram below, the user manages the sensor details for the monitoring system. The air quality sensors sends the air quality sensor data to the system. Next, the system will return dashboard info, air quality information, and an option to export as csv.

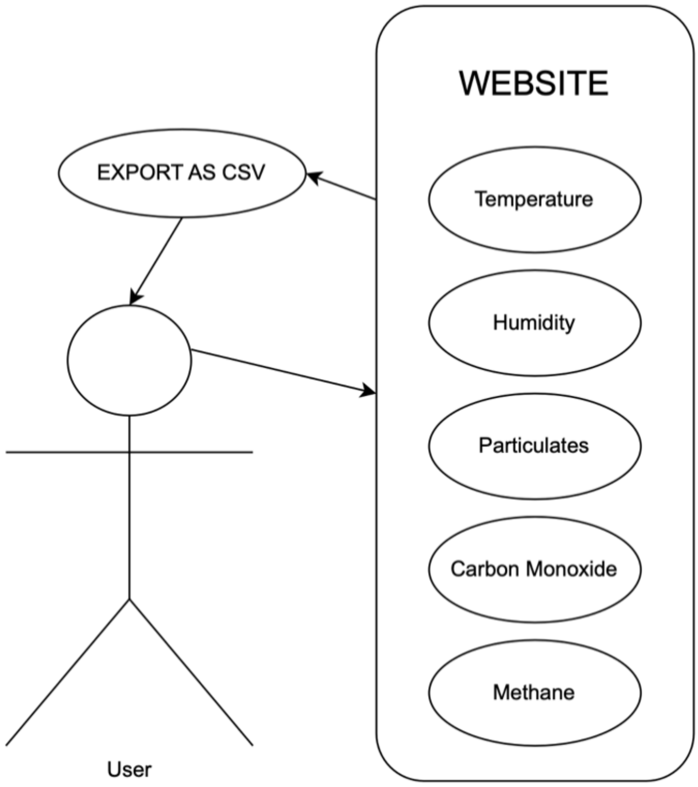


Figure 9: Diagram of the Monitoring System

G. Entity Relationship Diagram

Based on the diagram above, an entity relationship diagram (ERD) illustrates the system's information and entities relationship. ERD is composed of three things such as identifying and defining the entities, determining entity interaction and cardinality of the relationship.

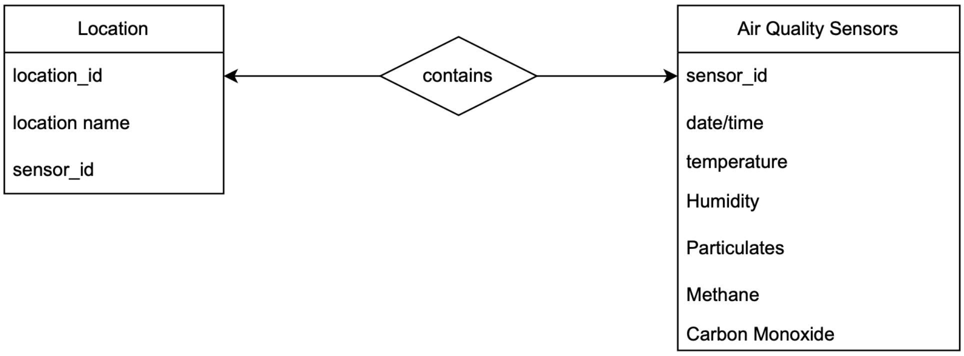


Figure 10: Entity Relationship Diagram of the Monitoring System

V. Results

This chapter aims to provide a comprehensive analysis of the system while providing the output of the Air Quality Monitoring.

A. Circuits and Devices

The prototype was developed by following these procedures:

1. Hardware Selection:

   (a) The researcher used an ESP32 microcontroller. The ESP32 is a powerful and popular choice for IoT projects due to its built-in Wi-Fi and Bluetooth capabilities.

   (b) The MQ5 and MQ7 gas sensors were used, which are commonly used for detecting gases like LPG, natural gas, carbon monoxide, and methane.

   (c) A power supply module was used to regulate the power supply for the system.

   (d) A DHT11 sensor to measure temperature and humidity.

2. Circuit Design:

   (a) A schematic diagram was created using a pen and paper.

   (b) Determined the necessary connections and pin assignments for each component based on the datasheets for details and compatibility.

3. Wiring and Connections:

   (a) Transfer the circuit into a fiber glass for a more permanent setup.

   (b) Make the necessary connections by finalizing the components onto the board.

Sensor Integration:

1. Connect the MQ5 and MQ7 gas sensors to the appropriate pins on the ESP32 microcontroller.
2. Connect the DHT11 sensor to the ESP32's digital input/output pins designated for data communication.
3. Follow the datasheets for each sensor to ensure correct wiring and connections.

Programming:

1. Write the code for the ESP32 microcontroller using Arduino IDE.
2. Utilize the libraries and functions provided by the ESP32 ecosystem to interface with the sensors, read their data, and control the system's functionality.
3. Implement data processing, communication protocols, and any additional features desired for the air quality monitoring system.

Testing and Debugging:

1. Upload the code into the ESP32 microcontroller and power on the prototype.
2. Monitor and verify the output on a display or through a serial communication interface to ensure the system is functioning as expected.
3. Debug any issues or errors encountered during testing by checking the code, connections, and sensor readings.

Calibration and Validation:

1. Calibrate the sensors (MQ5, MQ7, and DP) according to their respective calibration procedures and standards. Follow the instructions provided in the sensor datasheets.

2. Validate the accuracy and reliability of the measured air quality

The figures below show the circuit, prototype, and the official equipment of the Department of Environment and Natural Resources in the Provincial Capitol Grounds along with Engr. Carter, who oversees the operation.

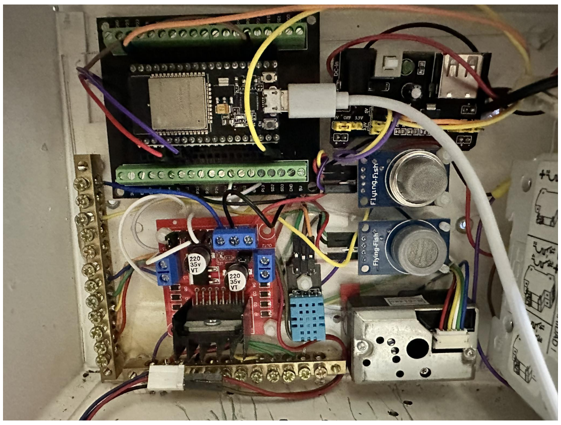


Figure 11: Air Quality Circuit

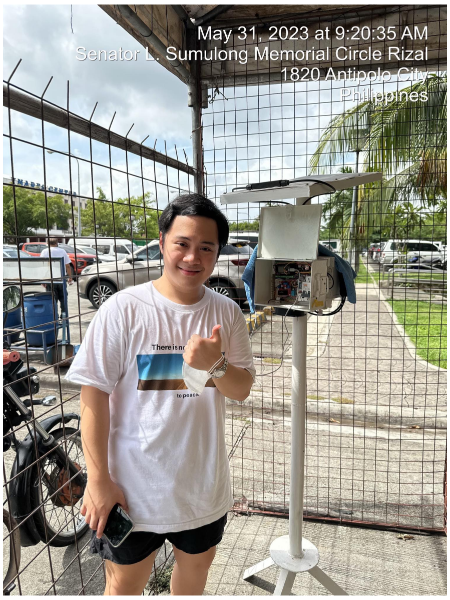

May 31, 2023 at 9:20:35 AM
Senator L. Sumulong Memorial Circle Rizal
1820 Antipolo City
Philippines

Figure 12: Air Quality Monitoring Device

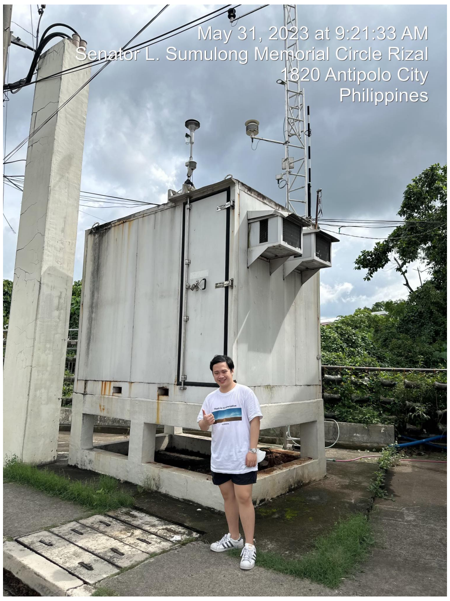

May 31, 2023 at 9:21:33 AM
Senator L. Sumulong Memorial Circle Rizal
1820 Antipolo City
Philippines

Figure 13: DENR Equipment inside the Container Van

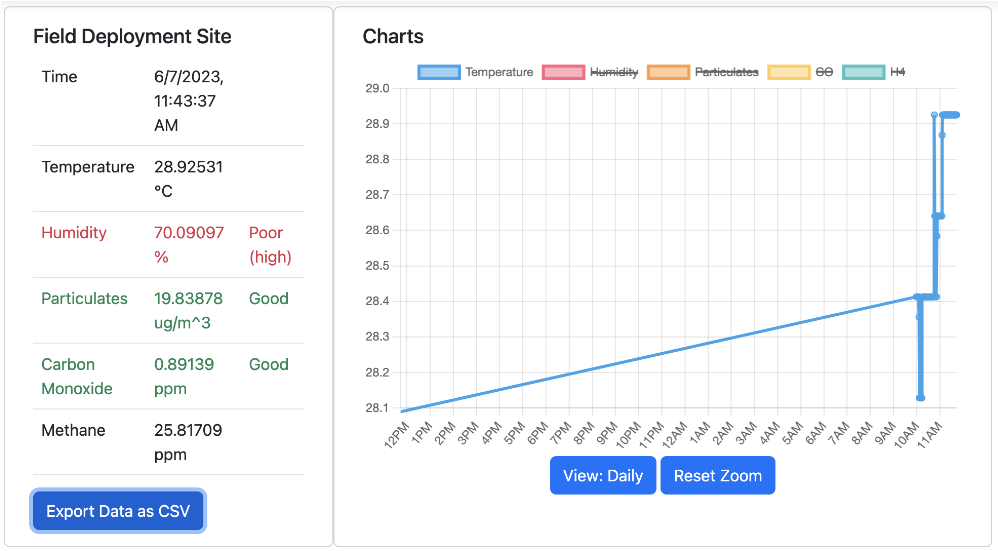
May 31, 2023 at 10:44:40 AM
Senator L. Sumulong Memorial Circle Rizal
1820 Antipolo City
Philippines
There is no alternative
to peace.
FRAGILE

Figure 14: Engineer Carter operating the Equipment

B. Overview of the System

The study presents a web-based monitoring system for the Air Quality in the LGUs. The system interface consists of a Monitoring page, and a downloadable csv report.

The figure below shows the functionalities of the system. On the left side, you can see the date and time, the parameters being measured, and a button to export the data as CSV. On the right side, it provides the graph of the toggled parameter. In this case, the temperature is shown on the graph.



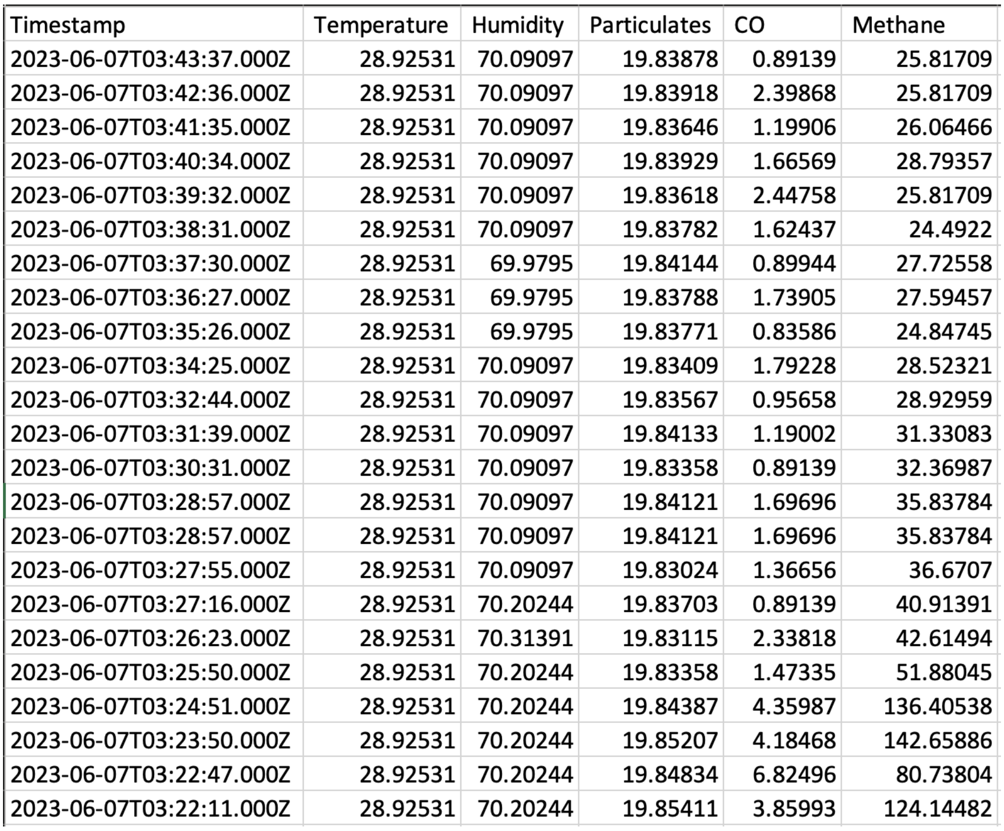
Field Deployment Site

Time	6/7/2023, 11:43:37 AM	
Temperature	28.92531 °C	
Humidity	70.09097 %	Poor (high)
Particulates	19.83878 ug/m^3	Good
Carbon Monoxide	0.89139 ppm	Good
Methane	25.81709 ppm	

Figure 15: Air Quality Website

C. CSV Report

The figure below shows the generated csv when exported. The data includes the timestamp of the monitoring, temperature, humidity, particulates, carbon monoxide, and methane.

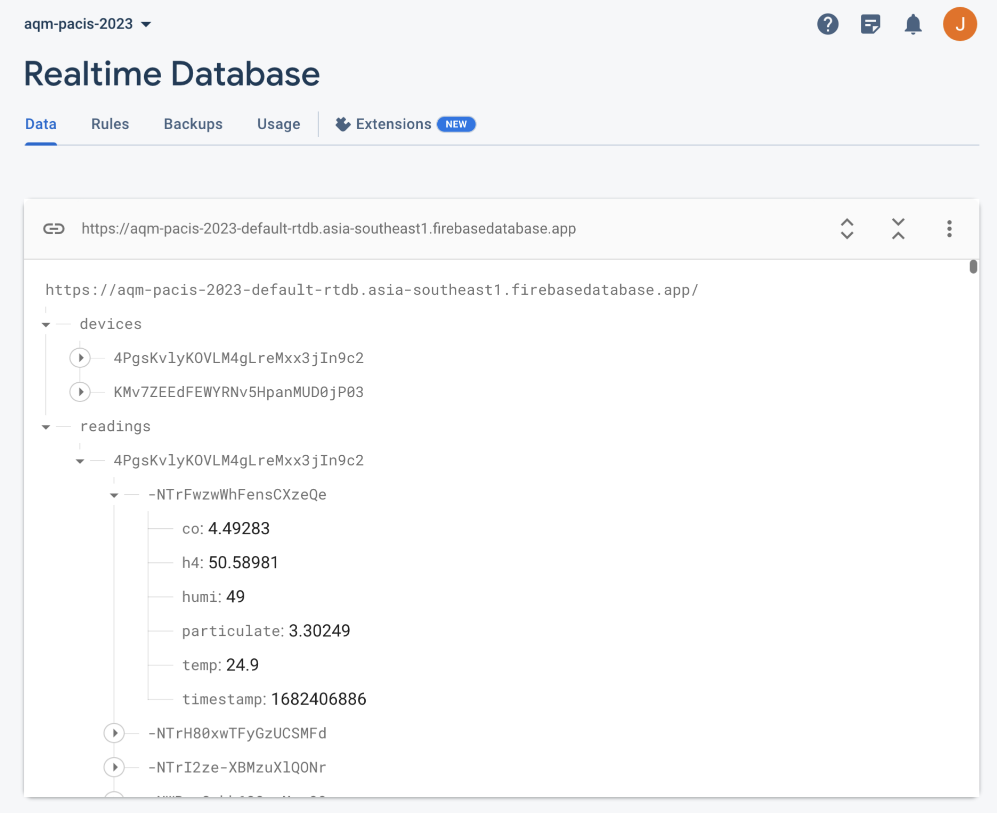


Timestamp	Temperature	Humidity	Particulates	CO	Methane
2023-06-07T03:43:37.000Z	28.92531	70.09097	19.83878	0.89139	25.81709
2023-06-07T03:42:36.000Z	28.92531	70.09097	19.83918	2.39868	25.81709
2023-06-07T03:41:35.000Z	28.92531	70.09097	19.83646	1.19906	26.06466
2023-06-07T03:40:34.000Z	28.92531	70.09097	19.83929	1.66569	28.79357
2023-06-07T03:39:32.000Z	28.92531	70.09097	19.83618	2.44758	25.81709
2023-06-07T03:38:31.000Z	28.92531	70.09097	19.83782	1.62437	24.4922
2023-06-07T03:37:30.000Z	28.92531	69.9795	19.84144	0.89944	27.72558
2023-06-07T03:36:27.000Z	28.92531	69.9795	19.83788	1.73905	27.59457
2023-06-07T03:35:26.000Z	28.92531	69.9795	19.83771	0.83586	24.84745
2023-06-07T03:34:25.000Z	28.92531	70.09097	19.83409	1.79228	28.52321
2023-06-07T03:32:44.000Z	28.92531	70.09097	19.83567	0.95658	28.92959
2023-06-07T03:31:39.000Z	28.92531	70.09097	19.84133	1.19002	31.33083
2023-06-07T03:30:31.000Z	28.92531	70.09097	19.83358	0.89139	32.36987
2023-06-07T03:28:57.000Z	28.92531	70.09097	19.84121	1.69696	35.83784
2023-06-07T03:28:57.000Z	28.92531	70.09097	19.84121	1.69696	35.83784
2023-06-07T03:27:55.000Z	28.92531	70.09097	19.83024	1.36656	36.6707
2023-06-07T03:27:16.000Z	28.92531	70.20244	19.83703	0.89139	40.91391
2023-06-07T03:26:23.000Z	28.92531	70.31391	19.83115	2.33818	42.61494
2023-06-07T03:25:50.000Z	28.92531	70.20244	19.83358	1.47335	51.88045
2023-06-07T03:24:51.000Z	28.92531	70.20244	19.84387	4.35987	136.40538
2023-06-07T03:23:50.000Z	28.92531	70.20244	19.85207	4.18468	142.65886
2023-06-07T03:22:47.000Z	28.92531	70.20244	19.84834	6.82496	80.73804
2023-06-07T03:22:11.000Z	28.92531	70.20244	19.85411	3.85993	124.14482

Figure 16: Air Quality CSV

D. Google Firebase

The figure below shows the firebase of the device. The devices folder shows two different hashes, indicating the 2 deployment sites in the website. The readings folder shows the timestamp and the parameters being collected by the device.



aqm-pacis-2023
J
Realtime Database
Data
Rules
Backups
Usage
Extensions
NEW
https://aqm-pacis-2023-default-rtdb.asia-southeast1.firebasedatabase.app

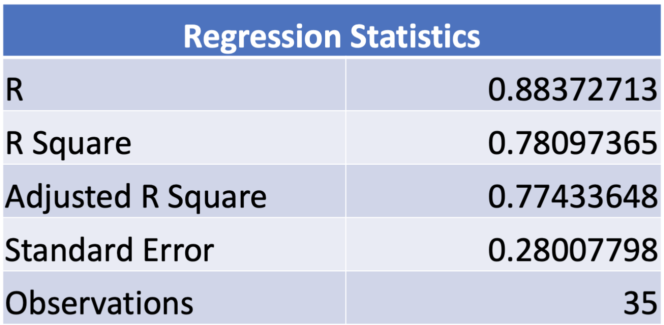
```
https://aqm-pacis-2023-default-rtdb.asia-southeast1.firebasedatabase.app/
devices
    4PgsKvlyKOVLM4gLreMxx3jIn9c2
    KMv7ZEEdFEWYRNv5HpanMUD0jP03
readings
    4PgsKvlyKOVLM4gLreMxx3jIn9c2
        -NTrFwzwWhFensCXzeQe
            co: 4.49283
            h4: 50.58981
            humi: 49
            particulate: 3.30249
            temp: 24.9
            timestamp: 1682406886
        -NTrH80xwTFyGzUCSMFd
        -NTrI2ze-XBMzuXlQONr
```

Figure 17: Air Quality Database

VI. Discussions

A. Temperature

This study attempts to compare the temperature values obtained from the DENR Instrument and this study's Air Quality Monitoring System. To evaluate the performance of the system, this study computed for the Pearson correlation coefficient (r) which was 0.8837. The coefficient suggests a strong positive linear relationship between values obtained from DENR and this system. Furthermore, the computed coefficient of determination (R^2) is 0.7810 which shows that 78.1 percent of the variance of values obtained from DENR is explained by the values obtained from the system.



Regression Statistics	
R	0.88372713
R Square	0.78097365
Adjusted R Square	0.77433648
Standard Error	0.28007798
Observations	35

Table 3: Regression Statistics for Temperature

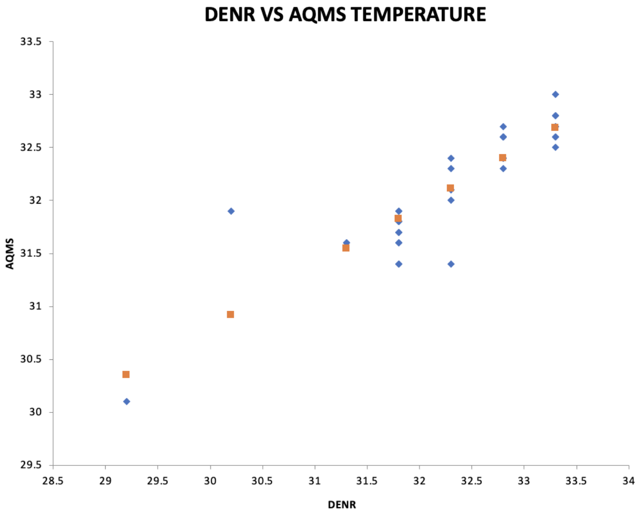
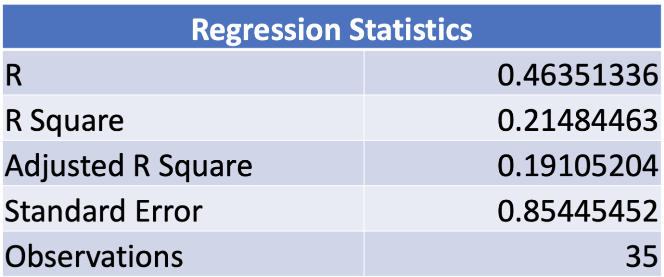


Figure 18: DENR VS AQMS (Temperature)

B. Humidity

This study attempts to compare the humidity values obtained from the DENR Instrument and this study's Air Quality Monitoring System. To evaluate the performance of the system, this study computed for the Pearson correlation coefficient (r) which was 0.4635. The coefficient suggests a moderate positive positive linear relationship between values obtained from DENR and this system. Furthermore, the computed coefficient of determination (R^2) is 0.2148 which shows that 21.5 percent of the variance of values obtained from DENR is explained by the values obtained from the system.



Regression Statistics	
R	0.46351336
R Square	0.21484463
Adjusted R Square	0.19105204
Standard Error	0.85445452
Observations	35

Table 4: Regression Statistics for Humidity

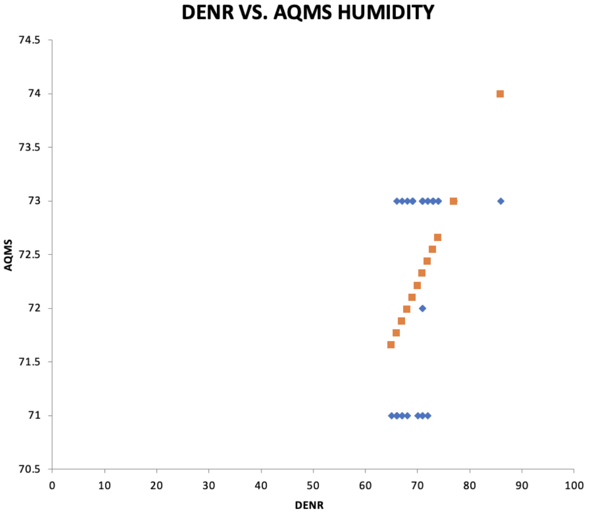
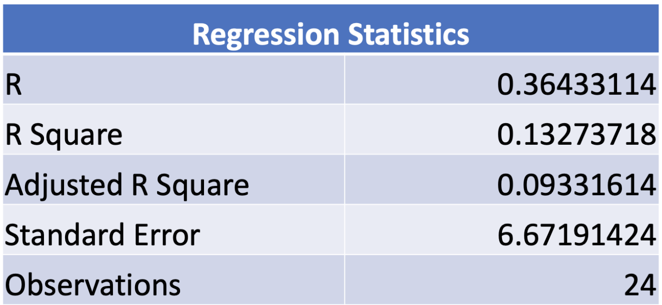


Figure 19: DENR VS AQMS (Humidity)

C. Particulates

This study attempts to compare the particulate values obtained from the DENR Instrument and this study's Air Quality Monitoring System. To evaluate the performance of the system, this study computed for the Pearson correlation coefficient (r) which was 0.3643. The coefficient suggests a moderate positive positive linear relationship between values obtained from DENR and this system. Furthermore, the computed coefficient of determination (R^2) is 0.1327 which shows that 21.5 percent of the variance of values obtained from DENR is explained by the values obtained from the system.



Regression Statistics	
R	0.36433114
R Square	0.13273718
Adjusted R Square	0.09331614
Standard Error	6.67191424
Observations	24

Table 5: Regression Statistics for Particulates

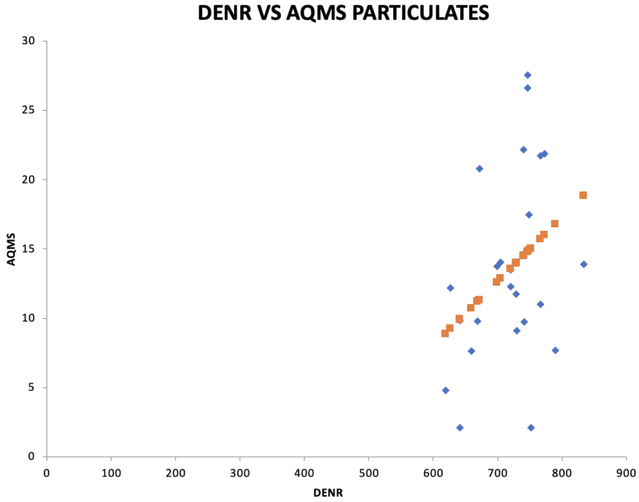


Figure 20: DENR VS AQMS (Particulates)

VII. Conclusions

In conclusion, this thesis study focused on the design and development of an low-cost air quality monitoring system using a microcontroller. The main goal was to create a device that could track air quality in both indoor and outdoor environments. Through the use of ESP32, gas sensors, and data processing techniques, several key objectives were achieved.

The study successfully implemented an Air Quality Monitoring System prototype that incorporated gas sensors such as MQ7 and MQ5 to detect pollutants like methane, carbon dioxide, and particulate matter. The system was able to measure and display these pollutants, as well as humidity and temperature, providing users with real-time information on the air quality in their surroundings.

The analysis provided insights into the relationships between the Temperature, Humidity, and Particulates parameters in the air quality monitoring system. The Temperature parameter exhibited a strong positive linear relationship, while the Humidity parameter showed a moderate positive linear relationship. On the other hand, the Particulates parameter had a weak positive linear relationship. The R-squared values indicated that the independent variables had varying degrees of explanatory power, with Temperature being the strongest predictor and Particulates the weakest. These findings can guide further analysis and decision-making in improving air quality monitoring and management.

The device's functionalities were designed to be user-friendly, allowing users to gather results, view them through a dashboard, generate weekly and monthly reports, and create graphs based on the collected data. This functionality enhances the usability and accessibility of the system, enabling users to make informed decisions and take necessary actions to improve air quality. By utilizing Arduino as the platform for this air quality monitoring system, the study demonstrated the flexibility and affordability of the technology. Arduino's open-source nature and wide range of compatible sensors and modules make it a suitable choice for developing innovative solutions in various domains, including environmental monitoring.

The successful implementation of this air quality monitoring system has significant implications for environmental monitoring and public health. It provides a low-cost and accessible solution for individuals, communities, and organizations to monitor air quality and make informed decisions about their environments.

The findings of this study have practical implications for air quality monitoring and management. The identified predictors can guide decision-making processes aimed at mitigating air pollution. For instance, policymakers and urban planners can utilize the results to design effective strategies to control emissions during periods of high temperature and humidity or to optimize urban layouts to enhance air circulation.

However, it is important to acknowledge the limitations of the study. The prototype developed in this study may have certain constraints regarding accuracy, calibration, and long-term stability, which need to be addressed in future research and development. Additionally, the study focused on a specific set of pollutants and environmental factors, and expanding the system to include a broader range of pollutants and parameters could further enhance its utility.

In conclusion, the air quality monitoring system developed in this thesis study using Arduino showcases the potential of affordable and accessible technology for environmental monitoring. By providing real-time data, user-friendly interfaces, and actionable insights, this system contributes to the broader goal of improving air quality and promoting public health. The study opens avenues for future research and development in this field, with possibilities for further enhancements and applications of Arduino-based environmental monitoring systems.

VIII. Recommendations

Based on the findings and outcomes of the thesis study on the air quality monitoring system using Arduino, several recommendations can be made to further enhance the system and its application. These recommendations are as follows:

1. Calibration and Validation: Conduct thorough calibration and validation of the gas sensors used in the system. This step is crucial to ensure accurate and reliable measurements of air pollutants. Collaborate with experts or relevant organizations to establish calibration protocols and compare sensor readings with reference instruments to improve the precision of the measurements.

2. Expansion of Sensor Array: Consider expanding the sensor array to include additional pollutants and environmental parameters. This could involve integrating sensors for volatile organic compounds (VOCs), nitrogen dioxide (NO2), ozone (O3), and other relevant pollutants. Additionally, incorporating environmental parameters such as noise levels and air pressure can provide a more comprehensive understanding of the overall air quality.

3. Data Analysis and Interpretation: Develop advanced data analysis techniques to derive meaningful insights from the collected data. Explore the use of machine learning algorithms to identify patterns, correlations, and predictive models. This can help in identifying pollutant sources, understanding their impacts, and providing proactive measures for air quality management.

4. Long-term Stability and Maintenance: Ensure the long-term stability and reliability of the system by implementing regular maintenance protocols. This includes sensor calibration, system upgrades, hardware upgrades, and monitoring of data transmission and storage. Establish partnerships or collaborations with local organizations or authorities responsible for air quality management to ensure continuous operation and support.

IX. Bibliography

[1] . M. T. Cincinelli, A., “Indoor air quality and health,” *International Journal of Environmental Research and Public Health*, vol. 4, no. 11, pp. 104–108, 2017.

[2] G. Philippines, “High ranking of philippine cities in the world air quality report,” 2019.

[3] D. o. E. Government of the Republic of the Philippines and N. R. E. M. B. (DENR-EMB), “Air quality.,” 2011.

[4] D. o. E. Government of the Republic of the Philippines and N. R. (DENR), “Denr proposes “no exposed soil” policy to reduce air pollution in metro manila.,” 2011.

[5] S. R. M.-S. J. K. F. V. X. e. a. Ansah, M., “Characterizing foliage influence on lorawan pathloss in a tropical vegetative environment,” *IET Wireless Sensor Systems*, vol. 10, no. 5, p. 198–207, 2020.

[6] N. E. Kaivonen, S., “Real-time air pollution monitoring with sensors on city bus,” *Digital Communications and Networks*, vol. 6, no. 1, p. 23, 2020.

[7] Read and Digest., “What are the different types of pollution?,” 2018.

[8] W. H. Organization, “Air pollution,” *International Research Journal of Engineering and Technology, IRJET.*, March 2018.

[9] N. I. of Health, “Air pollution,” *International Research Journal of Engineering and Technology, IRJET.*, March 2018.

[10] B. H. A. V. A. T. A. K. Harsh G., Dhananjay, “An iot bases monitoring system for smart cities,” 2019.

[11] P. K. C. Monika S., Misha K., “Iot based air pollution monitoring system using arduino,” *International Research Journal of Engineering and Technology, IRJET.*, 2019.

[12] P. S. Vijayakumar S., “Research on an iot based air pollution monitoring system on an iot based air pollution monitoring system,” 2019.

[13] R. P. D. B. P. U. Mahesh, P., “A survey paper on air pollution monitoring using iot monitoring using iot,” 2018.

[14] S. S. C. P. D. S. M. A. C. A. C. S. S. H. Saha, A., “A raspberry pi controlled cloud based air and sound monitoring pollution monitoring system with temperature and humidity sensing,” 2018.

[15] P. Yamunathangam, K. Pritheka, “Iot enabled air pollution monitoring and awareness creation system,” *International Journal of Recent Technology and Engineering, IRJET.*, 2018.

[16] S. T. A. S. Poonam P., Ritik G., “Iot based air pollution monitoring system using arduino,” *International Research Journal of Engineering and Technology IRJET*, 2017.

[17] A. N. A. Y. G. S. Rajat S., Pravin S., “Iot based air and sound pollution monitoring system,” *International Journal of Engineering Research Technology IJERT*, 2017.

[18] R. A. Nitin Sadashiv D., John S., “Iot based air pollution monitoring and predictor system on beagle bone black,” *International Conference on Nextgen Electronic Technologies, ICNET.*, 2017.

[19] Y. L. e. a. Chen, “Air quality data clustering using epls method,” vol. 36, pp. 225–232, 2017.

X. Appendix

A. Source Code

Air Quality Monitoring System - Prototype

```
#include <Arduino.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include ”DHT.h”
#include ”time.h”

#include ”secrets.h”

#define CO_PIN 34
#define CO_A 99.042
#define CO_B −1.518

#define H4_PIN 32
#define H4_A 1163.8
#define H4_B −3.874

#define DUST_PIN 35
#define DUST_A 0.046318
#define DUST_B −19.804
#define DUST_S 280
#define DUST_D 40

#define FAN_PIN 13
DHT dht(15, DHT11);
#define DUST_LED 14

#define TEMP_A 0.56930
#define TEMP_B 13.725

#define HUMI_A 0.11147
#define HUMI_B 64.406

FirebaseData fbdo;
FirebaseAuth fbau;
FirebaseConfig fbco;

String uid;
String path_timestamp;
String path_readings;

void setup() {
  Serial.begin(115200);

  // WiFi Initialization
  Serial.print(”Connecting to Wifi”);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.println(”.”);
  }
  Serial.println();
  Serial.print(”Connected to network with IP: ”);
  Serial.println(WiFi.localIP());

  // Configure Time
  configTime(0, 0, ”pool.ntp.org”);

  Serial.printf(”Firebase Client v%s\n\n”, FIREBASE_CLIENT_VERSION);
  fbco.host = FIRE_HOST;
  fbco.api_key = FIRE_AUTH;
  fbau.user.email = USER_NAME;
  fbau.user.password = USER_PASS;
  Firebase.begin(&fbco, &fbau);
  Firebase.reconnectWiFi(true);
  Firebase.setMaxRetry(fbdo, 3);
  Firebase.setMaxErrorQueue(fbdo, 30);
  Serial.print(”Getting UID”);
  while (fbau.token.uid == ””) {
    Serial.println(”.”);
  }
  Serial.println();
  uid = fbau.token.uid.c_str();
  path_timestamp = ”/devices/” + uid + ”/latest”;
  path_readings = ”/readings/” + uid;
  Serial.print(”UID: ”);
  Serial.println(uid);

  // Configure pins
  pinMode(CO_PIN, INPUT);
  pinMode(DUST_PIN, INPUT);
  pinMode(FAN_PIN, OUTPUT);
  pinMode(DUST_LED, OUTPUT);
  dht.begin();

  digitalWrite(DUST_LED, 1);
}

time_t now;
```

```
void loop() {

  digitalWrite(FAN_PIN, 1);

  delay(10000); // 10 secs to fan air in

  digitalWrite(FAN_PIN, 0);

  int coRead = analogRead(CO_PIN);
  float coVin = (coRead * 5.0 / 4095.0);
  float coRat = (5 - coVin) / coVin; // needed for all MQ-X
  float co = CO_A * pow(coRat, CO_B);

  int h4Read = analogRead(H4_PIN);
  float h4Vin = (h4Read * 5.0 / 4095.0);
  float h4Rat = (5 - h4Vin) / h4Vin; // needed for all MQ-X
  float h4 = H4_A * pow(h4Rat, H4_B);

  digitalWrite(DUST_LED, 0);
  delayMicroseconds(DUST_S);

  int dustRead = analogRead(DUST_PIN);
  float dust = DUST_A * ((dustRead * 5.0 / 4095.0)) - DUST_B;

  delayMicroseconds(DUST_D);
  digitalWrite(DUST_LED, 1);

  float temp = dht.readTemperature();
  float humi = dht.readHumidity();

  if(!isnan(temp))temp = TEMP_A*temp+TEMP_B;
  if(!isnan(humi))humi = HUMI_A*humi+HUMI_B;

  if(isnan(temp))temp=0;
  if(isnan(humi))humi=0;
  if(isnan(co))co=0;
  if(isnan(h4))h4=0;
  if(isnan(dust))dust=0;
  if(co<0)co=0;
  if(h4<0)h4=0;
  if(dust<0)dust=0;

  Serial.print(”CO: ”);
  Serial.print(co);

  Serial.print(” H4: ”);
  Serial.print(h4);

  Serial.print(” Dust: ”);
  Serial.print(dust);

  Serial.print(” Temp: ”);
  Serial.print(temp);

  Serial.print(” Humi: ”);
  Serial.println(humi);
  while(!Firebase.ready());
    time(&now);
    FirebaseJson fbjs;
    fbjs.add(”timestamp”, now);
    fbjs.add(”co”, co);
    fbjs.add(”h4”, h4);
    fbjs.add(”temp”, temp);
    fbjs.add(”humi”, humi);
    fbjs.add(”particulate”, dust);
    Firebase.pushJSON(fbdo, (path_readings).c_str(), fbjs);
    Serial.print(fbdo.errorReason().c_str());
    Firebase.setInt(fbdo, path_timestamp.c_str(), now);

  delay(50000); // 1min sampling rate
}
```

Air Quality Monitoring System - Dashboard

```
<!DOCTYPE html>
<html lang=”en”>
    <head>
        <meta charset=”utf-8”>
        <meta name=”viewport” content=”width=device-width, initial-scale=1”>
        <title>Air Quality Monitoring </title>
        <link href=”https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css” rel=”stylesheet” in
    </head>
    <body>

        <header class=”navbar navbar-expand-lg navbar-dark sticky-top bg-dark flex-md-nowrap p-0 shadow”>
            <a class=”navbar-brand me-0 px-3 fs-6” href=”index.html”>Air Quality Monitoring</a>

        </header>

        <main class=”col-md-9 mx-auto col-lg-10 px-md-4” id=”cardholder”>
            <div class=”pt-3”></div>
            <div class=’alert alert-danger’ role=’alert’ id=’alert-noid’ hidden>No Device ID requested. Maybe yo
            <div class=’alert alert-danger’ role=’alert’ id=’alert-invid’ hidden>Device ID invalid. Maybe you a
            <div class=”row”>
                <div class=”card col-sm-12 col-md-12 col-lg-4”>
                    <div class=”card-body”>
                        <h5 class=”card-title” id=”device-name”>Device Name</h5>
                        <!-- <p class=”card-text” id=”device-location”>Device Location</p>
                        <p class=”card-text”>Time: <span id=”device-time”>???</span></p>
                        <p class=”card-text”>Temperature: <span id=”device-temp”>???</span> C </p>
                        <p class=”card-text”>Humidity: <span id=”device-humi”>???</span>%</p>
                        <p class=”card-text”>Particulates: <span id=”device-particulate”>???</span> g/mˆ3</p>
                        <p class=”card-text”>Carbon Monoxide: <span id=”device-co”>???</span> ppm</p>
                        <p class=”card-text”>Methane: <span id=”device-h4”>???</span> ppm</p> -->

                        <table class=”table”>
                            <tr>
                                <td>Time</td>
                                <td><span id=”device-time”>???</span></td>
                                <td></td>
                            </tr>
                            <tr>
                                <td>Temperature</td>
                                <td><span id=”device-temp”>???</span>  C </td>
                                <td></td>
                            </tr>
                            <tr id=”device-humi-col”>
                                <td>Humidity</td>
                                <td><span id=”device-humi”>???</span> %</td>
                                <td id=”device-humi-int”></td>
                            </tr>
                            <tr id=”device-particulate-col”>
                                <td>Particulates</td>
                                <td><span id=”device-particulate”>???</span> ug/mˆ3</td>
                                <td id=”device-particulate-int”></td>
                            </tr>
                            <tr id=”device-co-col”>
                                <td>Carbon Monoxide</td>
                                <td><span id=”device-co”>???</span> ppm</td>
                                <td id=”device-co-int”></td>
                            </tr>
                            <tr>
                                <td>Methane</td>
                                <td><span id=”device-h4”>???</span> ppm</td>
                                <td></td>
                            </tr>
                        </table>


                        <p class=”card-text text-start”><a href=”#” id=”exporter” class=’btn btn-primary’>Expor
                    </div>
                </div>
                <div class=”card card col-sm-12 col-md-12 col-lg-8”>
                    <div class=”card-body”>
                        <h5 class=”card-title”>Charts</h5>

                        <canvas class=”card-text” id=”myChart”></canvas>
                        <p class=”card-text text-center”>
                            <a href=”#” id=”switcher” class=’btn btn-primary’>View: Daily</a>
                            <a href=”#” id=”zoomer” class=’btn btn-primary’>Reset Zoom</a>
                        </p>
                    </div>
                </div>
            </div>
        </main>

        <script src=”https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js” integrity=”sh
        <script src=”https://cdn.jsdelivr.net/npm/feather-icons@4.28.0/dist/feather.min.js” integrity=”sha384-u
        <script src=”https://cdn.jsdelivr.net/npm/chart.js”></script>
        <script src=”https://cdnjs.cloudflare.com/ajax/libs/hammer.js/2.0.8/hammer.min.js” integrity=”sha512-UX
        <script src=”https://cdn.jsdelivr.net/npm/moment@2.27.0”></script>
        <script src=”https://cdn.jsdelivr.net/npm/chartjs-adapter-moment@0.1.1”></script>
        <script src=” https://cdn.jsdelivr.net/npm/chartjs-plugin-zoom@2.0.1/dist/chartjs-plugin-zoom.min.js ”>
        <script type=”module”>
            import { initializeApp } from ’https://www.gstatic.com/firebasejs/9.20.0/firebase-app.js’
            import { getAuth } from ’https://www.gstatic.com/firebasejs/9.20.0/firebase-auth.js’
            import { getDatabase, ref, onValue, child, get, query, limitToLast, orderByChild } from ’https://www

            const firebaseConfig = {
                databaseURL: ”https://aqm-pacis-2023-default-rtdb.asia-southeast1.firebasedatabase.app”,
            };
```

```
//Chart.register(zoomPlugin)

const chart = new Chart(
    document.getElementById("myChart"),
    {
        type: 'line',
        data: {
            labels: [],
            datasets: [
                {
                label: "Temperature",
                data: [],
                },
                {
                label: "Humidity",
                data: [],
                hidden: true,
                },
                {
                label: "Particulates",
                data: [],
                hidden: true,
                },
                {
                label: "CO",
                data: [],
                hidden: true,
                },
                {
                label: "H4",
                data: [],
                hidden: true,
                },
            ]
        },
        options: {
            scales: {
                x: {
                    type: 'time',
                },
            },
            aspectRatio: 1.5,
            plugins: {
                zoom: {
                    zoom: {
                        wheel: {
                            enabled: true,
                        },
                        pinch: {
                            enabled: true
                        },
                        mode: 'xy',
                    },
                    pan: {
                        enabled: true,
                        mode: 'xy',
                    },
                },
            },
        }
    }
)

const app = initializeApp(firebaseConfig);
const database = getDatabase(app);
const device = new URLSearchParams(window.location.search).get('device')

let last_timestamp = 0
let scale = 0 // 0 = daily, 1 = weekly, 2 = yearly

if(device == null) {
    document.getElementById("alert-noid").hidden = false
}
else {
    const rInfo = ref(database, "devices/" + device)
    get(rInfo).then((snapshot)=>{
        const info = snapshot.val()
        document.getElementById("device-name").innerHTML = info.name
        // document.getElementById("device-location").innerHTML = info.location
    })

    const rReadings = query(ref(database, "readings/" + device), orderByChild('timestamp'))
    onValue(rReadings, (snapshot) => {
        const snapval = snapshot.val()
        const keys = Object.keys(snapval)
        const lastReading = snapval[keys[keys.length-1]]
        document.getElementById("device-time").innerHTML = new Date(lastReading.timestamp * 1000).t
        document.getElementById("device-temp").innerHTML = lastReading.temp
        document.getElementById("device-humi").innerHTML = lastReading.humi
        document.getElementById("device-particulate").innerHTML = lastReading.particulate
        document.getElementById("device-co").innerHTML = lastReading.co
        document.getElementById("device-h4").innerHTML = lastReading.h4

        if(lastReading.particulate < 80){
            document.getElementById("device-particulate-col").className = "text-success"
            document.getElementById("device-particulate-int").innerHTML = "Good"
        }
        else if(lastReading.particulate < 230){
            document.getElementById("device-particulate-col").className = "text-success"
            document.getElementById("device-particulate-int").innerHTML = "Fair"
        }
        else if(lastReading.particulate < 350){
            document.getElementById("device-particulate-col").className = "text-warning"
            document.getElementById("device-particulate-int").innerHTML = "Unhealthy"
        }
        else if(lastReading.particulate < 600){
            document.getElementById("device-particulate-col").className = "text-warning"
            document.getElementById("device-particulate-int").innerHTML = "Very Unhealthy"
        }
        else if(lastReading.particulate < 900){
            document.getElementById("device-particulate-col").className = "text-danger"
            document.getElementById("device-particulate-int").innerHTML = "Acutely Unhealthy"
        }
        else {
            document.getElementById("device-particulate-col").className = "text-danger"
            document.getElementById("device-particulate-int").innerHTML = "Emergency"
        }

        if(lastReading.co < 4.5){
            document.getElementById("device-co-col").className = "text-success"
            document.getElementById("device-co-int").innerHTML = "Good"
        }
        else if(lastReading.co < 9.5){
            document.getElementById("device-co-col").className = "text-success"
            document.getElementById("device-co-int").innerHTML = "Fair"
        }
        else if(lastReading.co < 12.5){
            document.getElementById("device-co-col").className = "text-warning"
            document.getElementById("device-co-int").innerHTML = "Unhealthy"
        }
        else if(lastReading.co < 15.5){
            document.getElementById("device-co-col").className = "text-warning"
            document.getElementById("device-co-int").innerHTML = "Very Unhealthy"
        }
        else if(lastReading.co < 30.5){
            document.getElementById("device-co-col").className = "text-danger"
            document.getElementById("device-co-int").innerHTML = "Acutely Unhealthy"
        }
        else {
            document.getElementById("device-co-col").className = "text-danger"
            document.getElementById("device-co-int").innerHTML = "Emergency"
        }

        if(lastReading.humi < 25){
            document.getElementById("device-humi-col").className = "text-danger"
            document.getElementById("device-humi-int").innerHTML = "Poor (low)"
        }
        else if(lastReading.humi < 30){
            document.getElementById("device-humi-col").className = "text-warning"
            document.getElementById("device-humi-int").innerHTML = "Fair (low)"
        }
        else if(lastReading.humi < 60){
            document.getElementById("device-humi-col").className = "text-success"
            document.getElementById("device-humi-int").innerHTML = "Good"
        }
        else if(lastReading.humi < 70){
            document.getElementById("device-humi-col").className = "text-warning"
            document.getElementById("device-humi-int").innerHTML = "Fair (high)"
        }
        else {
            document.getElementById("device-humi-col").className = "text-danger"
            document.getElementById("device-humi-int").innerHTML = "Poor (high)"
        }

        for(let i=last_timestamp; i<keys.length; i++) {
            chart.data.labels.push(snapval[keys[i]].timestamp * 1000)
            chart.data.datasets[0].data.push(snapval[keys[i]].temp)
            chart.data.datasets[1].data.push(snapval[keys[i]].humi)
            chart.data.datasets[2].data.push(snapval[keys[i]].particulate)
            chart.data.datasets[3].data.push(snapval[keys[i]].co)
            chart.data.datasets[4].data.push(snapval[keys[i]].h4)
        }

        // Object.keys(snapval).forEach(key => {
        //     if (snapval[key].timestamp > last_timestamp) {
        //         chart.data.labels.push(snapval[key].timestamp * 1000)
        //         chart.data.datasets[0].data.push(snapval[key].temp)
        //         chart.data.datasets[1].data.push(snapval[key].humi)
        //         chart.data.datasets[2].data.push(snapval[key].particulate)
        //         chart.data.datasets[3].data.push(snapval[key].co)
        //         chart.data.datasets[4].data.push(snapval[key].h4)
        //     }
        // })

        if(scale==0) chart.options.scales.x.min = lastReading.timestamp * 1000 - 86400000
        if(scale==1) chart.options.scales.x.min = lastReading.timestamp * 1000 - 86400000 * 7
        if(scale==2) chart.options.scales.x.min = lastReading.timestamp * 1000 - 86400000 * 30

        last_timestamp = Object.keys(snapval).length-1

        chart.update()
    })
}

function exportCSV(){
    let csvdata = "Timestamp, Temperature, Humidity, Particulates, CO, Methane\n"
    for(let i=chart.data.labels.length-1; i >= 0; i--){
```

```
                    const date = new Date(chart.data.labels[i]).toISOString()
                    csvdata += date + ","
                    csvdata += chart.data.datasets[0].data[i] + ","
                    csvdata += chart.data.datasets[1].data[i] + ","
                    csvdata += chart.data.datasets[2].data[i] + ","
                    csvdata += chart.data.datasets[3].data[i] + ","
                    csvdata += chart.data.datasets[4].data[i] + "\n"
                }
                const blob = new File([csvdata], "data.csv", { type: 'text/csv;charset=utf-8,' })
                const objUrl = URL.createObjectURL(blob)
                window.location.assign(objUrl);
                URL.revokeObjectURL(objUrl);
            }

            document.getElementById("exporter").onclick = exportCSV

            function setScale(){
                scale = (scale + 1) % 4
                if (scale == 0) document.getElementById("switcher").innerHTML = "View: Daily"
                if (scale == 1) document.getElementById("switcher").innerHTML = "View: Weekly"
                if (scale == 2) document.getElementById("switcher").innerHTML = "View: Monthly"
                if (scale == 3) document.getElementById("switcher").innerHTML = "View: All"

                if(scale==0) chart.options.scales.x.min = chart.data.labels[chart.data.labels.length - 1] - 864
                if(scale==1) chart.options.scales.x.min = chart.data.labels[chart.data.labels.length - 1] - 864
                if(scale==2) chart.options.scales.x.min = chart.data.labels[chart.data.labels.length - 1] - 864
                if(scale==3) chart.options.scales.x.min = chart.data.labels[0]

                chart.update()
            }

            document.getElementById("switcher").onclick = setScale

            function resetZoom(){
                chart.resetZoom()

                chart.update()
            }

            document.getElementById("zoomer").onclick = resetZoom

        </script>
    </body>
</html>
```

Air Quality Monitoring System - Dashboard

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Air Quality Monitoring </title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" i
    </head>
    <body>

        <header class="navbar navbar-expand-lg navbar-dark sticky-top bg-dark flex-md-nowrap p-0 shadow">
            <a class="navbar-brand me-0 px-3 fs-6" href="#">Air Quality Monitoring</a>


        </header>

        <main class="col-md-9 mx-auto col-lg-10 px-md-4" id="cardholder">
            <div class="pt-3"></div>

        </main>

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="s
        <script src="https://cdn.jsdelivr.net/npm/feather-icons@4.28.0/dist/feather.min.js" integrity="sha384-u
        <script type="module">
            import { initializeApp } from 'https://www.gstatic.com/firebasejs/9.20.0/firebase-app.js'
            import { getAuth } from 'https://www.gstatic.com/firebasejs/9.20.0/firebase-auth.js'
            import { getDatabase, ref, onValue, child, get } from 'https://www.gstatic.com/firebasejs/9.20.0/fi

            const firebaseConfig = {
                databaseURL: "https://aqm-pacis-2023-default-rtdb.asia-southeast1.firebasedatabase.app",
            };

            const app = initializeApp(firebaseConfig);

            const database = getDatabase(app);

            get(ref(database, `devices`)).then((snapshot) => {
                if (snapshot.exists()) {
                    const data = snapshot.val()
                    console.log(data);
                    const cardholder = document.getElementById("cardholder")
                    for(const device of Object.keys(data)){
                        if(data[device].debug) { continue }
                        console.log(device)
                        let card = ""
                        card += "<div class='card'>"
                        card += "<div class='card-body'>"
                        card += "<h5 class='card-title'>" + data[device].name + "</h5>"
                        card += "<p class='card-text'>" + data[device].location + "</p>"
                        card += "<a href='viewer.html?device=" + device + "' class='btn btn-primary'>View Data<
                        card += "</div>"
                        card += "</div>"
                        cardholder.innerHTML += card
                    }
                    for(const device of Object.keys(data)){
                        if(!data[device].debug) { continue }
                        console.log(device)
                        let card = ""
                        card += "<div class='card'>"
                        card += "<div class='card-body'>"
                        card += "<h5 class='card-title'>" + data[device].name + "</h5>"
                        card += "<p class='card-text'>" + data[device].location + "</p>"
                        card += "<a href='viewer.html?device=" + device + "' class='btn btn-primary'>View Data<
                        card += "</div>"
                        card += "</div>"
                        cardholder.innerHTML += card
                    }
                } else {
                    console.log("No data available");
                }
                }).catch((error) => {
                console.error(error);
            });


        </script>
    </body>
</html>
```

XI. Acknowledgment

I would like to express my deepest gratitude and appreciation to all the individuals who have supported and contributed to the completion of this thesis. Without their invaluable assistance, guidance, and encouragement, this work would not have been possible.

First and foremost, I would like to thank my supervisor, Avegail D. Carpio, M.Sc, for her unwavering support, expertise, and insightful guidance throughout this research journey. Their patience, encouragement, and commitment to excellence have been instrumental in shaping this thesis and enhancing its quality.

I am indebted to the faculty members of the Department of Physical Sciences and Mathematics (DPSM), whose teachings and knowledge have broadened my horizons and enriched my understanding of the subject matter. Their passion for research and dedication to fostering an intellectually stimulating environment have been an immense source of inspiration.

I extend my sincere appreciation to the members of the panel, especially our professor Dr. Vincent Peter C. Magboo, for their valuable insights, critical evaluation, and suggestions that significantly contributed to the improvement of this thesis. Their expertise and commitment to academic excellence have played a vital role in shaping this research endeavor.

I am grateful to the staff and administrators of the Department of Physical Sciences and Mathematics (DPSM) for their administrative support and assistance throughout my academic journey. Their professionalism and efficiency have eased many of the logistical challenges I encountered along the way.

I would like to express my heartfelt thanks to my family, Isidro M. Pacis Jr. and Cristina C. Pacis, for their unwavering love, encouragement, and belief in my abilities. Their constant support and understanding during the highs and lows of this research endeavor have been a source of strength and motivation.

I am deeply grateful to my colleagues,classmates, and friends for their stimulating discussions, valuable insights, and friendship. Their diverse perspectives

and intellectual exchange have significantly enriched my research experience.

Last but not least, I would like to acknowledge the financial support provided by the local government of Binangonan, Rizal. Their generous funding enabled me to conduct this research and present its findings.

In conclusion, the completion of this thesis would not have been possible without the support, guidance, and contributions of the individuals mentioned above. I am sincerely thankful for their immeasurable help and encouragement throughout this academic pursuit.