PARTICULATE MATTER SENSOR NODE

By

David Young (daviday2)

Mahip Deora (mdeora2)

Zachary Plumley (plumley3)

Final Report for ECE 445, Senior Design, Fall 2021

TA: Josephine Melia

8 December 2021

Project No. 12

Abstract

The Particulate Matter Sensor Node is a device that collects weather data, specifically particulate matter data, and logs it in an internal storage device. It utilizes a solar panel and rechargeable lithium ion battery to operate remotely. The node contains a control panel which enables users to change the data collection frequency and check the device's status. Additionally, a website was built to parse any collected data and display it using graphs. The sensor node was shown to accurately collect weather data for a month without an internet connection or external power source in varying weather conditions.

Contents

1. Introduction

1.1 Problem and Solution Overview

Particulate matter (PM) is a common air pollutant that comes from a variety of sources including fossil fueled power plants, combustion engines, industrial sites, and construction sites[1] to name a few. It is composed of microscopic solid and liquid particles such as smoke, dust, soot, salts, acids, and metals. PM emissions generally fit into 3 categories: PM1.0, PM2.5, and PM10. The number following PM indicates the particle's size in micrometers. For instance, PM2.5 is composed of particles that are between 1.0 and 2.5 micrometers. PM emissions, particularly particles that are smaller than 10 micrometers, have been linked to an array of health complications[2]. These health issues include but are not limited to nonfatal heart attacks, decreased lung function, increased respiratory symptoms, such as irritation of the airways, coughing or difficulty breathing, and premature death[3].

The EPA currently documents PM emissions; however, their data isn't comprehensive and lacks granularity[4]. There are several areas in the United States in which no PM data is being measured within a radius of 100 miles[4]. This is largely because current weather stations require a wireless network, such as the MIADS Weather Data stream, to transmit data[5]. Therefore, it is difficult to deploy these stations near PM sources that lack power or internet. Our project aims to collect data for these currently unreachable locations by being mobile, autonomous, and affordable. To do this, we created a device that measures temperature, humidity, wind speed, wind direction, and PM 1.0/2.5/10 data without an external power source or internet connection. In addition, our device is cost efficient such that it can be used at scale by the EPA and researchers. We included a solar panel, rechargeable lithium ion battery, and a microSD card so our system can operate remotely for an extended period of time. At the start of our project, we considered using long range wireless technologies like LoRa instead of a microSD card; however, LoRa has a maximum transmission radius of 10 miles which would limit the utility of our device. Once PM data is collected by the device, it can be uploaded to a web dashboard and viewed in a user-friendly UI. The dashboard allows users to visualize their data in the form of charts and graphs. It also allows users to view data that was collected by other users. These features mean that our system is mobile and scalable which will help the EPA fill in missing data and regulate PM emission sources.

1.2 High Level Requirements

1. Our system is able to accurately collect air data, specifically PM data, wind speed, wind direction, temperature, and humidity, within 10% accuracy. Accuracy was measured by comparing collected data to data from local weather stations.
2. Our system is able to collect and write PM data at a frequency of one sample every five minutes for thirty days using a solar panel and rechargeable battery to operate without human oversight.
3. Our system is able to store PM data in a microSD card and output the PM data to a web UI and MySQL database. The web UI will refresh every time it detects new data from the MySQL database.

2 Design

Our design, shown in Figure 1 below, consists of four subsystems: power supply, data processing, sensor array, and web UI. The power supply subsystem provides the device with a 3.3V, 5V, and 8V power supply. The sensor subsystem collects weather data and transmits it to the data processing subsystem. The data processing subsystem analyzes and stores that data in the SD card. The data stored on the SD card can be uploaded to the website, which parses and creates visualizations of the collected data. The power supply, data processing, and sensor array subsystems are housed inside of a weather proof enclosure that allows for accurate data collection as well as protection from the elements.

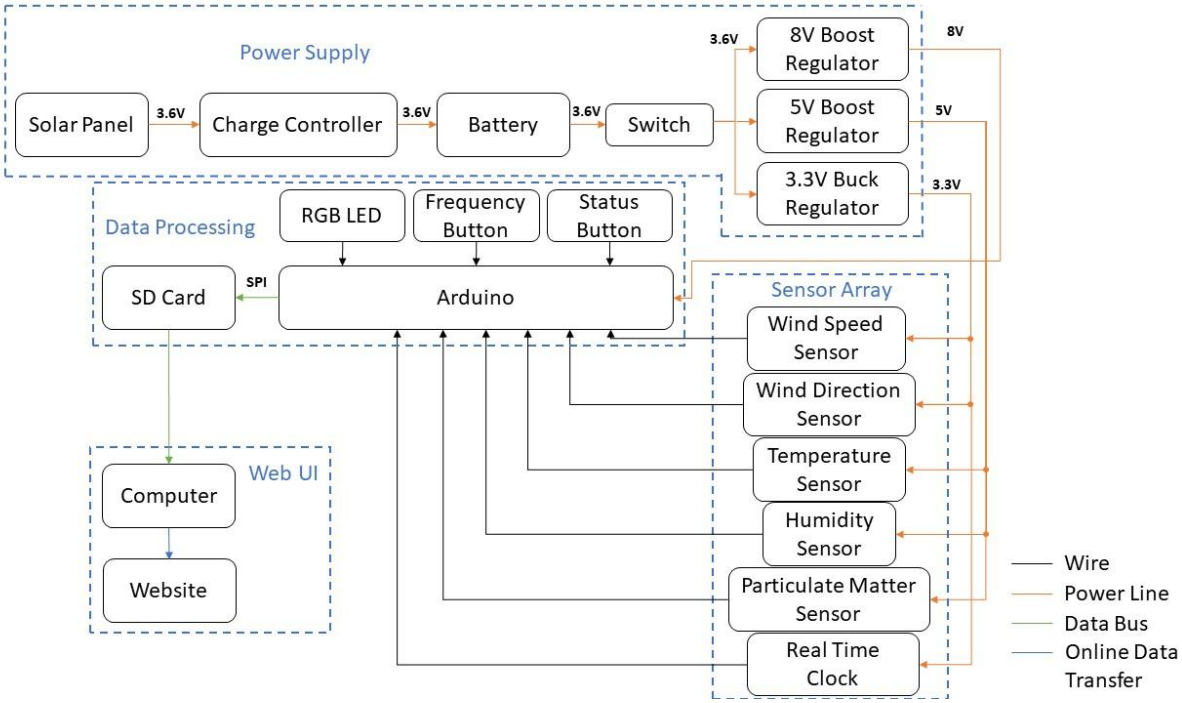

Figure 1. A block diagram showing components and modules

2.1 Physical Design

Figure 2 shows our completed prototype. The wind sensor and the solar panel are exposed on the outside of our enclosure while the rest of our components are housed inside. Our enclosure has vents in the top compartment so air can flow over our PM and temperature and humidity sensors; this is necessary to get accurate readings. These vents are recessed into the side to prevent rain from damaging our sensors. The top of our enclosure is slanted; this sets the solar panel at the optimal tilt angle for year round energy collection and prevents rain and snow from accumulating on top of our device. The yellow panel on the front of our enclosure is removable so the user can access the microSD card, control buttons, status LED, and internal sensors.

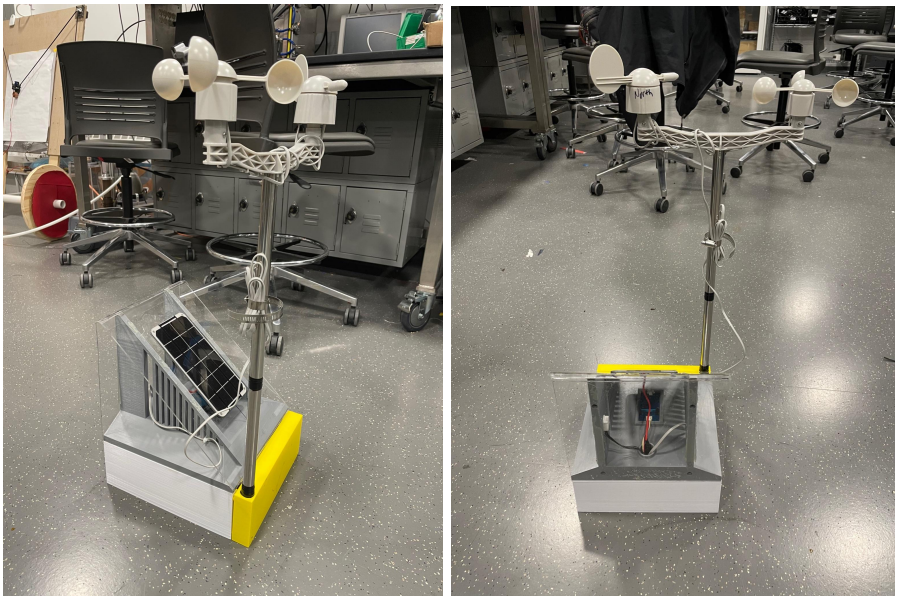

Figure 2. Shows our system from a physical level

2.2 Power Supply

The power supply unit produces, stores, and distributes electricity. A solar panel generates electricity and charges a battery using a charge controller. The battery pack feeds through a power switch into three voltage regulators which supply power to all of our components.

2.2.1 Solar Panel

The solar panel charges our battery pack to significantly extend our device's battery life. We calculated that our 3.5 watt 19% efficient solar panel could produce 168 Wh/month in Washington in the winter. Washington was chosen because it is the least sunny state[6]. These calculations are shown in Table 1. We could have omitted the solar panel and oversized our battery pack to extend our device's battery life; however, the solar panel outputs more energy than could have been added with an oversized battery.

Table 1 Solar Panel Output for Washington State in the Winter

	Hours of Peak Sun/day	Solar Panel Wattage	Days/month	kWh/month
Washington (Winter)	1.6	3.5	30	168

2.2.2 Charge Controller

Our design included a charge controller to safely charge the battery system. It prevents the batteries from being charged beyond 3.6V and discharged under 2.6V. The charge controller prioritizes using energy from the solar panel over energy from the battery pack. This prevents needlessly cycling the battery and extends its lifetime. In addition, the charge controller functions as a maximum power point tracker which maximizes the solar panel's output under a variety of solar conditions.

2.2.3 Battery

Our battery pack consists of eight rechargeable lithium ion cells. Each cell is 2500 mAh and 3.6V which means that our battery pack has a 20 Ah (72 Wh) capacity and a 3.6V output. By connecting additional

battery packs in parallel, our device's battery capacity can be easily expanded. We considered using lead acid or NiMH batteries instead of lithium ion batteries but they were heavier and more expensive.

2.2.4 Voltage Regulators

We used switching regulators to supply our components with power. The regulators took a 3.6V input and converted it to 3.3V, 5V, and 8V. We originally used linear regulators but changed our design to incorporate switching regulators because linear regulators were too inefficient. Switching regulators are slightly more expensive but that additional cost was justified by the fact that switching regulators are up to 90% efficient. Towards the end of our project, we had issues with the microcontroller in our PCB so we were forced to incorporate an Arduino Uno in our design. This change required us to include an 8V regulator for the Arduino. The power system in our PCB still functioned so we used its regulators to supply our device with 3.3V and 5V. Figure 3 shows our regulator's design.

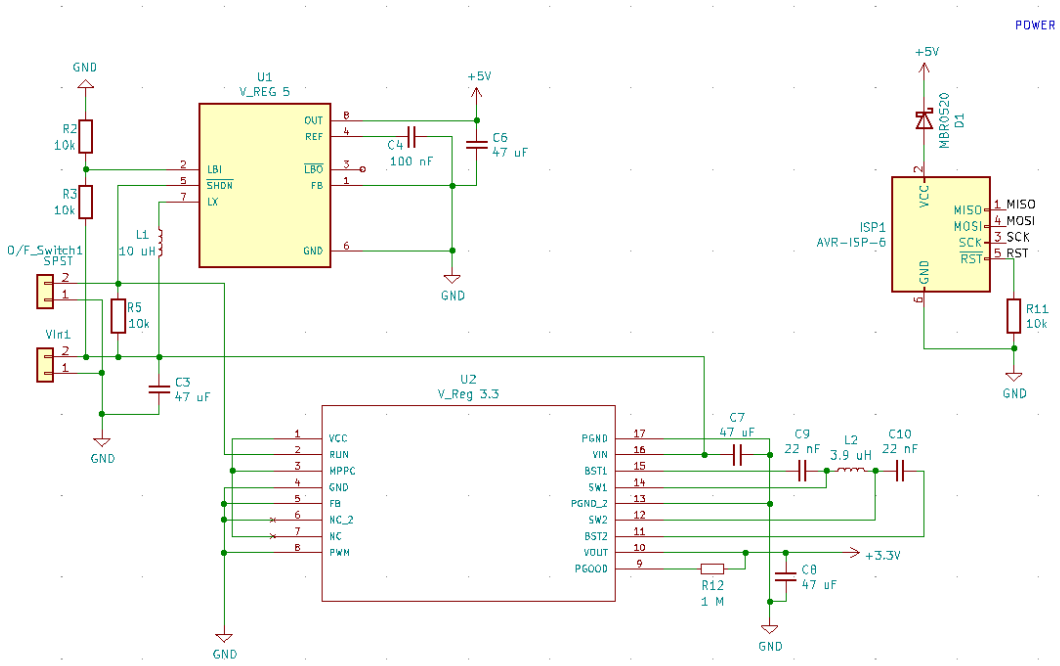

Figure 3. Schematic of the 3.3V and 5V regulators and ISP

2.3 Sensor Array

The Sensor Array consists of 5 sensors: a wind speed sensor, a wind direction sensor, a temperature/humidity sensor, a particulate matter sensor, and a RTC module. Each sensor will communicate with the microcontroller via GPIOs. The sensors will be powered by the power subsystem through the 3.3V and 5.5V voltage regulators. Figure 4 below shows the schematic of the sensor array.

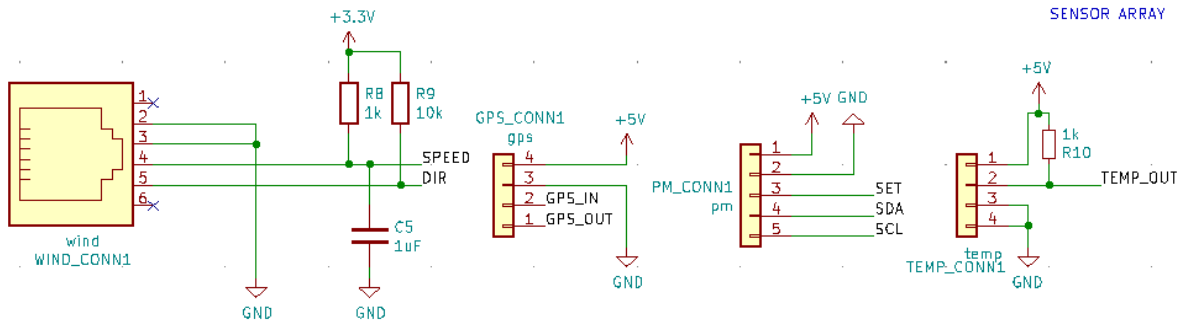

Figure 4. Schematic of the sensor array

2.3.1 Wind Sensor

The wind sensor returns both wind speed and direction via two analog signals. In addition, the wind sensor utilizes 3.3V and a RC filter using two 10k resistors and two 1uf capacitors.

2.3.2 Temperature Sensor

The temperature sensor returns both the current temperature and humidity via a digital signal. The temperature sensor takes 5V in and is able to withstand outdoor conditions.

2.3.3 PM Sensor

The PM sensor returns the current PM 1.0, PM 2.5, and PM 10 levels via a digital signal. Specifically the PM sensor utilizes I^2C communication and returns a 32 bit packet to our microcontroller.

2.3.4 RTC

The RTC is used to keep track of current time and append that information to our log files. The RTC utilizes I^2C communication and takes in 3.3V. In addition, the RTC has a 3V battery to power the internal crystal oscillator.

2.4 Data Processing

The Data Processing Unit manages the storage of all sensor data. A microcontroller controls an SD card and communicates to sensors via GPIOs. The data processing unit will write to the SD card at a five minute, one hour, or three hour frequency. It also contains a small user interface with an LED, a button, and a three position slide switch. Figure 5 below showcases the circuit for our data processing unit.

One design alternative that could have been added is the usage of LoRa or another network based hardware. In this alternative, data would be transmitted in real time via a network (LoRa) instead of being stored on-device. As mentioned above there are limitations to LoRa, mainly its communication range. These limitations would limit where a user can place our system and collect data. Thus, we decided to use the on-device memory design approach over a network based one to address these limitations.

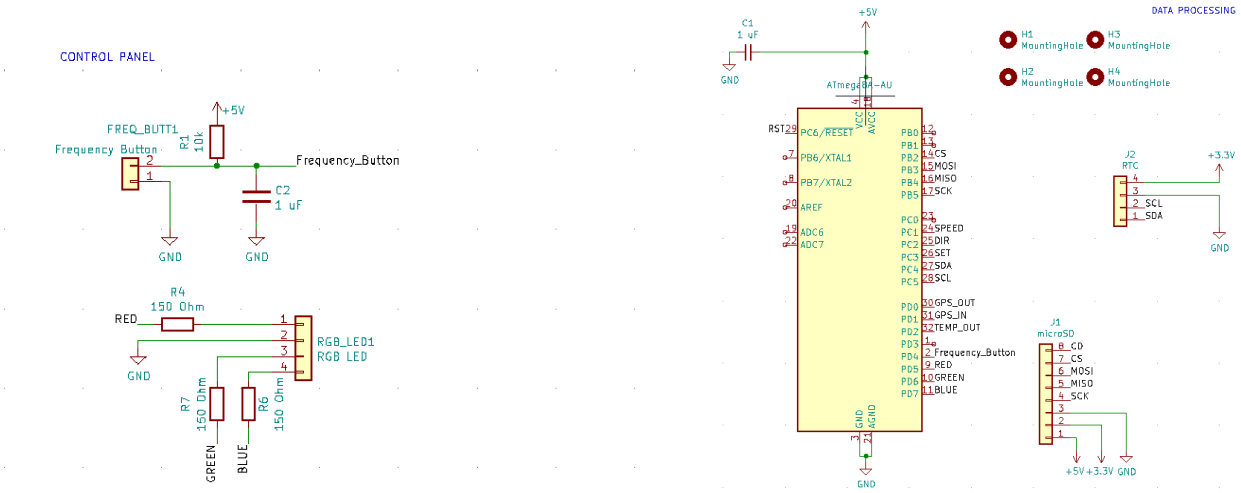

Figure 5. Schematic for data processing subsystem

2.4.1 Microcontroller

Our proposed design was to use the Atmega328-AU microcontroller for all of our data computation needs. We chose the Atmega328 due its compatibility with existing arduino libraries and sensors. Unfortunately, as we came to integrate the microcontroller with the sensor array, we realized that the internal 8 MHz oscillator crystal wasn't sufficient enough for our sensors. Upon research, the internal 8 MHz oscillator crystal has an inaccuracy of 10%[7]. This poor accuracy greatly effectives communication protocols such as UART, which most of our sensors use. Due to this limitation and in the interest of time, we decided to swap the microcontroller with an Arduino Uno.

2.4.2 On-Device Memory

One of the key features of our project is the ability to collect data autonomously in any location. To accomplish this, our system has a 8 GB SD card that directly communicates with our microcontroller. All sensor data collected is written to a .txt file residing on the SD card.

2.4.3 Control Unit

The last aspect of the data processing subsystem is the ability to view the status of sensors and change the write frequency. In the control panel we have two push down buttons (debounced using a 10k resistor and a 1 uF capacitor) and 1 RBG LED. One of the push down buttons is used to change the data write frequency (five minute, one hour, three hour). Upon changing the write frequency, the RBG LED will let the user know that the system has acknowledged the change in frequency (blue for five minutes, red for an hour, and green for three hours). Upon pressing the status button, the RBG LED will cycle through various signals per sensor. This status option allows users to check that all sensors are writing to memory correctly and that the battery is charged.

2.5 Web UI

The Web UI is a simple webpage that enables a user to visualize and interact with their data. Users will remove the SD card from the box and plug it into their computer inorder to extract the data. They can

then upload the data file to our webpage. The webpage will store the data in a MySQL database and compute visualizations via a Python backend. Figure 6 below shows the flow of our web UI application.

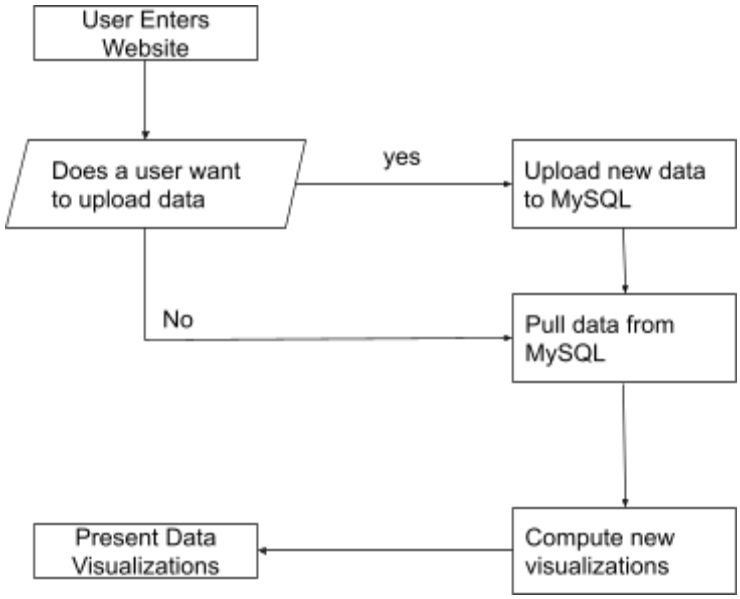

Figure 6. Web UI flowchart

2.5.1 Data Storage

One of the main components of the web UI is the ability to store the PM data for any data collection cycle. To accomplish this, we decided to use a MySQL database hosted on the Google Cloud Platform (GCP). MySQL was a sufficient option for this project, since our data is in a row/column order. In addition, we decided to host our database on GCP due to the scalability of cloud computing.

2.5.2 Data Visualization

The 2nd phase of the web UI is to compute various data visualizations using the data pulled from MySQL. The main visualizations we computed are line graphs that plotted a data type (temperature, wind speed, PM, etc.) against time. To compute our visualizations we used a wide range of web technologies such as Flask, Pandas, Numpy, and Charts.js. A few design alternatives include usage of a dashboard platforms (e.g. Tableau) or using a dashboard framework (e.g. Dash). We ended up building our dashboard from scratch since this option gave us more freedom and flexibility with computing our visualizations.

3. Design Verification

3.1 Power Supply

One of our high level requirements was that our sensor node must be able to record data for thirty days at all collection frequencies. To test this, we ran the sensor node at five minute data collection intervals for three hours and measured the battery voltage every sixty seconds. Because the voltage of a lithium ion battery decreases linearly as it discharges, it can be used to indicate the battery's charge. Figure 7 shows a graph of our battery's voltage throughout the test. Over the course of three hours, our battery's voltage dropped by .0583V which corresponds to 5.83% of the battery's total charge. Table 11 in Appendix C shows how this extrapolates to 2.18 days of battery life. Our device failed this test because we were forced to use an Arduino that doesn't support sleep mode. In active mode, an ATMega328 would have consumed half as much power as the Arduino, and in sleep mode an ATMega328 would have consumed one tenth as much power as the Arduino. In addition, we could not put our sensors in sleep mode which meant they consumed far more power than we originally planned. We also had to conduct this test indoors to record the battery voltage which meant that the solar panel couldn't provide supplemental energy. If we had used an ATMega328, we expect that our device would have passed this test. This is based on power consumption calculations that indicate our device would only consume 40.938 kWh/month which is just over half of our battery's capacity. These calculations are included in Table 12 in Appendix C.

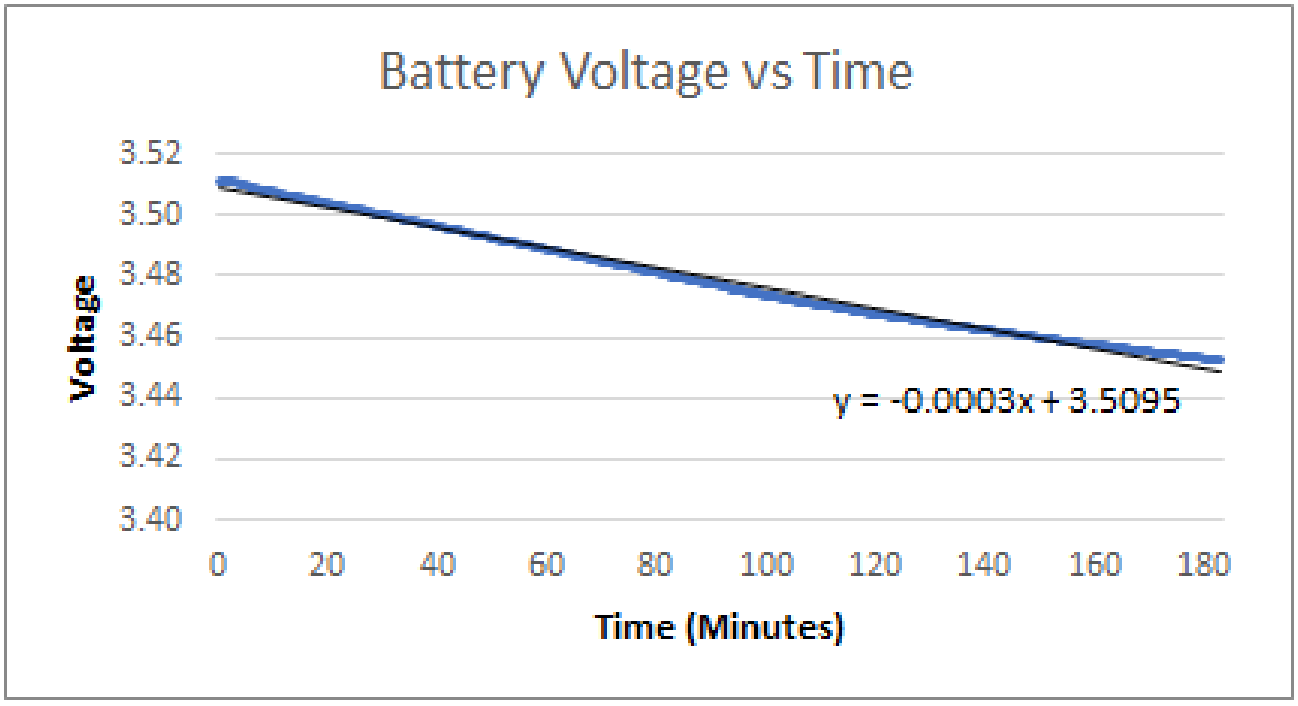

Figure 7. Battery voltage versus time during operation at five minute collection intervals

3.2 Sensor Array

Our sensor array is deemed successful if our sensor data is within 10% of truth value, with the truth value being the data collected by the National Weather Services (NWS) weather station in Champaign, IL. To verify our sensor array we let our system run for one hour at a five minute write frequency, and then compared our data with NWS data. The data collected from this test are shown in Tables 13 and 14 in Appendix D. As we can see in Table 2 below, all of our sensors have an average error rate of ~6.5%. While the average for the wind speed is around 14.8%, there are multiple factors that contributed to this high error rate. These factors include but are not limited to the precision of the anemometer and the height at which the anemometer is deployed. However, given that the rest of our sensors have an error rate of ~6.5%, we can conclude that our sensor array is successful.

Table 2 % Difference From Particulate Matter Sensor Node and NWS Weather Data

Date	Time (PM)	Humidity	Temperature	PM 2.5	PM 10	Wind Speed
11/30/21	6:26:59	18.3261183	13.9134481	11.5384615	6.66666667	17.6276771
11/30/21	6:32:01	15.819209	13.5514019	4.34782609	3.44827586	9.50226244
11/30/21	6:37:01	10.3633917	10.5415861	4.34782609	3.44827586	10.6002554
11/30/21	6:42:02	9.91957105	10.5415861	4.34782609	3.44827586	25.3918495
11/30/21	6:47:03	7.89473684	6.47118301	4.34782609	6.66666667	5.5408971
11/30/21	6:52:04	1.61090459	2.01271186	4.34782609	3.44827586	8.92531876
11/30/21	6:57:05	1.48514852	2.93809024	4.34782609	6.66666667	22.0489978
11/30/21	7:02:06	3.79746835	5.17683239	4.34782609	6.66666667	16.5275459
11/30/21	7:07:07	2.37203496	2.01271186	12	12.5	19.3317422
11/30/21	7:12:08	1.35970334	2.01271186	12	12.5	20.4819277
11/30/21	7:17:09	1.99004975	2.47759621	12	12.5	18.200409
11/30/21	7:22:10	1.99004975	3.39425588	4.34782609	3.44827586	4.16666667
AVG (%)		**6.41069885**	**6.25367629**	**6.86008919**	**6.78400383**	**14.8621291**

3.3 Data Processing

One of the most crucial requirements of our data processing unit is the ability for it to change its write frequency (five minutes, one hour, three hour). To verify this requirement we ran our system for approximately five hours while cycling through all three write cycles. As we can see from Figure 8, the log file our system is able to write to memory at every write frequency specified by the user.

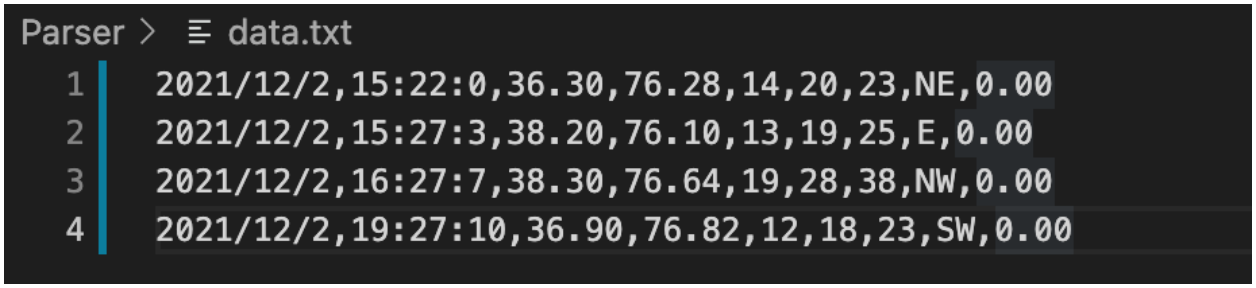

Figure 8. Log file

3.4 Web UI

In order to deem our Web UI subsystem successful we needed to verify that the data parsing algorithm was efficient. Specifically, the data parsing algorithm, which reads a .txt file and uploads the data to our database, should run under 10 seconds. To verify our algorithm, we ran a timed execution test using Python on three distinct data sets. For this test, we simulated one month worth of data at a five minute write frequency. As we can see in Figures 9, 10, and 11 below, each test completed in under 10 seconds which verifies our requirement.

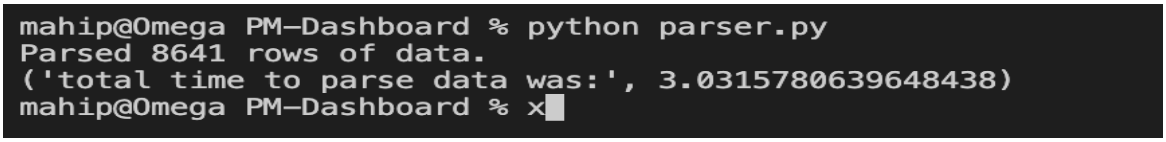

Figure 9. Test 1

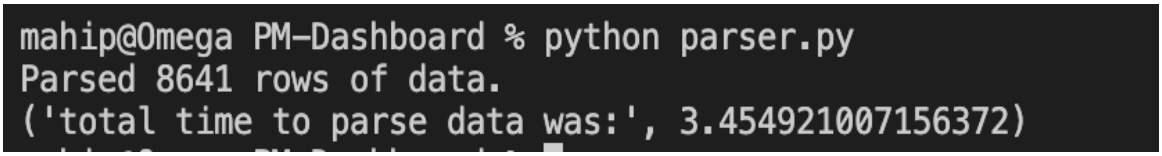

Figure 10. Test 2

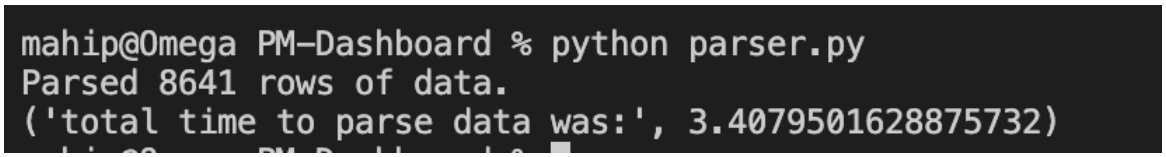

Figure 11. Test 3

4. Costs

4.1 Parts

We determined the cost of parts to be $272.52 and the cost of the physical structure to be $56.48 for a total of $329. In regard to the parts costs, the sensors cost the most. If we were to scale up the production of this device, producing some of the sensors ourselves or finding a different vendor could reduce the cost. In regard to the cost of the physical structure, casting the plastic as opposed to 3D printing it would reduce the cost significantly.

Table 3 Parts Costs

Part	Manufacturer	Retail Cost ($)	Quantity	Actual Cost ($)
ATmega328P (Microcontroller)	MEGA	$2.05	2	$4.10
Micro-SD card holder	RoHS	$7.50	2	$15
RTC	Adafruit	$5.95	1	$5.95
PM Sensor	SEED IOT	$32.90	1	$32.90
Wind Sensor	Shenzhen Fine Electronics	$79.25	1	$79.25
Temperature Sensor	Adafruit	$8.99	1	$8.99
GPS	Geekstory	$12.99	1	$12.99
Step-up Regulator	STMicroelectronics	$2.66	2	$5.32
200mAh Battery	Epoch Batteries	$3.99	6	$23.94
Solar Panel	Voltaic	$39.00	1	$39.99
Charge Controller	Adafruit	$9.95	1	$9.95
Battery Case	Ltvystore	$12.99	1	$12.99
Push Button Switch	NTE	$2.13	2	$4.26
10k ohm resistor	Stackpole Electronics Inc	$.1	7	$.7
1k ohm resistor	YAGEO	$.1	1	$.1
150 ohm resistor	Panasonic Electronic Components	$.32	3	$.96
1uf capacitor	Samsung Electro-Mechanics	$.1	1	$.1
100uf capacitor	Taiyo Yuden	$.1	3	$.3
100nf capacitor	Taiyo Yuden	$.1	1	$.1
47uf capacitor	Taiyo Yuden	$.61	4	$2.44
22nf capacitor	TDK Corporation	$.33	2	$.66
10 uH Inductor	Taiyo Yuden	$.25	1	$.25
4.2 uH inductor	Taiyo Yuden	$.33	1	$.33
Voltage Regulator	ANMBEST	$10.95	1	$10.95
Total				**$272.52**

Table 4 Physical Design Cost

Part	Manufacturer	Retail Cost ($)	Quantity	Actual Cost ($)
3D Printed Enclosure	UIUC Innovation Lab	$48.48	1	$48.48
Acrylic plastic	UIUC Innovation Lab	$8.00	1	$8.00
Total				**$56.48**

4.2 Labor

We determined the cost of labor for the project to be $73,800. We assume that a new graduate from UIUC's ECE department has an average salary of $88,000[8] per year or $41.07 per hour. In addition, we assume that each engineer will spend 15 hours a week for a total of 16 weeks, or 240 hours per engineer. Our formula to calculate the cost of each engineer will be:

($/hour) x 2.5 x hours to complete = total cost.

Table 5 Labor Cost

Name	Pay per hour	Total Time in hours	Multiplier	Total
Zack Plumley	$41.07	240	2.5	$24,600
David Young	$41.07	240	2.5	$24,600
Mahip Deora	$41.07	240	2.5	$24,600
Total				**$73,800**

4.3 Schedule

Over the course of the semester, we stuck to the schedule below. Looking back, we wish we had left more time to debug and verify hardware, as that's where we encountered the majority of our roadblocks. If we had done this sooner, we believe we would've had a fully functioning PCB by the day of the final demonstration.

Table 6 Schedule

Week	**Mahip**	**Zack**	**David**
10/11	Start designing Web UI	Set-up MYSQL server for Web-UI and design physical enclosure	Order Parts and design physical enclosure
10/18	Work on Web UI and start building physical enclosure	Start working on sensor related software	Start building physical enclosure
11/1	Test Web UI with Dummy Data and start assembly of data processing parts	Verify each sensors relationship with microcontroller	Start assembling power unit
11/8	Assembly of data processing parts and start integration of all subsystems	Start assembling sensor processing unit and start integration of all subsystems	Assembling power unit and start integration of all subsystems
11/15	Prepare for mock demo and continue assembly	Prepare for mock demo and continue assembly	Prepare for mock demo and continue assembly
11/22	Test and clean each subsystem integration	Test and clean each subsystem integration	Test and clean each subsystem integration
11/29	Final Demo	Final Demo	Final Demo
12/6	Final paper and presentation	Final paper and presentation	Final paper and presentation

5. Conclusion

5.1 Accomplishments

We've very happy that we were able to build a device that accomplished all of the high level requirements we set for it. The PM Sensor Node is able to accurately collect air data in most weather conditions independent of any power source or internet connection. Additionally, the website provides the user with a very intuitive way of interacting with the data collected from the device. Despite many hiccups along the way, we had a lot of fun designing, building, debugging, and testing the device! We hope that future projects continue to attack the issue of air pollution.

5.2 Ethical considerations

The device is meant to be fully unsupervised when our system is collecting data at a given location. We have identified a few ethical and safety issues that could occur from our system.

5.2.1 Overheating

Our rechargeable battery could potentially overheat and pose a fire risk. This a particular concern since our box is meant to be placed in a remote location, and could pose a wildfire risk. In accordance with IEEE Code of Ethics #3[9], "to avoid real or perceived conflicts of interest whenever possible". To prevent overheating and battery failures, we will charge each of our batteries in parallel using a charge controller to prevent overcharging.

5.2.2 Box Placement

Our system could be placed in private property without prior approval. This would be in breach of IEEE Code of Ethics #9[9] which states "to avoid injuring others property". To counter this, we will provide labeling on the box as well as documentation which states where a user can and cannot place their box.

5.2.3 Data Bias

Data collected by our device may introduce bias about sources of PM. For example, if our system is solely placed around factories that produce large quantities of PM, then it may be construed that all factories are large sources of PM. This breaches IEEE Code of Ethics #2[9] "To improve the understanding by individuals and society". If the device is used in a manner that produces biased data, it has the potential to incorrectly identify sources of PM.

5.3 Future work

There are several things that can be done to improve the initial version of the PM Sensor Node. First and foremost, the PCB can be improved by adding a 16Mhz crystal oscillator and a low power mode. The crystal oscillator will improve the accuracy of the clock and the low power mode will improve the efficiency of the device. Additionally, the physical design can be improved by adding weather proofing elements and changing sensor placement. Although the device was able to withstand common weather types, it does a poor job of stopping more niche phenomena such as sideways rain. Also, the wind direction and wind speed sensors are placed in a way that can, at certain times of the day, block sunlight from the solar panel; these sensors can easily be moved to the opposite side of the device to mitigate this issue. Lastly, the user interface can be improved by adding an LCD screen and improving the data

transfer process. Ideally, the user shouldn't have to open the device in order to check its status or to retrieve data. An LCD screen, buttons, and perhaps a wireless data transfer method can be added to the outside of the box in order to improve the user experience.

References

[1] "Particulate Matter (PM) Basics," *EPA*. [Online]. Available: https://www.epa.gov/pm-pollution/particulate-matter-pm-basics#PM. [Accessed: 22-Oct-2021].

[2] "Health and Environmental Effects of Particulate Matter (PM)," *EPA*. [Online]. Available: https://www.epa.gov/pm-pollution/health-and-environmental-effects-particulate-matter-pm. [Accessed: 07-Dec-2021].

[3] Danel Vincent, "Airborne particulate matter and their health effects," *Encyclopedia of the Environment*, 16-Aug-2019. [Online]. Available: https://www.encyclopedie-environnement.org/en/health/airborne-particulate-health-effects/. [Accessed: 05-Dec-2021].

[4] "PM Data In the US," *EPA*. [Online]. Available: https://maps.response.epa.gov/portal/apps/webappviewer/index.html?id=81f0cb69daf141f89b50e768d11a672b. [Accessed: 22-Oct-2021].

[5] N.O.A.A. US Department of Commerce, "Citizen Weather Observer Program," *National Weather Service*, 08-Jul-2019. [Online]. Available: https://www.weather.gov/cle/CWOP. [Accessed: 22-Oct-2021].

[6] "Washington Sunlight Hours & Renewable Energy Information," *TurbineGenerator*, 27-Mar-2018. [Online]. Available: https://www.turbinegenerator.org/solar/washington/. [Accessed: 07-Dec-2021].

[7] "Microchip technology," *Microchip.com*. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf. [Accessed: 05-Dec-2021].

[8] "Salary Averages," *The Grainger College of Engineering.* [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages. [Accessed: 22-Oct-2021]

[9] "IEEE code of Ethics," *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 22-Oct-2021].

Appendix A Requirement and Verification Tables

Power Subsystem

Table 7 Power Subsystem Requirements and Verification

Requirements	Verification	Verification Met
1. Batteries must be supplied 3.7V +/- 10% during charging 2. The battery must be able to supply power to the sensor node for a month when the node is making measurements at all data collection frequencies (5 minutes, 1 hour, 3 hours). 3. The power supply must be able to supply both 5V and 3.3V throughout our system.	1. Measure the open-circuit voltage with a voltmeter, ensuring that it is between 3.33 V and 4.07V 2. Measure power consumption over one day for each frequency, extrapolate that to a month, and ensure that the battery and solar panel can supply that much energy. 3. Using a Multimeter we probe the voltage regulator to ensure that it can output 3.3V and 5V based on the sensor and other factors	1. Yes 2. Yes 3. Yes

Sensor Array Subsystem

Table 8 Sensor Array Subsystem Requirements and Verification

Requirements	Verification	Verification Met
1. Sensors must be able to collect data without a significant (greater than 10%) loss in accuracy in various types of weather, specifically sunny, windy, rainy, and stormy weather	1. a. Take the device outside and compare collected data with that of local weather stations b. Repeat step a in sunny, windy, rainy, and stormy weather	1. Yes

Data Processing Subsystem

Table 9 Data Processing Subsystem Requirements and Verification

Requirements	Verification	Verification Met
1. Needs to be able to store 30 days worth of air data (PM, speed, direction, temperature, humidity, and device location) 2. Writes to device memory on 5 minute, 1 hour, or 3 hours intervals depending on the frequency setting 3. Control panel displays the correct frequency setting and error signals within 0.5 seconds of interaction	1. Calculate how many bytes are needed to store air data to 4 significant figures in a .txt file by the maximum number of collections made in 30 days (8640 using 5 minute intervals). If the number of bytes is less than 2GB, our device is compatible with all MicroSD cards. 2. For each frequency setting, collect data for 24 hours while also logging the time at which data is being logged. Verify that the intervals specified in the data match that of the device. 3. Time the response between changing the frequency or an error being thrown and the status shown by the control panel. a. Rotate through the frequency settings to verify that each color displayed matches that of its corresponding frequency b. Throw all possible errors (sensor disconnect, low power, data not being written to sd card) to verify that each color displayed matches that of its corresponding error	1. Yes 2. Yes 3. Yes

Web UI Subsystem

Table 10 Web UI Subsystem Requirements and Verification

Requirements	Verification	Verification Met
1. The website must be able to parse and display the data within 10 seconds of the file being uploaded.	1. Using the date/time library in python, verify that our queries and data upload is within 10 seconds	1. Yes

Appendix B Overall Circuit Schematic

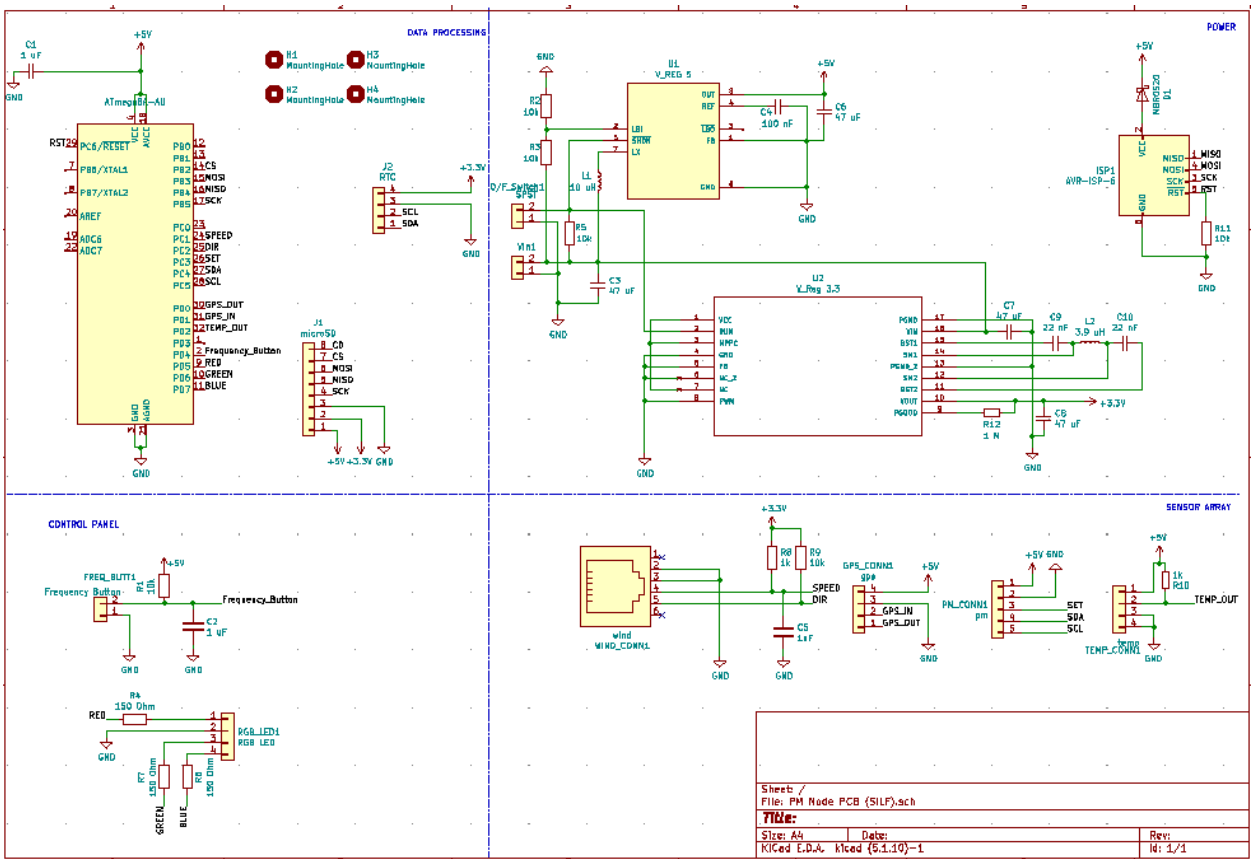

Figure 12. Overall Circuit Schematic

Appendix C Power Supply Validation Data

Table 11 Battery Life

Battery Percentage	Voltage	Time (Days)
100%	3.6	0
90%	3.5	0.22
80%	3.4	0.44
70%	3.3	0.65
60%	3.2	0.87
50%	3.1	1.09
40%	3	1.31
30%	2.9	1.52
20%	2.8	1.74
10%	2.7	1.96
0%	2.6	2.18

Table 12 Power Consumption Calculations for Five Minute Data Collection Intervals

Sensor	Voltage (V)	Current (mA)	Power (mW)	Active Time During Collection (sec)	minutes/ cycle	cycles/ month	hours/ month	Wh/ month
ATmega328P (Active)	5	5.2	26	35	0.583	8640	84.000	2.184
ATmega328P (Idle)	5	1	5	-	4.417	8640	636.000	3.180
Grove PM (Active)	5	75	375	31	0.517	8640	74.400	27.900
Grove PM (Standby)	5	0.15	0.75	-	4.483	8640	645.600	0.484
SparkFun microSD Transflash Breakout (R/W)	5	100	500	37.5 E-6	625 E-9	8640	90 E-6	45 E-6
SparkFun microSD Transflash Breakout (Sleep)	5	0.5	2.5	-	5.000	8640	720.000	1.800
Sparkfun Wind Speed	3.3	3.3	10.89	-	2.500	8640	360.000	3.920
Sparkfun Wind Direction	3.3	0.3086	1.0189	-	5.000	8640	720.000	0.734
DHT22 Sensor (active)	5	1.5	7.5	31	0.517	8640	74.400	0.558
DHT22 Sensor (stand by)	5	0.05	0.25	-	4.483	8640	645.600	0.161
COM-11120 RGB Diode	9.9	20	198	-	5.000	1.000	0.083	0.017
							Total	40.938

Appendix D Sensor Array Validation Data

Table 13 Particulate Matter Sensor Node Collected Data

Date	Time	Humidity (%)	Temperature (Fahrenheit)	PM 1.0 (ug/m^3)	PM 2.5 (ug/m^3)	PM 10 (ug/m^3)	Wind Direction	Wind Speed (MPH)
11/30/21	6:26:59	69.3	42.98	18	26	30	E	6.07
11/30/21	6:32:01	70.8	42.8	16	23	29	SE	11.05
11/30/21	6:37:01	74.3	41.36	16	23	29	E	7.83
11/30/21	6:42:02	74.6	41.36	16	23	29	E	3.19
11/30/21	6:47:03	76	39.56	16	23	30	E	3.79
11/30/21	6:52:04	80.7	37.76	16	23	29	E	5.49
11/30/21	6:57:05	80.8	38.12	16	23	30	E	8.98
11/30/21	7:02:06	79	39.02	16	23	30	E	5.99
11/30/21	7:07:07	80.1	37.76	17	25	32	E	4.19
11/30/21	7:12:08	80.9	37.76	17	25	32	E	4.15
11/30/21	7:17:09	80.4	37.94	17	25	32	E	9.78
11/30/21	7:22:10	80.4	38.3	16	23	29	E	4.8

Table 14 NWS Reported Weather Data

Date	Time	Humidity (%)	Temperature (Fahrenheit)	PM 2.5 (ug/m^3)	PM 10 (ug/m^3)	Wind Direction	Wind Speed (MPH)
11/30/21	6:26:59	82	37	23	28	E	5
11/30/21	6:32:01	82	37	22	28	E	10
11/30/21	6:37:01	82	37	22	28	E	7
11/30/21	6:42:02	82	37	22	28	E	4
11/30/21	6:47:03	82	37	22	28	E	4
11/30/21	6:52:04	82	37	22	28	E	5
11/30/21	6:57:05	82	37	22	28	E	7
11/30/21	7:02:06	82	37	22	28	E	5
11/30/21	7:07:07	82	37	22	28	E	5
11/30/21	7:12:08	82	37	22	28	E	5
11/30/21	7:17:09	82	37	22	28	E	8
11/30/21	7:22:10	82	37	22	28	E	5