IoT Environmental Justice: Air Quality Monitoring of Seneca Falls Landfill

MAE 4220/MAE 5220 Final Design Report

Evelyn Chiu (MechE B.S. ‘24)
Endon Demiri (MechE B.S. ‘24)
Hongpufan (Evan) Huang (MechE M.S. ‘25)
Isabel Mejía-Roberts (MechE B.S. ‘24)

May 16, 2024

1. EXECUTIVE SUMMARY

Our project addresses the critical issue of environmental injustice by focusing on air quality monitoring around the Seneca Meadows Landfill. On March 24, 2024, a complaint filed to the Supreme Court of the State of New York alleges that the Seneca Meadows Landfill—New York's largest landfill located in Waterloo, NY—has "substantially interfer[ed]" with locals' rights to clean air and a healthy living environment (enshrined in New York's Green Amendment) due to its "noxious, offensive odors." This landfill, which is the largest in New York State, releases harmful gasses like methane and carbon dioxide that contribute to poor air quality and health risks. It therefore becomes critical to provide data on air pollutant levels and air quality for communities local to the landfill who are disproportionately affected by pollution to advocate for a healthier living environment.

This project is a collaboration between Cornell University (Evelyn Chiu, Endon Demiri, Hongpufan Huang, and Isabel Mejía-Roberts) and Blueprint Geneva (Jackie Augustine) to provide the technology and methods needed to track air pollutant levels and monitor air quality in areas neighboring the Seneca Meadows Landfill. This partnership significantly influenced our design choices and project scope, ensuring that the final product meets the community's needs and is feasible for future use. In Section 2: REVIEW OF RELATED WORK, we explain how our research into related technologies and endeavors in air quality monitoring informed our choices for the system implementation, striking a balance of real-time data collection and analysis capabilities, cost-effectiveness, accessibility, and simplicity. With guidance from Blueprint Geneva, we designed and developed a sensor module housed in a Stevenson screen to monitor air quality parameters, including methane, hydrogen sulfide, particulate matter, volatile organic compounds, and environmental conditions like temperature, pressure, and humidity. The system utilizes the MCCI Catena 4410 Featherboard for data collection and transmission via The Things Network (TTN), ensuring real-time data availability and air quality monitoring for community analysis and use. In-depth details of our project scope, system design, testing, implementation, and analysis can be found in Section 4: PROJECT SCOPE and Section 5: TECHNOLOGICAL DEVELOPMENT. Lastly, our sensor addressed an objective for the sensors to be low-maintenance and easily replicable by non-engineers, per recommendation by Blueprint Geneva. This makes our sensor accessible to other communities facing similar challenges and allows for the possibility of mounting many sensors together to create a network that gives a cohesive look on pollutants around a single community. At the end of this report, we also include a rudimentary framework for future work on this project, some barriers that we encountered, and our final recommendations to future colleagues in Section 6: PLAN FOR NEXT STEPS.

As a result, our project lays the groundwork for a cost-effective and accessible method for air quality monitoring, which ultimately should help the Waterloo community to support their allegations of environmental rights' violations by the Seneca Meadows Landfill with relevant, quantitative data. Our collaboration with Blueprint Geneva ensures that the data collected will be used to advocate for systemic changes to improve air quality and public health in the region. With a focus on reproducibility, we hope that our community partner, as well as others, will be able to replicate this low cost air quality monitor in order to ensure a better quality of life for their surrounding communities.

2. SOCIAL CONTEXT

Environmental justice is an important, pressing issue that affects many sectors of the population. Generally, studies have shown that minorities and low-income communities are more negatively affected by climate change and environmental issues. These disparities are evident in numerous aspects of the environment, such as access to clean air. Clean air is critical for good health, yet studies show that communities of color and low-income areas are disproportionately exposed to air pollution and poor air quality. According to a study conducted in the United States by the Commission for Racial Justice, African Americans are almost 80% more likely than their white counterparts to live in a neighborhood where industrial air pollution is considered to cause the most significant health risks.

Poor air quality and exposure to pollution can lead to short-term and longer-term health complications. These negative effects are seen especially in vulnerable populations, including young children, older adults, and pregnant women. Some of these negative impacts include stroke, ischaemic heart disease, chronic obstructive pulmonary disease, lung cancer, and pneumonia, among others. Additionally, children exposed to high levels of air pollution may suffer from developmental delays and reduced lung function, which can affect their overall quality of life and long-term health prospects.

Pollution and poor air quality not only affect human health but also have broader environmental consequences, contributing significantly to climate change. Methane (CH_4), which is released from landfills, coal mining, certain industrial and agricultural practices, etc., is extremely harmful to the atmosphere as it is very efficient at trapping heat, thereby accelerating climate change. Another gas that is harmful to the environment and people's health is carbon dioxide (CO_2). Carbon dioxide can be released from the burning of fossil fuels, landfills, and also natural sources such as volcanoes. Carbon dioxide is another major contributor to climate change as it is one of the most common greenhouse gasses. Both of these gasses can also have adverse effects on health since they reduce the amount of oxygen breathed from the air.

These issues can also be observed in areas of upstate New York, such as Geneva, Waterloo, and Seneca Falls due to the proximity of Seneca Landfill. Seneca Meadows Landfill is the largest landfill in the state of New York, with more than 350 acres of landfill and more than 2 million tons of trash arriving yearly via trucks on the State Thruway. Unsurprisingly, the neighboring communities have launched several complaints and campaigns against the landfill, especially in the wake of their most recent expansion proposal. This proposal calls for the addition of 47 acres to connect the landfill's two massive mounds. This expansion would be enough to fill MetLife Stadium 10 times. This problem, however, is multifaceted as the landfill and its expansion would create several jobs for the surrounding communities. The landfill is a critical player in the local economy.

The landfill releases several gases that are harmful not only to people but also to the atmosphere, such as methane, carbon dioxide, and carbon monoxide. These surrounding communities are low-income, meaning that they have unequal access to adequate healthcare to address the negative health impacts that the air quality has. The environmental burden placed on these communities is further compounded by socio-economic factors, making it difficult for residents to advocate effectively for their health and environmental rights.

The landfill has tried to mitigate these issues by releasing aerators that emit flowery scented mist along the landfill's perimeter. While unique, this solution does not address the deeper root of the problem: the gasses and air pollution being released into the air in the nearby communities. The aerators simply mask the problem but do not address the larger issues: environmental and health concerns. When we visited Geneva and talked to members of the community at Waterloo Container, one team member described the scent of the aerators to be "bubble gum mixed with trash" highlighting the fact that these so called "solutions" are merely there to appease the community. It is abundantly clear that these aerators do not actually solve the actual problem at hand.

A local environmental justice non-profit, Blueprint Geneva, has developed and organized efforts to combat the injustices surrounding the negative effects of the Seneca landfill on the area. In previous years, they partnered with a local software engineer to create a website where users could log complaints of bad air smell (associated with bad air quality) to send those complaints to local politicians and push for them to address the issue at hand. On a larger scale, Blueprint Geneva focuses on bringing justice and equity to traditionally under-represented and under-resourced groups. Some of their other projects involve economic justice and opportunity, food justice, and urban agriculture.

The goal of our project is to create a sensor module that can help Blueprint Geneva provide accurate data and information about gasses and compounds in the areas surrounding the landfill so that they can later use this information to probe local lawmakers to enact systemic change. The development and implementation of such technology can empower the community with scientific evidence to back their claims, thereby strengthening their advocacy for a healthier and more just environment. Moreover, this initiative aligns with broader goals of promoting sustainable practices and ensuring that all communities, regardless of socio-economic status, can enjoy the right to clean air and a healthy living environment.

The issues observed in Seneca Falls are not unique to this region but are part of a broader pattern of environmental injustice seen across the United States and globally. Many communities around the world face similar struggles, as industrial sites, landfills, and other sources of pollution are disproportionately located in areas inhabited by marginalized groups. This pattern underscores the need for comprehensive policies and regulations that not only address pollution control but also ensure equitable distribution of environmental benefits and burdens.

Environmental justice also intersects with other forms of social justice, such as economic and racial justice. For example, economic inequalities can exacerbate environmental injustices, as poorer communities may lack the resources to fight against harmful environmental practices or to relocate to less polluted areas. Racial discrimination can further compound these issues, as historically marginalized racial groups may face additional barriers in advocating for their rights and accessing clean environments. Addressing environmental justice, therefore, requires a holistic approach that considers these intersecting issues and works towards systemic change.

Furthermore, public awareness and community engagement are crucial components of environmental justice efforts. Educating the public about the health risks associated with poor air quality and pollution, as well as about their rights to a clean environment, can empower individuals to take action and demand change. Community engagement initiatives, such as local meetings, workshops, and collaborations with non-profits like Blueprint Geneva, can help build a strong, informed, and active community that can effectively advocate for environmental justice.

Ultimately, achieving environmental justice requires the collective effort of communities, non-profits, policymakers, and other stakeholders. By working together, we can develop and implement strategies that address the root causes of environmental injustices and create healthier, more equitable environments for all. This includes not only mitigating the immediate impacts of pollution but also working towards long-term solutions that prevent environmental injustices from occurring in the first place. Through continued advocacy, education, and innovation, we can make significant strides towards a more just and sustainable world.

3. REVIEW OF RELATED WORK

Currently, in the Geneva community near the Seneca Meadows Landfill, there is a website called itstinks.org, which was created for publicity and tracking purposes. This website also tracks odors from the Ontario County Landfill and the Dunn Landfill. Users can report smells and their locations, which are then marked on a map and emailed to the DEC. This website primarily serves visualization purposes, allowing for the tracking of odor complaints and highlighting the areas affected by landfill emissions. A problem of this design is that oftentimes, the DEC ignores the complaints of the individuals who log odors. When they do address the complaints, they test the air at a time much later than when the complaint was put in and as a result, are unable to confirm the odors. While this tool is useful for community awareness, it focuses on a qualitative approach and lacks the real-time, comprehensive data collection and analysis capabilities that our project aims to provide.

Several air quality monitoring systems have been developed in the past with varying degrees of success. One such example is the study by Jabbar et al. (2022), which developed a LoRaWAN-based IoT system for outdoor air quality monitoring. This system demonstrated the feasibility of using LoRaWAN for long-range communication in environmental monitoring, similar to our project's approach. The use of LoRaWAN technology allows for long-range, low-power connectivity, enabling the deployment of a network of sensors over a wide area. This is a significant improvement over traditional short-range communication technologies, which often limit the scalability and effectiveness of air quality monitoring systems. This study deployed sensors that measured NO2, SO2, CO2, CO, PM, temperature, and humidity. The data was presented on a web-based dashboard that allowed users to determine the air quality.

In addition to academic research, several commercial solutions provide insights into effective air quality monitoring. Disrupt-X's Ignite Shield, for example, offers outdoor air quality monitoring services with cloud-based data analytics and visualization tools. These systems are able to provide insight into different conditions like gas monitoring in the home, water leakage detection, agricultural monitoring, and gas metering as well. However, due to its complexity, it comes at a high cost and may not be accessible to all communities.

Another innovative commercial system is the S-pod, developed by the Environmental Protection Agency (EPA). The S-pod is a solar-powered sensor system that combines air pollutant and wind field data with inverse algorithms to detect and locate source emissions of volatile organic compounds (VOCs). It was employed in Rubbertown, an industrial district in Louisville, Kentucky, to quantify VOC emissions near chemical facilities. The S-pod collects over 200,000 five-minute average data points about VOC concentrations, providing a comprehensive understanding of emission patterns. Its cost-effectiveness and simplicity make it suitable for facility fenceline monitoring.

Thus, in the development of our sensor, we took inspiration from these three devices. Our project incorporates some of the work done by Jabbar et al. by selecting sensors known for their reliability and accuracy, such as the BME688 and PMSA003I sensors. Similar to that project as well, we utilized LoRaWAN technology in our sensor. This would allow us to collect real-time data across long-distances, which would prove to be useful as we have more sensors deployed. However, we collect a slightly different set of sensor data than this study. The chemical compounds that our sensor detects are more oriented towards odor, pollutants and air quality from the landfill. Jabbar et al.'s study focuses on detecting air quality and does not have the CH4 and H2S detection abilities that our sensor does. Furthermore, to set our project apart from Jabbar et al.'s study, future prototypes of our device will also include wind direction and speed detection, allowing for us to understand how odors dissipate and travel with the weather.

A common problem that we observed across these three systems was that all of the systems included a dashboard that needed to be paid for monthly, which becomes costly over long periods of time. Our approach involves using a Python script to collect the data from TTN, which creates a text file that can later be plotted. This allows for data to be accessed without a monthly subscription plan, making it accessible to many users for a longer period of time and without any additional costs. Unlike the Disrupt-X ignite system, our system does not prioritize creating a profit from the device, and instead focuses on a community-centric approach that will empower local stakeholders to advocate for meaningful policy changes.

Lastly, our project prioritizes replicability, which differs from the previous work. Because the Disrupt-X's ignite shield was created for the commercial market and the S-pod was created for Rubbertown, their work is most likely not replicable by the common person and would need to be bought. Our project prioritizes replicability with a low-cost, which makes our project accessible to the marginalized communities by the landfill. Our project can also be easily replicated and deployed in various locations within one community for a lower cost, which would be beneficial in getting widespread knowledge within one community about how pollutants travel. This also means that our system's design is scalable and offers a blueprint and base design for other communities to adapt according to their needs.

To build off of what Geneva has currently with its website, itstinks.org, we hope that enough of our sensors will be deployed one day in conjunction with the website. Because this is the first semester that this project has been started, we hope to lay the groundwork for a system like this. When a complaint is logged into "itstinks.org", having the quantitative data of chemical compounds in the air around the same location that the complaint was logged will enhance the system and give more proof to each complaint that was logged. Thus, a system like this will give the community the voice it needs when it comes to environmental justice.

4. PROJECT SCOPE

4.1. PROJECT VISION

The Geneva community has expressed significant concerns regarding the reduced quality of life due to persistent odors and poor air quality originating from the Seneca Meadows Landfill. Traditionally, when community members report these issues, officials respond long after the initial complaints, by which time the odors have typically dissipated. This delay undermines the community's ability to substantiate their grievances effectively

To support the community's claims and affirm their experiences with the landfill, our project envisions developing a low-cost, easily replicable sensor network. These sensors will be strategically mounted around Geneva and the Seneca Meadows Landfill to continuously report on pollution sources and emission levels, comparing data across different locations. The sensors will measure compounds such as methane, hydrogen sulfide, reduced sulfur compounds, VOCs, carbon monoxide, and carbon dioxide, all of which contribute to odor production. Additionally, they will collect environmental data such as wind speed, direction, and humidity to analyze how odors travel and dissipate across the community.

The long-term goal of this project is to deploy multiple sensors connected over a LoRaWAN-based IoT network, enabling real-time monitoring of outdoor air quality over a large area. By mounting these sensors in various locations, the community will gain valuable insights into pollutant dispersion based on weather conditions, empowering them to advocate for a healthier environment.

4.2. DESIGN OBJECTIVE

Given the limited time frame of half a semester, our primary objective for this semester was to develop an initial prototype of the sensor and ensure its operational functionality. This includes both hardware and software development to create a tangible, reliable data collection system. Our first priority was looking at sensor selection and integration.

For our project, we wanted to select sensors that were reliable, relatively low-cost, and had a lot of existing documentation to reference and utilize in our code. We wanted to make sure that we selected a group of sensors that were able to measure the concentration of various air pollutants and environmental data. We also wanted to utilize LoRaWAN and TTN connectivity to ensure reliable data transmission and real-time monitoring capabilities. Mechanically, we created a casing for the device with two main roles: one, it will protect the sensors inside it but allow it to still operate while within this enclosure, and two, provide a way that we can mount it easily and upon different types of surfaces. Upon completion of our initial prototype, we tested the sensors around Ithaca to ensure that it was able to operate and provide environmental data. With the assumption that Ithaca has good air quality, we used our data to calibrate the sensors to use as a baseline for understanding the results that our sensor produces when in Geneva. Upon successful prototype testing, we deployed the first sensor at Waterloo Container, a site near the landfill that was pre-approved by our community partner, Jackie Augustine.

To set up this project in a place that can continue beyond the work we will accomplish this semester, we will compile all our work into a final design package for Blueprint Geneva to replicate and continue the project. This will be useful in generating data at other potential locations beyond the one at Waterloo Container. One includes mounting a sensor by Route 414, the road that the garbage trucks take to head to Seneca Meadows. Another option would be for a location near the factory in Geneva that produces a lot of air pollutants. A last option includes finding a "control" location with a reliable air quality, so that our results from near the landfill could be compared in real-time. With this in mind, our main design objective of the semester was to lay the groundwork for future development to improve the air quality monitoring systems for the Geneva community.

4.3. IOT SYSTEM UNDER CONSIDERATION

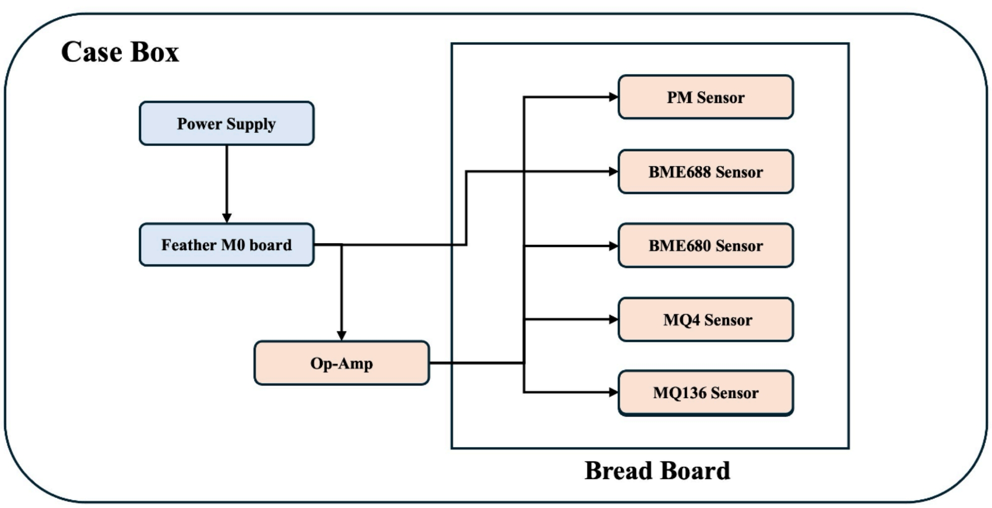


Figure 4-1: Design of Case Box

The power supply will come from a 120V outlet provided by Waterloo Container. There were originally plans for implementing a solar panel on the case design to make the module more self-sufficient, but we had to remove the solar panel based on technical complications that would hinder our deployment plans and our timeline for the semester-long project if we had decided to move forward with that design. The power supply is connected to our Featherboard to power the sensors, with some of the voltage provided by the Featherboard being amplified by an operational amplifier to power the sensors that require a larger voltage to operate. The op-amp is powered using a 9V battery. The sensors are connected on a breadboard to the featherboard within our case box — we are collecting data on chemical compounds methane, hydrogen sulfide, PM (particulate matter), VOCs (volatile organic compounds), VSCs (volatile sulfur compounds), and CO_2 (Carbon dioxide). These chemical compound data come directly from the BME688 sensor, PM sensor, MQ4 sensor, and the MQ136 sensor. We are also collecting data on some environmental conditions, including temperature, pressure, and humidity from the BME680 and BME688 sensors.

4.4. COMMUNITY PARTNER INFLUENCE ON PROJECT SCOPE

In our initial meeting with our community partner, she was most concerned about hydrogen sulfide and methane in the air from the landfill. She mentioned that there was a lack of data of the chemical composition of the air around landfills, and wanted us to deploy low maintenance sensors around the fenceline of the Seneca Meadows landfill and strategically place them in areas where pollutants would be. She also mentioned not having a lot of knowledge in sensors, and prioritized having a sensor that would be easily replicable for a non-engineer.

Coming into the meeting, our team's current idea was to select suitable sensors, define a chemical threshold of "stinkiness" and set up the hardware and software to notify the community when the threshold was reached. However, after our meeting, we learned that our community partner prioritized getting data and having it be analyzed — in her words, she did not want to "point fingers at the source of the pollution, and instead investigate it from the point of view of the community." Thus, in conjunction with our community partner, we adjusted this initial project idea and instead wanted to get a data stream that was accessible.

We also needed to adjust the timeline and the project scope a bit — our community partner was ambitious about our project, hoping that we would be able to deploy 3-5 sensors by the end of the semester, but considering that this was a new project, we discussed that we needed more time to develop a prototype. We all decided that it would be most beneficial for us to focus our time this semester in developing a well-tested prototype and creating a user guide for her to replicate in the future.

5. TECHNOLOGICAL DEVELOPMENT

5.1. SENSOR SELECTION

We currently have five sensors selected and integrated into our deployed prototype. We were initially planning to have more sensors but realized that it was not feasible within our one-semester timeline to mount more, especially within the power constraints of the board and size constraints of our sensor. The extensive list of compounds provided by our community partner was used to select the majority of the sensors. We also focused on selecting sensors that were low cost so that our community partner, as well as many others, can easily replicate our device. We also prioritized choosing sensors with a high level of documentation, which helped us understand their reliability and understand how to utilize them in our design. The ranking of compounds to detect that drove some of our selection process is below. Underlined are compounds that our purchased sensors can detect:

1. PM sensors - various sensitivities	6. NOx	11. Ammonia
2. Methane	7. Carbon Monoxide	12. Benzene

3. Hydrogen Sulfide
4. Reduced sulfur compounds
5. Total VOCs (Volatile organic compounds)
8. Carbon Dioxide
9. Ozone
10. Sulfur Dioxide
13. Dioxins
14. Various Aldehydes
15. Meteorological Data

5.1.1. ADAFRUIT BME 680 SENSOR (BLACK):

Model: BME680 *Manufacturer:* Bosch

The Adafruit BME 680 sensor can provide temperature(±1.0°C), humidity(±3%), barometric pressure (±1 hPa) data. It also includes a small MOX sensor which will detect the gas information by monitoring the resistance changing.

This sensor provides high-accuracy data that defines the base information of the environment and affects the target gas diffusion rate. This data gives us a baseline to analyze the pollution diffusion rate.

5.1.2. SPARKFUN ENVIRONMENTAL BME688 SENSOR (RED):

Model: BME688 *Manufacturer:* Bosch

The Adafruit BME 688 sensor is the further version of BME680. Both of the two sensors have the same function on temperature, humidity, and barometric pressure data. Compared with BME680, BME688 can detect the VOCs (volatile organic compounds), VSCs (volatile sulfur compounds), and CO_2 (Carbon dioxide) data. Because of the same functions, BME688 also can be the calibration method of BME680. Volatile organic compounds are emitted from certain solids or liquids, and can include an array of chemicals that can cause health complications.

5.1.3. ADAFRUIT PMSA003I AIR QUALITY BREAKOUT SENSOR (BLUE – NOT CONNECTED TO BREADBOARD):

Model: PMSA003I *Manufacturer:* Plantower

The Adafruit PMSA003I sensor uses laser scattering to radiate suspending particles in the air. It does so with a laser diode that emits light onto the air sample that was collected inside of the sensor. This causes the particles to scatter inside the light, which is directly related to the concentration of particles inside the air.

This sensor was appropriate for this design because it can detect particulate matter (PM) suspended in the air that has been emitted from industrial activities and vehicles, such as dust, smoke, pollen, and pollutants. This is crucial for assessing air quality because PM impacts human health and the environment. This sensor was also low-cost and able to be used outdoors, which were two requirements from Blueprint Geneva of our sensor selections.

5.1.4. MQ-4 METHANE GAS SENSOR (BLUE WITH METAL CAP):

Model: MQ-4 *Manufacturer:* Flying Fish

The MQ-4 Methane Gas Sensor uses the chemical SnO_2, which has lower conductivity in clean air. When there is methane present, the surface of the sensing material interacts with the methane molecules, which makes the sensor's conductivity increase. The electrical resistance of the sensor thus changes as a result, and users can convert the change of conductivity with the output signal of gas concentration.

This sensor was appropriate for this design because it can detect methane. Landfills produce methane emissions due to the decomposition of waste, and monitoring the levels of methane is critical for the health of the individuals in the community. This sensor was also low-cost and able to be used outdoors, which were two requirements from Blueprint Geneva of our sensor selections.

5.1.5. SRAQ-G018(MQ-136) HYDROGEN SULFIDE GAS SENSOR (GREEN WITH METAL CAP):

Model: SRAQ-G018 *Manufacturer: L-Com*

Similar to the MQ-4 Methane Gas Sensor, the SRAQ-G018 Hydrogen Sulfide Gas Sensor uses the chemical SnO_2, which has lower conductivity in clean air. When there is hydrogen sulfide present, the surface of the sensing material interacts with the hydrogen sulfide molecules, which makes the sensor's conductivity increase. The electrical resistance of the sensor thus changes as a result, and users can convert the change of conductivity with the output signal of gas concentration.

The MQ-136 sensor is widely used in H_2S gas alarms to detect the presence of this toxic gas. The MQ-136 sensor has good sensitivity to H_2S gas and is the most reliable low-cost H_2S sensor that fits the low-cost, long-life, outdoor sensor requirements of Blueprint Geneva.

Compared to the other sensors that we ordered, most hydrogen sulfide sensors were extremely expensive. As a group, we assumed that low cost would be of a higher priority to our community partner, so we were not going to order a hydrogen sulfide sensor. After discussion with our community partner, she mentioned that it was of high importance, and thus the high cost would not be as important. We moved forward with this particular sensor because it is of a lower cost, while still providing data at a level of accuracy that we need.

5.1.6. ANEMOMETERS (WIND SPEED AND WIND DIRECTION):

Model: B082YC65W2 & B082MJDG2X *Manufacturer: Yosoo*

Both of these sensors are weatherproof, as they are high-quality material with high hardness and external electroplating and spray plastic treatment, which has good anti-corrosion and anti-erosion performance. These sensors also have a 0-70m/s range and 0.0875m/s resolution, high precision, accurate measurement and anti-electromagnetic interference treatment.

Although these sensors were purchased, after further discussion with our community partner and further refinement of our project scope, we decided to not move forward with incorporating these into our final design. It would have been interesting to collect this data, and we hope future groups do so, but our community partner expressed that it was not as high of a priority as other sensors since she can find meteorological data online. This was because we were reaching towards the end of the semester and were struggling to implement this into our design. Thus, in agreement with our community partner, we decided that for this iteration of our air quality monitor module, it is not imminent to measure wind speed and wind direction.Our community partner, as well as ourselves, preferred to focus on troubleshooting and perfecting our current prototype to ensure a deployment of a working prototype by the end of the semester. The decision to cut the wind speed and wind direction sensors was the best course of action for our group, as we were able to fully focus on refining our prototype and allowed us to deploy our prototype on time.

5.2. WIRELESS COMMUNICATION AND SENSOR INTEGRATION

5.2.1. SENSOR INTEGRATION TO PROTOTYPE

The sensors are directly plugged into a breadboard inside our system and wired accordingly due to the data sheets provided by each of the sensors. The breadboard, Featherboard, and sensors were all mounted vertically within the Stevenson screen, which is open to allow airflow into the device but keep out rain and snow. This can be seen in the images of our deployed module in Appendix E.

We are using the MCCI Catena 4410 Featherboard provided inside our IoT Lab Kit to power our system and to connect to The Things Network. Because the MCCI Catena 4410 only offered a 3V signal, we utilized a TL081 op-amp to convert the 3V signal to a 5V one that our sensors required, in conjunction with a 10K and two 4.7K resistors in series (6.7K). We had originally planned to use an Arduino Uno due to its larger power supply but struggled to connect it to The Things Network, which is why we changed our plan to utilize an op-amp instead.

5.2.2. SENDING & STORAGE OF SENSING DATA

We have sent the data from our sensors into a data packet that went to TTN successfully. Based on our conversations with Blueprint Geneva, data will be collected directly from TTN in a JSON format onto a .txt file. We both decided to collect data in this way because having the raw data for Blueprint Geneva to look at would be best for the team of data analysts. We will instead write instructions on how to work with the JSON that is exported onto the .txt file so that the team of analysts understand how the data is used. In order to collect the data from TTN to this .txt file, we are using Paho and a modified version of the Python script provided in the training lab. When running this script, we are able to collect data onto a .txt file.

From the sensor to TTN, the data is scheduled to be sent every 90 seconds. This was chosen because based on some tests that we conducted in Ithaca of our sensor running for long periods at a time, sometimes the data would fail to upload from our sensors. When we were doing these tests, we had scheduled the data to send every ten minutes, and if the data didn't send, we would be missing out on some of the data within the time period. We also chose this time because having the data to compare every 90 seconds allows for a data analyst to be able to see spikes more clearly in the data and match them with meteorological data to determine how the chemical concentrations have changed in conjunction with the weather conditions.

Because the Python script is running locally within our computers, we will provide Blueprint Geneva with the script and instructions on how to look at the .txt file in our user guide, so that our community partner can collect the data onto her own computer and use the data as she wishes. We had originally discussed that we would collect the data onto Google Drive for our community partner, but were worried with the longevity of that process because of our Cornell Google Drives expiring when we graduate.

5.3. HARDWARE

5.3.1. SENSOR HOUSING

Our sensor module is housed within a Stevenson screen which allows for air to freely flow through the casing while keeping the items housed inside dry. This permits us to safely store the sensors while also getting accurate readings from the ambient air. Although Stevenson screens can be purchased off the shelf, ours is 3D printed out of PLA and customizable and is highly modular to allow it to be easily put together and taken down. Our initial housing design featured custom mounting for our solar panel, however, since we have moved away from integrating a solar panel, our final housing for our first prototype no longer features mounting for a solar panel. All of these different versions are in our User Guide for future groups to use. See appendix C (section 7.3.) for a render of our module housing as well as the different iterations of the housing. The 3D printed Stevenson screen was originally red, because of the filament available. Due to concerns that it would melt, we painted it white. The odors and fumes will not impact the readings from our sensors as we chose a paint that does not release any sort of particulates, and we also tested it. Generally though, PLA has a melting point of 170 degrees celsius, so keeping the casing red would not have been an issue.

5.3.2. SOLAR PANEL

In order to make our sensor module self-sufficient, and requiring minimal maintenance, we purchased a 6V, 2W solar panel to be placed at a 35 degree angle on top of the Stevenson screen. Studies have shown that the best angle to position a solar panel, in order for it to achieve the most power, is between 30 and 45 degrees, justifying our placement at 35 degrees. This solar panel would have powered the entire module, however, we had some complications with the adapter as well as last minute changes to our boards which prompted us to no longer use a solar panel to power our module. Instead, Waterloo Container will be providing us with access to a power supply in order to power our module.

5.4. LABORATORY TESTING

Calibration of the sensors is critical, as it provides the user with accurate data Each sensor selected for this project requires specific pre-deployment calibration procedures to optimize performance and data accuracy:

1. Adafruit BME 680 AND Sparkfun BME688 Sensors: Both sensors measure temperature, humidity, and barometric pressure and require a stabilization period after the initial power-up. For accurate calibration, these sensors should be operated in a controlled environment for 48 hours to stabilize sensor output and verify data consistency against known environmental standards.
2. Adafruit PMSA003I Air Quality sensor: This sensor utilizes a laser scattering method that requires calibration against known particle concentrations. Before deployment, it should be tested in a dust-controlled chamber where particle sizes and concentrations are known and stable. Calibration will involve adjusting the sensor's response to match the controlled concentrations precisely.
3. MQ-4 Methane Gas Sensor and SRAQ-G018 (MQ-136) Hydrogen Sulfide Gas Sensor: Both sensors are based on SnO_2 semiconductor technology and require a burn-in period to stabilize the sensing element. These sensors should undergo a 72-hour continuous operation in clean air to achieve baseline stability. Post burn-in, they should be exposed to precise concentrations of methane and hydrogen sulfide, respectively, to calibrate their response curves. This process adjusts the sensors' output to known gas concentrations to ensure that the field readings are accurate.

In addition to these pre-deployment calibrations, we conducted practical testing of our sensor modules inside and outside one of the buildings on campus. This testing aimed to validate the sensor responses under different environmental conditions. For the methane and hydrogen sulfide sensors, which output voltage readings, we collected data in a controlled indoor environment and then in an outdoor setting. By comparing these readings, and using them as a baseline, we established a calibration curve that translates voltage differences into meaningful gas concentration values in parts per million (ppm). This calibration curve provided a conversion factor, which we implemented in the decoder on The Things Network to convert the raw voltage data into useful gas concentration measurements (See Appendix G Section 7.7). This conversion factor is critical for ensuring that the data transmitted by our sensors is accurate and actionable. Additionally, this factor can be adjusted in the decoder as needed upon sensor recalibration, ensuring the ongoing accuracy and reliability of the data over time.

Calibration of these sensors not only involves physical conditioning but also adjustment of the software algorithms that interpret the sensor signals into meaningful data. This will ensure that the sensors provide reliable data on the concentration of various pollutants aiding in rapid response to air quality concerns.

5.5. ROLE OF COMMUNITY PARTNERSHIP IN TECHNOLOGICAL DEVELOPMENT

In our initial conversations, our community partner in Blueprint Geneva gave us an extensive list of chemical compounds that would affect the odor in the area. She wanted to include all of these into our module, but we did not think it was realistic, so we worked with her to identify the most important chemical compounds. These compounds were methane, particulate matter, hydrogen sulfide, and general VOCs. Because the final vision of the project is to ensure that the modules that we are creating are easily replicable, we wanted to ensure that the sensors that we chose are easily procurable. During our conversations with our community partner, we made sure that both their needs were being met, but also that we would produce something within the scope of our skills.

However, a question that came up in our initial development in the sensor was, what does easily replicable mean? This was something that we needed to collaborate heavily with our community partner to determine, as we wanted to ensure that whatever we were designing was able to be replicated by her in the future. These included choosing sensors that would be able to be easily connected to a breadboard with wires, which is simple for replication as long as we provide the correct documentation. We also decided to create a case that could be 3D printed, because our concern with buying one online and modifying it was that there would be a potential for the supplier to stop selling that particular product.

Also, when we were encountering technical difficulties, we turned to our community partner to assist with what she thought would be the most reasonable areas to focus on. We had wanted to include an anemometer to determine the correlation between wind speed, wind direction, and the concentration of chemical compounds in the air, but struggled with being able to include it due to the power it required. After contacting our community partner, she mentioned that it was low priority and that instead, we should focus on trying to develop the module to determine the chemical compounds, because she could use existing anemometers and weather data to determine the wind.

5.6. FIELD DEPLOYMENT

5.6.1. DATA ANALYSIS

In order to confirm our data's reliability, we collected data on May 8th, the day after we had installed our sensor, and compared it to data from the same day from Waterloo Weather Station (42.879658, -76.812537). For reference, there is around a three mile distance between the two locations. The data from Waterloo Weather Station gave us the temperature and pressure data at different hours across the day, allowing us to make a direct comparison. We made plots comparing the data between Waterloo Weather Station and our sensor. These plots made it easy to determine that there is about a 3 degree Celsius difference between the two temperatures on our temperature plot (Appendix H, Plot 1) and a 0.5 kg/m^2 difference in pressure in our pressure plot (Appendix H, Plot 2). Furthermore, when we compare the lines in both plots, the gaps are almost consistent and the lines follow the same trend. Therefore, our environmental data appears to be reliable.

To verify the reliability of our sensors that detect chemical compounds, we kept in contact with the officers at Waterloo Container and asked them to report the days and times that they detected odor. Furthermore, according to what they said, there unfortunately hasn't been any strong odor since the day that we did the installation, only small scents here and there.

Since the H2S is one of the main sources of foul odor in the air, we analyzed the H2S data first and used the data from the night of 5/13 and the afternoon of 5/15. This was because these were two times that Mark, our contact from Waterloo Container, reported some smell. The plots can be found in Appendix H, Plot 3 and Plot 4. The average value of H2S is 3.36ppm on the night of 5/13. The average value of H2S is 3.4ppm in the afternoon of 5/15. According to the definition of Hydrogen Sulfide Health Hazards from US Department of Labor Occupational Safety and Health Administration (Appendix I), the "odor becomes more offensive at 3-5 ppm." This data makes sense in tangent with Mark's report, as the odor was faintly noticeable in Waterloo and was at the lower end of the spectrum reported in the Hydrogen Sulfide Health Hazards. On the other hand, Mark reported that the day after installation, there was no odor in the air. This is shown in Plot 5, where we plotted H2S concentrations around the time that Mark told us. The average is below the OSHA threshold of 3ppm. There are less data points recorded, however, as we were not running our code the whole day like we did on the days that Mark did record odor. This shows that our hydrogen sulfide data appears to be reliable.

After plotting the H2S data, we decided to look at the CH4 data at the same time. Among them, the average value of the night of 5/13 is 3.92 ppm, and average of the afternoon of 5/15 is 3.902 ppm (Appendix H, Plot 6 and 7). It is difficult to affirm the reliability of this data with the knowledge that there is odor, as there is less knowledge present about what levels of methane produce an odor. However, with the knowledge that atmospheric methane is 1.7 ppm, our sensor produces readings that are above the level of atmospheric methane, which would make sense considering that our sensor is located near the landfill.

5.6.2. MODULE DEPLOYMENT

5.6.2.1. FIRST SITE VISIT

Our team visited Waterloo Container on April 30th, 2024. The primary purpose of this trip was to meet the staff at Waterloo Container, understand the problems that they face with the landfill more, assess the mounting location, and evaluate the potential problems that we may encounter when mounting. Here, we looked at the location they picked and discussed our sensor's needs. One problem we discussed with the staff was power connectivity – we wanted an option to plug our sensor into a steady power supply. The second problem, which we will be working on testing throughout Week 7, is the workable distance between the LoRaWAN gateway and the sensor, especially passing through the metal walls of the factory.

The staff at Waterloo container with whom we spoke with provided us with a possible mounting location which is located directly downwind of the landfill within their facilities, preventing it from being taken by passersby. They also expressed displeasure with the smell that comes from the landfill, but since it was raining, the smell was not as strong. After our conversation, we agreed on the mounting location, they agreed to help us mount our gateway, as well as provide power to the gateway and the module itself. The module will be installed on a 10in by 10in wooden platform, provided by Waterloo container, on a fence that is within their facilities and directly downwind of the landfill. Appendix D (section 7.4) shows photos from our first site visit.

5.6.2.2. SECOND (FINAL) SITE VISIT

Our second, and final, visit to Waterloo Container was on May 7th, 2024. The goal of this visit was to install our module, and begin collecting air quality data. Prior to this visit we had assembled all necessary hardware equipment and tested data transmission from the module to our gateway and TTN through buildings and walls. Our tests on campus had successful results, so we expected installation at Waterloo container to be smooth. The staff at Waterloo container built a 10 inch by 10 inch wooden platform atop of a wired fence within their premises that faces directly towards the landfill. They also provided us with access to a 120V power outlet (where we brought an adapter cable to power our module) and an ethernet cable to plug in our gateway (provided by Blueprint Geneva). The installation process started off smoothly, but we were encountering issues with sending data through the gateway. We ended up staying longer than initially expected to troubleshoot the issue. It turned out that the ethernet connection they had provided us was faulty. Once we addressed this issue the rest of the installation was very smooth and resulted in successful transmission of data from our module to the TTN every two (2) minutes. See appendix E (section 7.5) for photos of our second site visit. These include pictures documenting the installation process, as well as the prototype itself after it was successfully installed.

6. PLAN FOR NEXT STEPS

6.1. FUTURE WORK

Although we worked hard to deliver an air quality sensor module that aligned with our community partner's vision, due to some setbacks we were not able to incorporate all aspects requested into our final prototype. Thus, we have some recommendations for future work on this module. One of the requests of our community partner was for a given sensor module to be able to be mounted anywhere, in other words, the module would have to be self sufficient. Initially we had hoped to include a solar panel in our deployed prototype, but due to the time constraints of the project, we had to remove the solar panel from our final design. It would be good for future groups to incorporate a solar panel, or perhaps another way to power the module without it needing to be plugged into an outlet.

Initially, our community partner mentioned that she wanted to include wind speed and wind direction sensors to be able to track the direction that particulates the sensor reads are coming from. Again, we purchased the necessary equipment to install these sensors, but our community partner let us know that it was the lowest priority item to complete, and that she could collect that data from meteorological data on the web, and thus we would not need to include these in our prototype. Since we already have these items purchased, it would be beneficial for future groups to integrate these sensors into future iterations.

Additionally, our community partner had a large array of compounds to be detected, as listed in section 5.1. Future groups should consider integrating more sensors, and sensors of higher quality into future versions of this module.

Lastly, it would be greatly beneficial to create more of these modules, so that they might be placed around Seneca Meadows Landfill and odor concentrations and their movement can be tracked accurately. It would also be good to have one of these modules be in a forest, where air is much cleaner, relatively, to serve as a control when comparing to the data collected by the modules that are closer to the landfill.

6.2. BARRIERS

Our project was a new project for this semester, and starting from scratch without any previous work was a barrier for us because we needed a little more time to establish the project scope and understand which results were important to obtain from the sensor device. We had a miscommunication with our community partner in the beginning of this project about what to prioritize in our sensor design, which hindered us in our ability to fully understand what our goals with our community partner was in creating the device. We were initially confused and prioritized low-cost, wind data, and replicability, but our community partner was less concerned about the cost and more concerned about the chemical compound detection. However, this was resolved around Week 3 of our project during a meeting with our community partner.

Other barriers that we encountered were hardware based. We had initially started our project with the plan of using an Arduino because we had many different sensors that required a higher voltage than the one that the Featherboard could provide. There were also sensors that were optimized to work with Arduino, which led to our original decision of using an Arduino. However, we had a lot of difficulty getting the Arduino to connect to LoRaWAN because it does not have the LoRa module on it. Thus, we made the decision to switch to using the MCCI Catena 4410 Featherboard that we used in class in our training labs. However, a barrier that came up with switching to the Featherboard meant that we needed to amplify our voltage and redevelop some of the sensor code. To get enough voltage for the sensors that required 5V, we used an operational amplifier. In our circuitry, we used a 9V battery as the power and ground of the voltage common collector (V_{cc}). We also used a 10K ohm resistor and a 6.7K ohm resistor (since we didn't have the correct resistor values, we used a 2K ohm and 4.7K ohm resistor in series). Another hardware issue that we faced was finding a way to accurately calibrate our data for the collection of methane and hydrogen sulfide data. We attempted to find a laboratory on campus that would allow us to run some tests to calibrate our sensor, but struggled with doing so. Instead, we ran the data outdoors in Ithaca, which we thought would be a more controlled environment with less methane and hydrogen sulfide concentrations.

Finally, the last barrier that we encountered was on the day of deployment. Waterloo Container generously provided us with a quick way to mount, but we were having trouble connecting to TTN successfully. Our device kept sending join requests, but it wouldn't pass through. We eventually realized that it was because the ethernet cable that they provided us wasn't actually connected to ethernet, which slowed down our deployment during our second site visit.

6.3. RECOMMENDATIONS

Given that our community partner wants to create and deploy more versions of our sensor module in the near future, and also given that this project is new this year, we decided to create a User Guide for future users to be able to replicate our design. One of the motivators behind this decision to create thorough documentation was that during the beginning stages of this project we felt like we had very little guidance and direction for the project. During these beginning stages, we found that some people who had used similar sensors to the ones we would use left behind scripts and documentation for how to use and wire the sensors. Taking inspiration from the concept of creating thorough guidance for future groups as well as our community partner, we created our Usage Guide, linked below for our community partner to reference and recreate our module. In our report, this can also be found in Appendix J (section 7.8): https://docs.google.com/document/d/1TMZZNGCIHwsBH1pM-1c0YCPtGI07GgFws_pcTUm6mzQ/edit?usp=sharing

Other suggestions/recommendations that we would have for future groups, as well as other users of this module would be to ensure regular communication with the community partner to make sure that the group and community partner are on the same page and have the same goals. From a hardware perspective, it would be beneficial if the Stevenson screen was ¼ inch wider and longer. A new Stevenson screen can either be purchased off the shelf, or also our CAD (.STEP) files can be modified. Additionally, since there were quite a few changes to our wiring setup, we suggest future groups have better wire management and use stiffer wires. Lastly, when installing at an existing facility, it is important to go within working hours as to respect the employees' and staff's time.

7. APPENDICES

7.1. APPENDIX A - WIRING

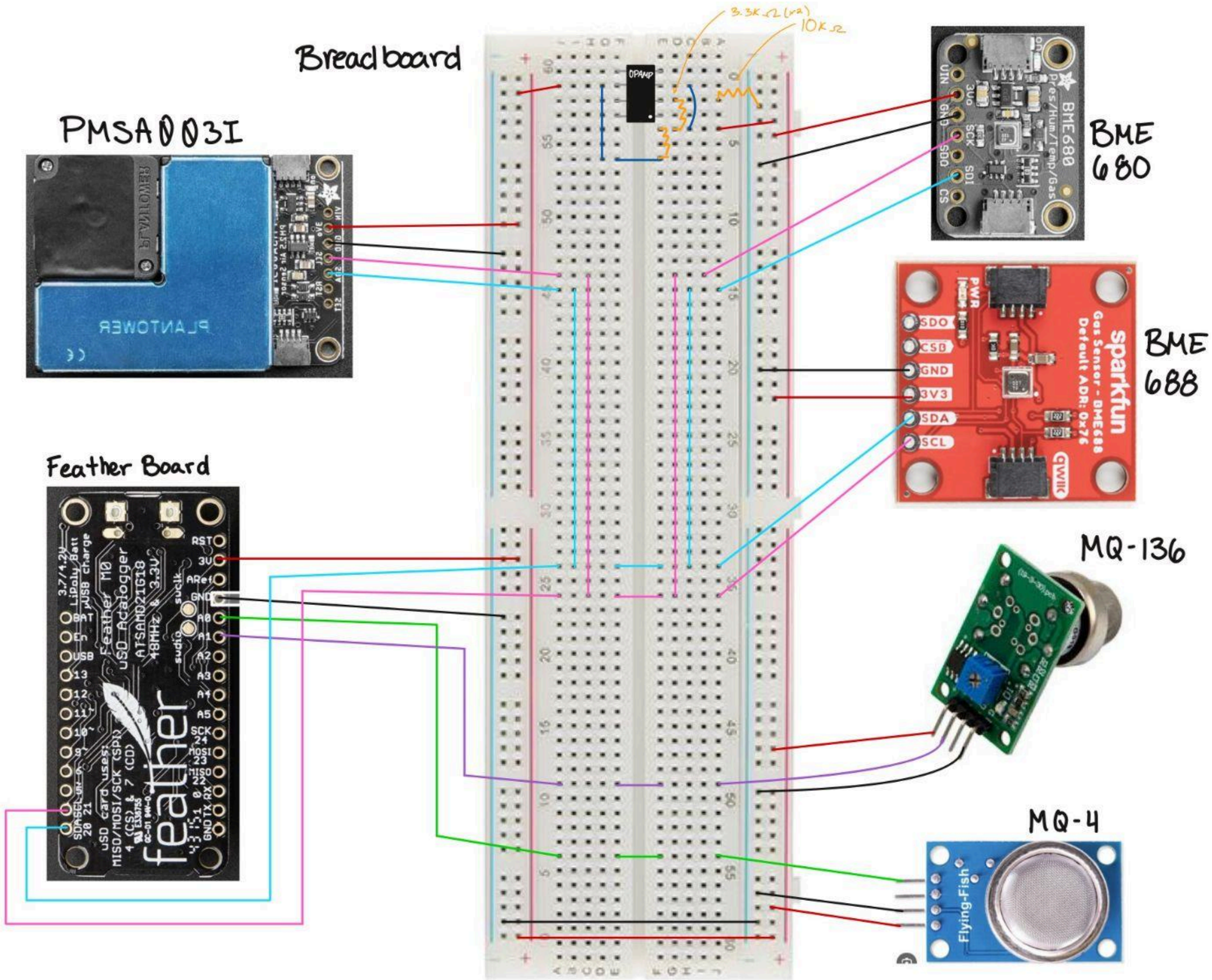


Figure 1: Wiring Diagram - Simplified

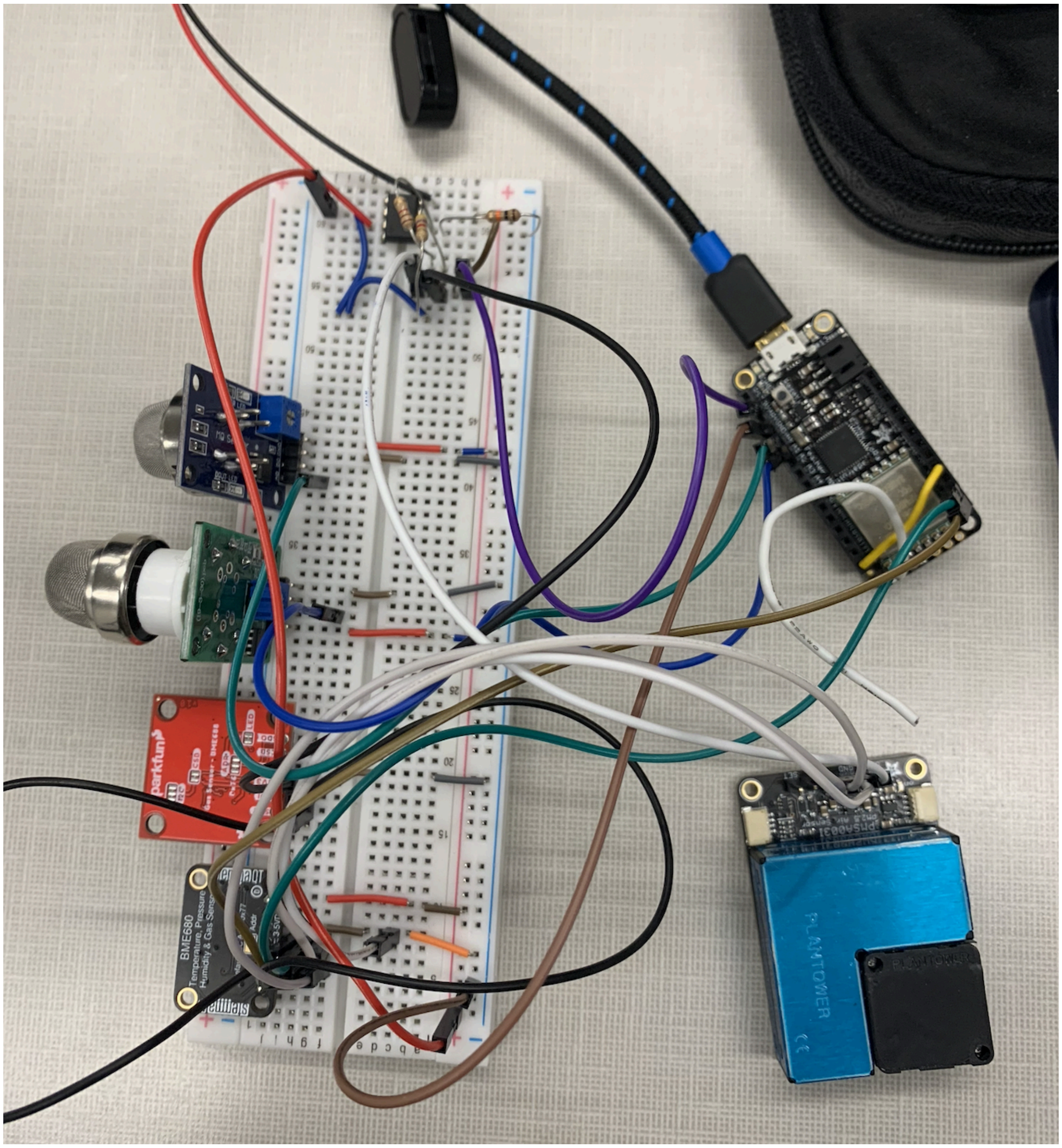


Figure 2: Final Wiring

7.2. APPENDIX B - BILL OF MATERIALS

Name	Supplier	Part Number	Unit Cost	Quantity	Total Cost ($)
Feather board	IoT Class	N/A	$0.00	1	$0.00
4PCS Breadboards Kit Include 2PCS 830 Point 2PCS 400 Point Solderless Breadboards for Proto Shield Distribution Connecting Blocks	Amazon	N/A	$8.99	1	$8.99
5V DC Supply 0-5V Output Anemometers Wind Speed Monitoring Sensor Outdoor Weather Station CALT	Amazon	N/A	$66.90	1	$66.90
Op Amp	Cornell		$0.00	1	$0.00
9V Battery	Cornell		$0.00	1	$0.00
10 kOhm resistor	Cornell		$0.00	1	$0.00
6.8 kOhm resistor	Cornell		$0.00	2	$0.00
BME680 Gas, Humidity, Pressure, Temperature Sensor Qwiic, STEMMA QT Platform Evaluation Expansion Board	DigiKey	N/A	$18.95	1	$18.95
5pcs MQ4 MQ-4 Methane Gas Sensor Natural Coal Co Methane Detector Module for Arduino Raspberry Pi ESP8266 MQ4 5V DC (MQ-4)	Amazon	N/A	$11.99	1	$11.99
SparkFun Environmental Sensor - BME688 (Qwiic)	SparkFun	N/A	$24.95	1	$24.95
Hydrogen Sulfide Gas Sensor Module, 10-200 ppm, Analog and TTL level Output, MQ136 Sensing Element	L-com	N/A	$113.99	1	$113.99
Stevenson Screen	Cornell RPL	N/A	0	1	0
				Total	**$245.77**

Figure 3: Bill of Materials – Items included on final prototype

Included In prototype?	Name	Supplier	Unit Cost	Quantity	Total Cost ($)	Notes/Purpose
☑	Feather board	IoT Class	$0.00	1	$0.00	
☑	4PCS Breadboards Kit Include 2PCS 830 Point 2PCS 400 Point Solderless Breadboards for Proto Shield Distribution Connecting Blocks	Amazon	$8.99	1	$8.99	Breadboard
☑	5V DC Supply 0-5V Output Anemometers Wind Speed Monitoring Sensor Outdoor Weather Station CALT	Amazon	$66.90	1	$66.90	Anemometer - Wind Speed
☑	Op Amp	Cornell	$0.00	1	$0.00	
☑	9V Battery	Cornell	$0.00	1	$0.00	
☑	10 kOhm resistor	Cornell	$0.00	1	$0.00	
☑	6.8 kOhm resistor	Cornell	$0.00	2	$0.00	
☑	BME680 Gas, Humidity, Pressure, Temperature Sensor Qwiic, STEMMA QT Platform Evaluation Expansion Board	DigiKey	$18.95	1	$18.95	environmental condition sensor
☑	5pcs MQ4 MQ-4 Methane Gas Sensor Natural Coal Co Methane Detector Module for Arduino Raspberry Pi ESP8266 MQ4 5V DC (MQ-4)	Amazon	$11.99	1	$11.99	methane and CO sensor
☑	SparkFun Environmental Sensor - BME688 (Qwiic)	SparkFun	$24.95	1	$24.95	environmental and gas sensor
☑	Hydrogen Sulfide Gas Sensor Module, 10-200 ppm, Analog and TTL level Output, MQ136 Sensing Element	L-com	$113.99	1	$113.99	Hydrogen Sulfide sensor
☑	Stevenson Screen	Cornell RPL	0	1	0	3D print from RPL
☑	6V 2W Solar Panel - ETFE - Voltaic P126	Adafruit	$20.95	1	$20.95	Solar Panel
☑	3M Arduino UNO USB Data Sync Cable for Arduino Mega 2560 Rev 3 R3 Microcontroller	Amazon	$7.99	1	$7.99	Arduino cable
☐	Adafruit PMSA003I Air Quality Breakout - STEMMA QT / Qwiic	Adafruit	$44.95	1	$44.95	PM sensor
☐	Wind Anemometer, 0-70m/s Environment Signal Output Pulse Type Three Cups Wind Sensor, 5-30V DC Power Supply	Amazon	$36.41	1	$36.41	Wind Speed
☐	SD card reader	Adafruit	$7.50	2	$15.00	recording data
☐	Anemometer Wind Meter, Wind Direction Sensor Transmitter, 360 Degree Wind Transmitter Transducer Wind Speed Anemometer Instruments for Ambient Weather(0-3V(5V Power Supply))	Amazon	$33.44	1	$33.44	Anemometer - Wind Direction
☐	3.8 / 1.3mm or 3.5 / 1.1mm to 5.5 / 2.1mm DC Jack Adapter Ca	Adafruit	$1.50	1	$1.50	Adapter cable
				Total:	**$406.01**	

Figure 4: Full BoM — all items that were ordered, and intended to be used in future versions of module

Linked here is our complete BoM sheet: https://docs.google.com/spreadsheets/d/1dDi027faNzxTstRXUI8kxGjVRQHH-l1nuQI9Zu6NyQE/edit?usp=sharing

7.3. APPENDIX C HARDWARE RENDERS & PROTOTYPES

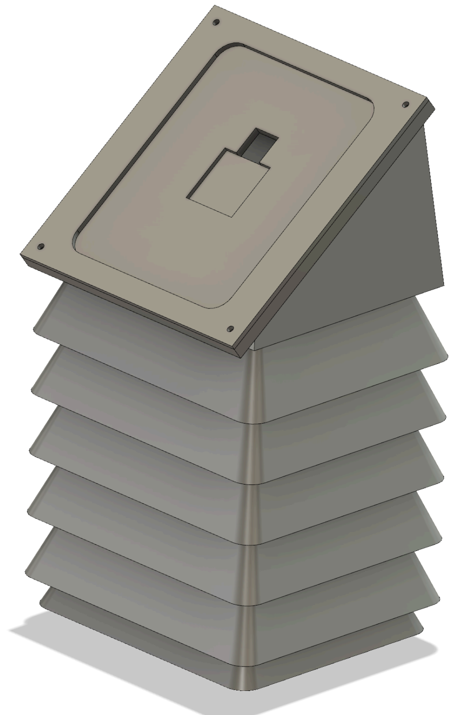


Figure 5: Stevenson screen render — this was the first iteration of the module

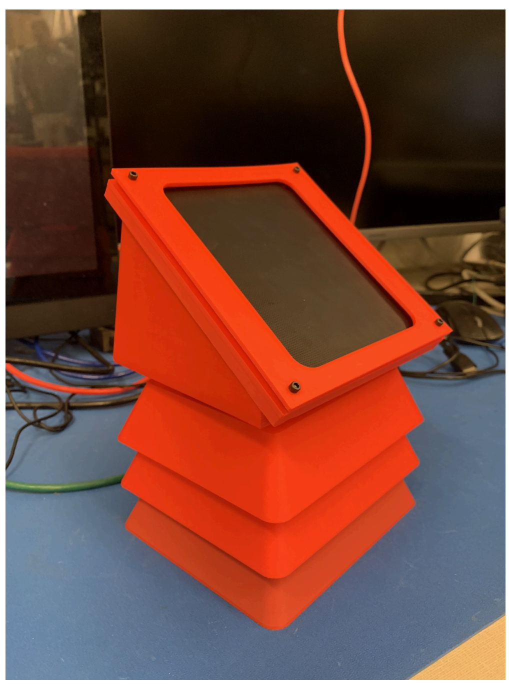


Figure 6: Stevenson screen initial prototype

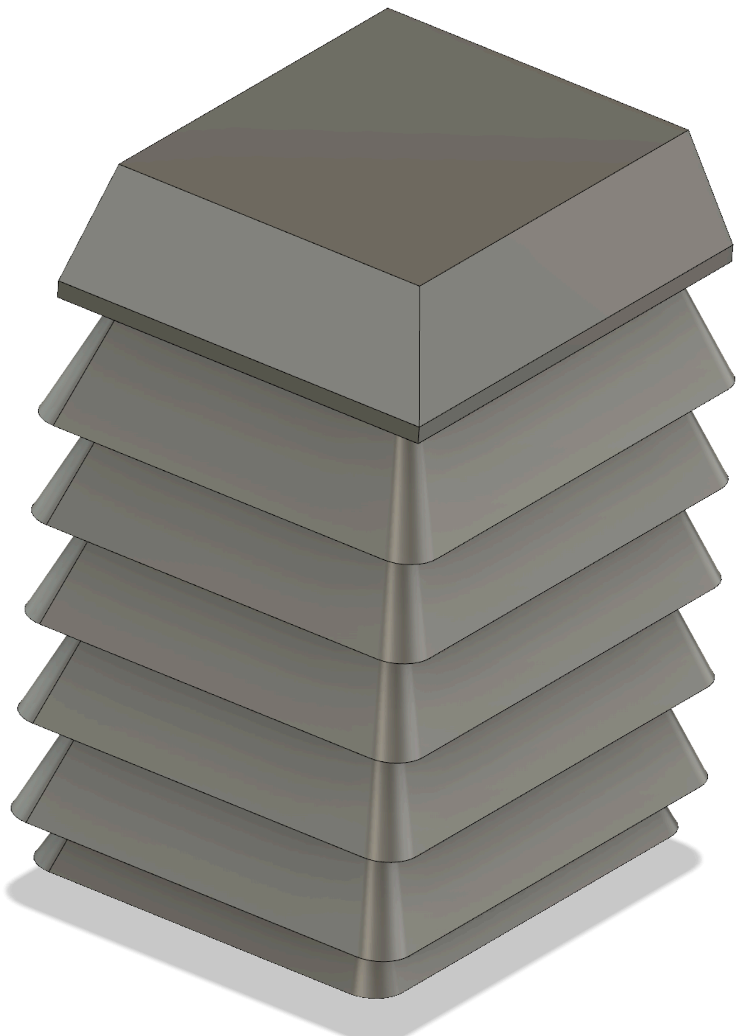


Figure 7: Stevenson screen render – deployed prototype

7.4. APPENDIX D - FIRST SITE VISIT

Figure 8: Location of Waterloo container relative to Seneca Landfill on a map, notice they are across the street from each other

Figure 9: View of Seneca Landfill from proposed mounting location at Waterloo container, April 30th 2024 (Rainy day).

Figure 10: Proposed Mounting Location.

7.5. APPENDIX E - FINAL DEPLOYMENT (2ND SITE VISIT)

Figures 11, 12, 13: Installation Process

Figure 14, 15, 16: Deployed Prototype

7.6. APPENDIX F - DATA COLLECTION PROCESS

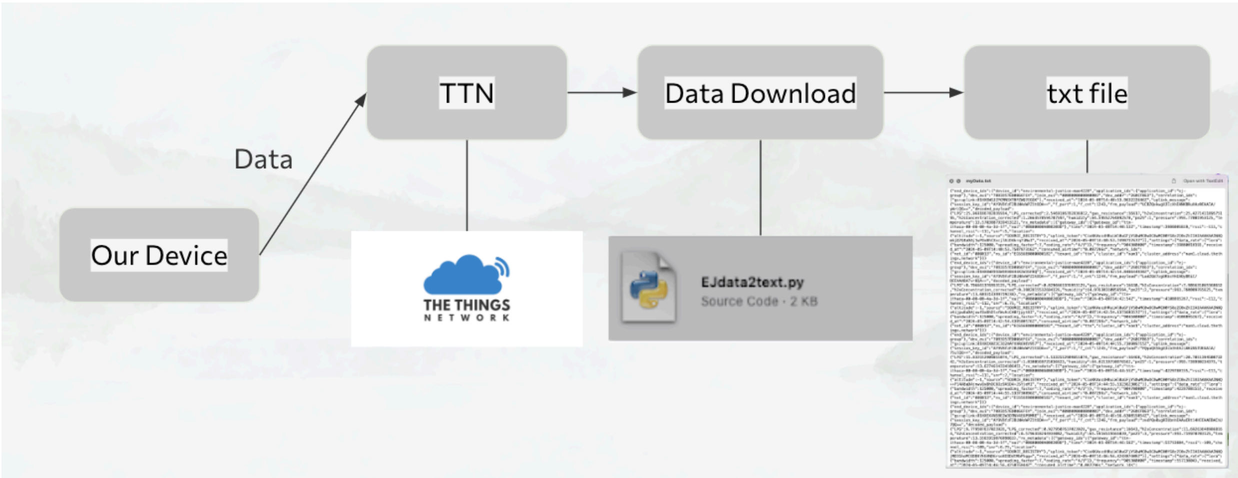


Figure 18: Data Collection Process

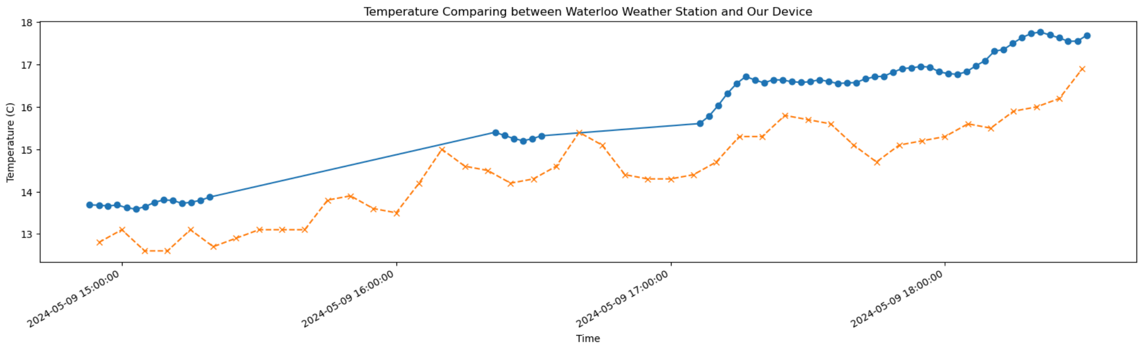
7.7. APPENDIX G - The Things Network Decoder

Formatter code *

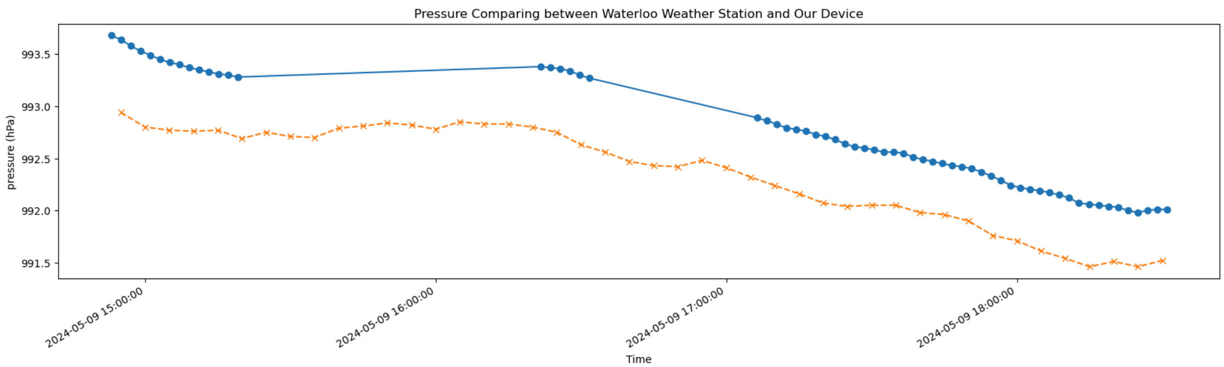
```
input.bytes.forEach(function (byte, index) {
    view.setUint8(index, byte);
});

var data = {};

data.temperature = view.getFloat32(0, true);  // From byte 0
data.humidity = view.getFloat32(4, true);     // From byte 4
data.pressure = view.getFloat32(8, true);     // From byte 8
data.gas_resistance = view.getFloat32(12, true);  // From byte 12
data.LPG = view.getFloat32(16, true);         // From byte 16
data.pm25 = view.getFloat32(20, true);        // From byte 20
data.h2sConcentration = view.getFloat32(24, true); // From byte 24

// Calibration baselines and scale factors
var baseline_LPG = 0.5; // Assumed baseline for fresh air LPG in parts per million
var baseline_H2S = 0.1; // Assumed baseline for fresh air H2S in parts per million
var scale_factor_LPG = 10; //  scale factor for LPG
var scale_factor_H2S = 20; //  scale factor for H2S

// Applying calibration
data.LPG_corrected = (data.LPG - baseline_LPG) / scale_factor_LPG;
data.h2sConcentration_corrected = (data.h2sConcentration - baseline_H2S) / scale_factor_H2S;
```

Figure 19: TTN Decoder Screenshot

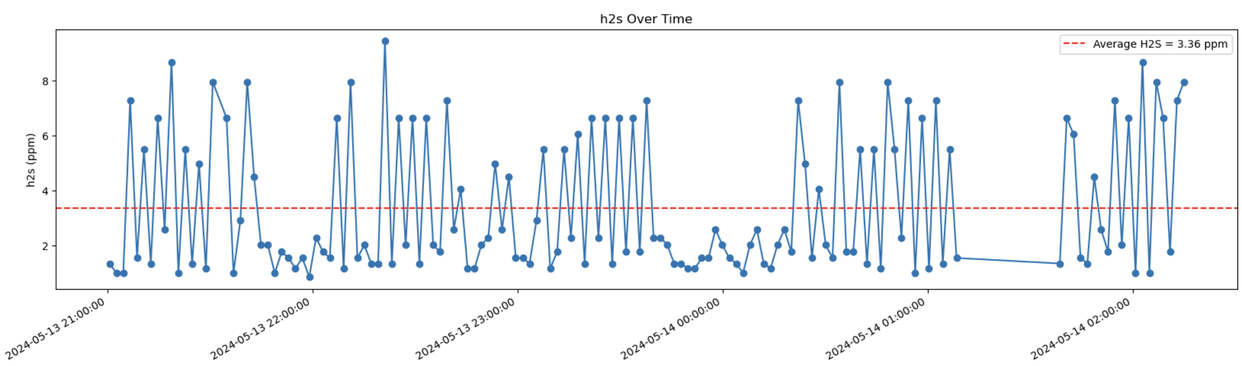
7.8. APPENDIX H - DATA PLOTS



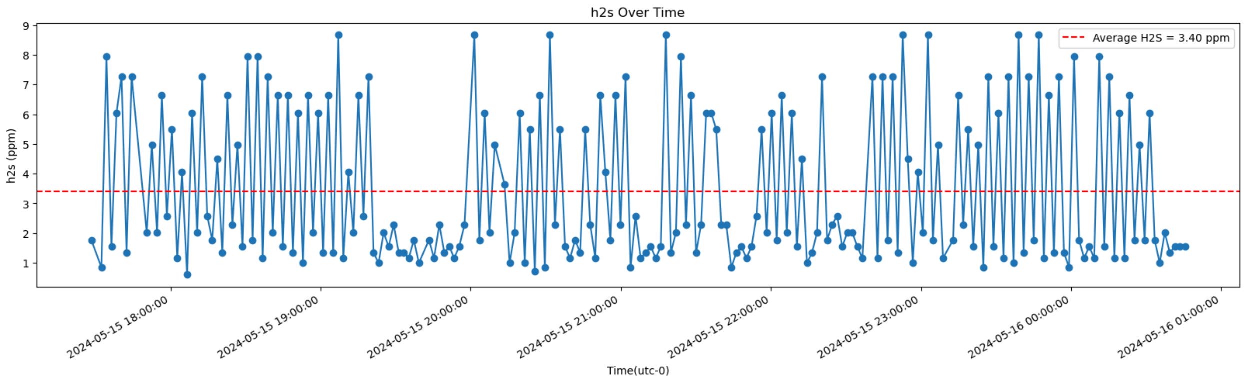
Plot 1: Temperature Comparing between Waterloo Weather Station and Our Device



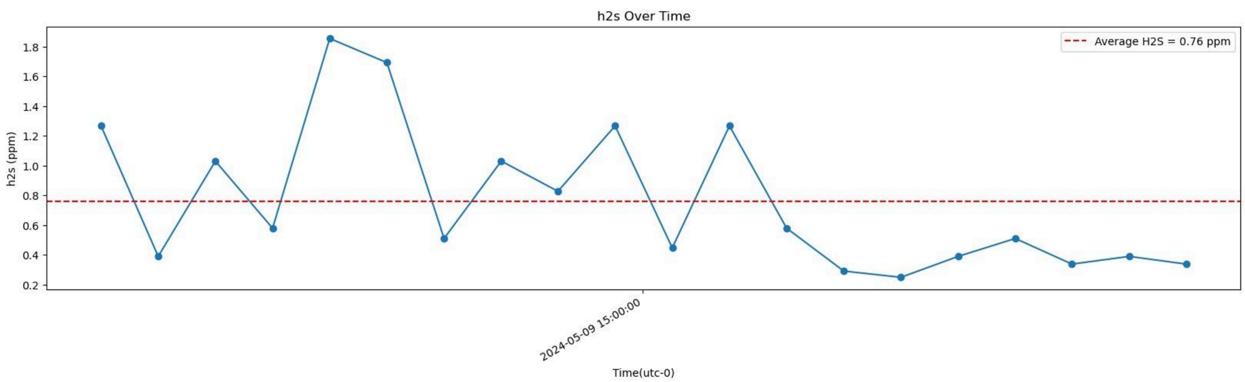
Plot 2: Pressure Comparing between Waterloo Weather Station and Our Device



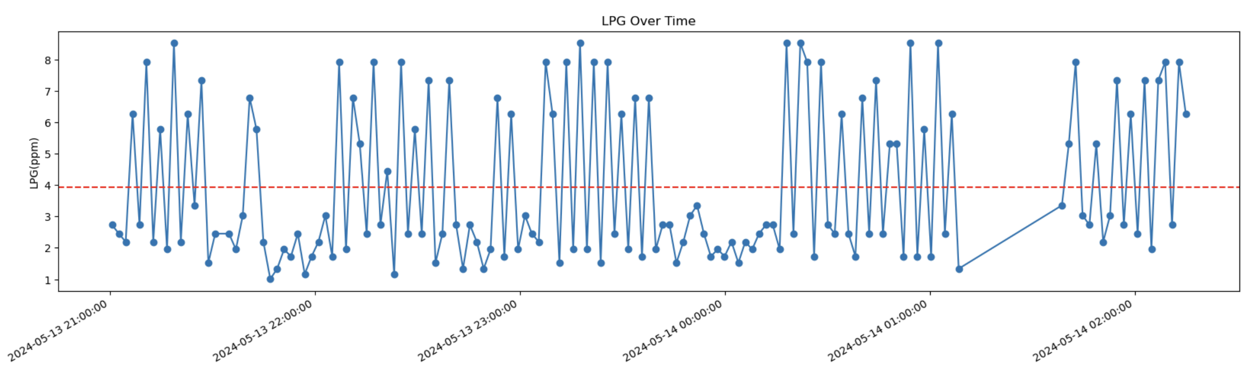
Plot 3: 2024.5.13 Night H2S Data



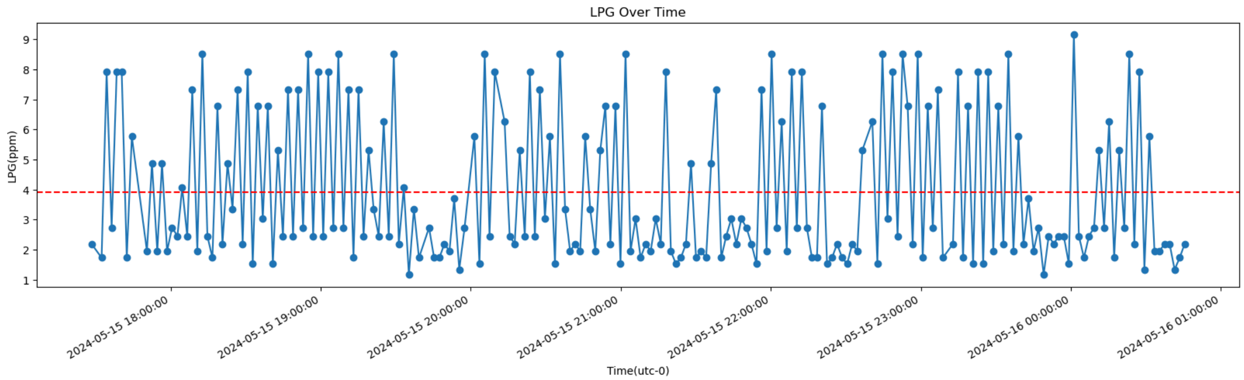
Plot 4: 2024.5.15 Afternoon H2S Data



Plot 5: 2024.5.9 Afternoon H2S Data



Plot 6: 2024.5.13 Night CH4 Data



Plot 7: 2024.5.15 Afternoon CH4 Data

7.9. APPENDIX I - HYDROGEN SULFIDE LEVELS

Concentration (ppm)	Symptoms/Effects
0.00011-0.00033	Typical background concentrations
0.01-1.5	Odor threshold (when rotten egg smell is first noticeable to some). Odor becomes more offensive at 3-5 ppm. Above 30 ppm, odor described as sweet or sickeningly sweet.
2-5	Prolonged exposure may cause nausea, tearing of the eyes, headaches or loss of sleep. Airway problems (bronchial constriction) in some asthma patients.
20	Possible fatigue, loss of appetite, headache, irritability, poor memory, dizziness.
50-100	Slight conjunctivitis ("gas eye") and respiratory tract irritation after 1 hour. May cause digestive upset and loss of appetite.
100	Coughing, eye irritation, loss of smell after 2-15 minutes (olfactory fatigue). Altered breathing, drowsiness after 15-30 minutes. Throat irritation after 1 hour. Gradual increase in severity of symptoms over several hours. Death may occur after 48 hours.
100-150	Loss of smell (olfactory fatigue or paralysis).
200-300	Marked conjunctivitis and respiratory tract irritation after 1 hour. Pulmonary edema may occur from prolonged exposure.
500-700	Staggering, collapse in 5 minutes. Serious damage to the eyes in 30 minutes. Death after 30-60 minutes.
700-1000	Rapid unconsciousness, "knockdown" or immediate collapse within 1 to 2 breaths, breathing stops, death within minutes.
1000-2000	Nearly instant death

Figure 20: Hydrogen Sulfide Health Hazards from US Department of Labor Occupational Safety and Health Administration

7.10. APPENDIX J - USAGE GUIDE

Usage Guide

MAE 4220/5220: Environmental Justice Project

Hardware

Materials

Below is a list of the materials included in our deployed prototype:

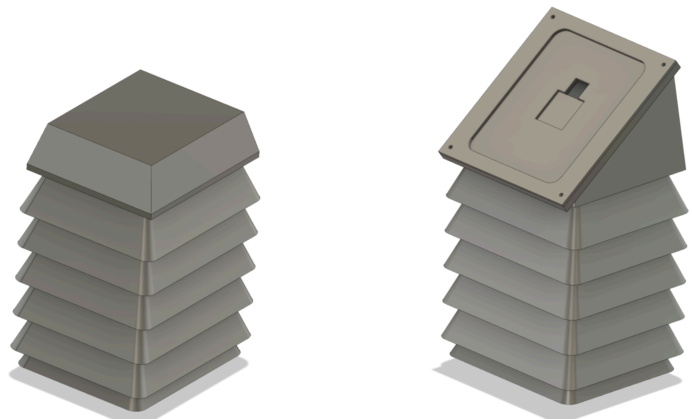
Name	Supplier	Part Number	Unit Cost	Quantity	Total Cost ($)
Feather board	IoT Class	N/A	$0.00	1	$0.00
4PCS Breadboards Kit Include 2PCS 830 Point 2PCS 400 Point Solderless Breadboards for Proto Shield Distribution Connecting Blocks	Amazon	N/A	$8.99	1	$8.99
5V DC Supply 0-5V Output Anemometers Wind Speed Monitoring Sensor Outdoor Weather Station CALT	Amazon	N/A	$66.90	1	$66.90
Op Amp	Cornell		$0.00	1	$0.00
9V Battery	Cornell		$0.00	1	$0.00
10 kOhm resistor	Cornell		$0.00	1	$0.00
6.8 kOhm resistor	Cornell		$0.00	2	$0.00
BME680 Gas, Humidity, Pressure, Temperature Sensor Qwiic, STEMMA QT Platform Evaluation Expansion Board	DigiKey	N/A	$18.95	1	$18.95
5pcs MQ4 MQ-4 Methane Gas Sensor Natural Coal Co Methane Detector Module for Arduino Raspberry Pi ESP8266 MQ4 5V DC (MQ-4)	Amazon	N/A	$11.99	1	$11.99
SparkFun Environmental Sensor - BME688 (Qwiic)	SparkFun	N/A	$24.95	1	$24.95
Hydrogen Sulfide Gas Sensor Module, 10-200 ppm, Analog and TTL level Output, MQ136 Sensing Element	L-com	N/A	$113.99	1	$113.99
Stevenson Screen	Cornell RPL	N/A	0	1	0
				Total	**$245.77**

Linked here is the spreadsheet with all the materials for the deployed prototype, as well as other purchased materials that could be integrated in future iterations:
https://docs.google.com/spreadsheets/d/1v7TjDxiowJ6HdjvsePG3u-fTjrRjHOI-s4wH6BQ8oVI/edit?usp=drive_link

Creating the Sensor

Link to Hardware Folder:
https://drive.google.com/drive/folders/18xwQQG_xWu48qTrQ_b0HBWQB3uSolMK4?usp=sharing
A few comments:

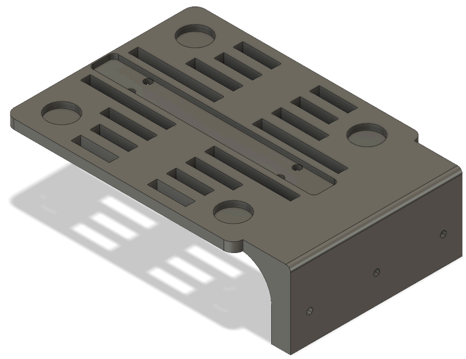
- The pieces fit together like Legos, some force is needed for them to be put together. No glue is necessary, but if you want extra security it is recommended to use some sort of epoxy or superglue. [Note: some glues are strong enough to melt through some 3D prints, I recommend using Gorilla Glue]
- There are different top pieces, and different bottom pieces. This depends on the mounting configuration (mounting on a vertical surface vs mounting on a horizontal surface)
- Depending on the configuration you choose, it should look something like this (no solar panel vs solar panel mount):



(left) No solar panel (right) solar panel

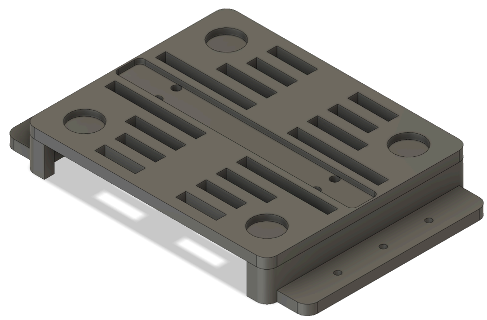
Instructions

1. Download the .STEP files
2. Using a 3D printer, print the following amounts of each file:
    a. For no Solar Panel configuration
        i. 1 Mounting_Plate
        ii. 1 Base_2_Plate
            1. This part has holes for M3 heat set inserts boards can be attached to it
        iii. 5 Middle_SS
        iv. 20 Connector
        v. 1 Top_NoSolarPanel
        vi. One of the following - depending if you are mounting horizontally or vertically
            1. Bottom_VertMount



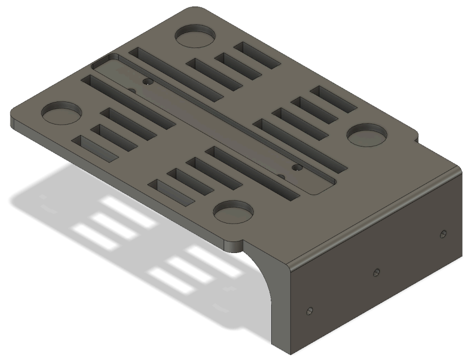
Bottom for vertical mounting

2. Bottom_HorzMount



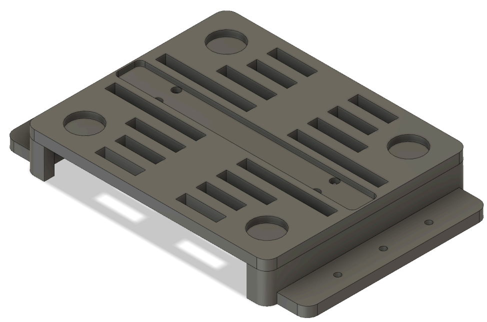
Bottom for horizontal mounting (the module at Waterloo container is mounted horizontally)

b. For Solar Panel configuration
   i. 1 Mounting_Plate
   ii. 1 Base_2_Plate
      1. This part has holes for M3 heat set inserts boards can be attached to it
   iii. 5 Middle_SS
   iv. 20 Connector
   v. 1 Top_Solar_Panel
   vi. 1 Lid_Solar_Panel_Holder
      1. This is to hold the solar panel in place
      2. Should be mounted with M3 screws to Top_Solar_Panel
   vii. 1 Lid_Solar_Panel_Mount
      1. This has a mounting spot for the solar panel that we purchased – see bill of materials (also linked here: https://www.adafruit.com/product/5366)
      2. Has holes for M3 heat sets in order to fasten Lid_Solar_Panel_Mount with M3 screws
      3. This part should be glued to Top_Solar_Panel
   viii. One of the following - depending if you are mounting horizontally or vertically – these can be mounted via wood screws to wooden surfaces
      1. Bottom_VertMount



Bottom for vertical mounting

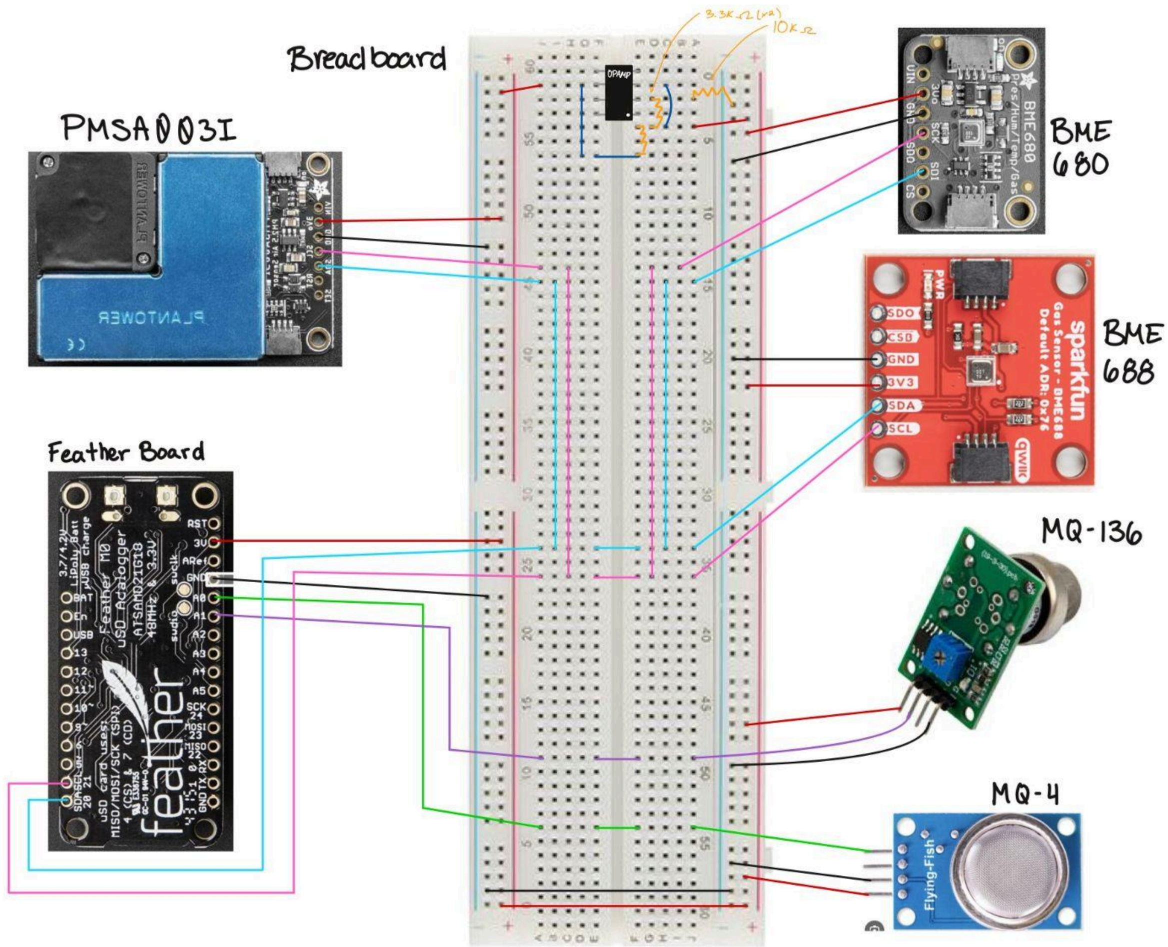
2. Bottom_HorzMount



Bottom for horizontal mounting (the module at Waterloo container is mounted horizontally)

3. Print time should be around 30 hours total for either configuration
4. Once printed you can put everything together and glue where necessary. Please reach out to the owner of the folder if you have any questions or issues!

Wiring Diagram:



Usage Guide

MAE 4220/5220: Environmental Justice Project

Hardware

Materials

Below is a list of the materials included in our deployed prototype:

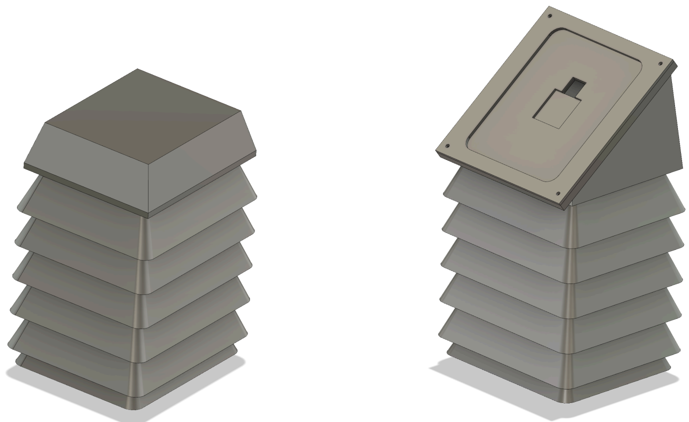
Name	Supplier	Part Number	Unit Cost	Quantity	Total Cost ($)
Feather board	IoT Class	N/A	$0.00	1	$0.00
4PCS Breadboards Kit Include 2PCS 830 Point 2PCS 400 Point Solderless Breadboards for Proto Shield Distribution Connecting Blocks	Amazon	N/A	$8.99	1	$8.99
5V DC Supply 0-5V Output Anemometers Wind Speed Monitoring Sensor Outdoor Weather Station CALT	Amazon	N/A	$66.90	1	$66.90
Op Amp	Cornell		$0.00	1	$0.00
9V Battery	Cornell		$0.00	1	$0.00
10 kOhm resistor	Cornell		$0.00	1	$0.00
6.8 kOhm resistor	Cornell		$0.00	2	$0.00
BME680 Gas, Humidity, Pressure, Temperature Sensor Qwiic, STEMMA QT Platform Evaluation Expansion Board	DigiKey	N/A	$18.95	1	$18.95
5pcs MQ4 MQ-4 Methane Gas Sensor Natural Coal Co Methane Detector Module for Arduino Raspberry Pi ESP8266 MQ4 5V DC (MQ-4)	Amazon	N/A	$11.99	1	$11.99
SparkFun Environmental Sensor - BME688 (Qwiic)	SparkFun	N/A	$24.95	1	$24.95
Hydrogen Sulfide Gas Sensor Module, 10-200 ppm, Analog and TTL level Output, MQ136 Sensing Element	L-com	N/A	$113.99	1	$113.99
Stevenson Screen	Cornell RPL	N/A	0	1	0
				Total	**$245.77**

Linked here is the spreadsheet with all the materials for the deployed prototype, as well as other purchased materials that could be integrated in future iterations: https://docs.google.com/spreadsheets/d/1v7TjDxiowJ6HdjvsePG3u-fTjrRjHOI-s4wH6BQ8oVI/edit?usp=drive_link

Creating the Sensor

Link to Hardware Folder:
https://drive.google.com/drive/folders/18xwQQG_xWu48qTrQ_b0HBWQB3uSolMK4?usp=sharing
A few comments:

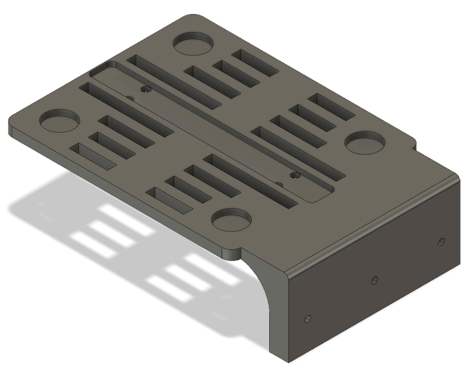
- The pieces fit together like Legos, some force is needed for them to be put together. No glue is necessary, but if you want extra security it is recommended to use some sort of epoxy or superglue. [Note: some glues are strong enough to melt through some 3D prints, I recommend using Gorilla Glue]
- There are different top pieces, and different bottom pieces. This depends on the mounting configuration (mounting on a vertical surface vs mounting on a horizontal surface)
- Depending on the configuration you choose, it should look something like this (no solar panel vs solar panel mount):



(left) No solar panel (right) solar panel

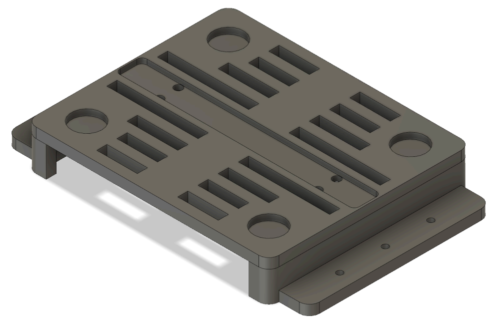
Instructions

1. Download the .STEP files
2. Using a 3D printer, print the following amounts of each file:
    a. For no Solar Panel configuration
        i. 1 Mounting_Plate
        ii. 1 Base_2_Plate
            1. This part has holes for M3 heat set inserts boards can be attached to it
        iii. 5 Middle_SS
        iv. 20 Connector
        v. 1 Top_NoSolarPanel
        vi. One of the following - depending if you are mounting horizontally or vertically
            1. Bottom_VertMount



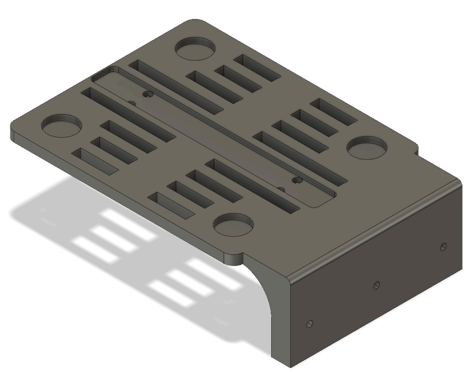
Bottom for vertical mounting

            2. Bottom_HorzMount



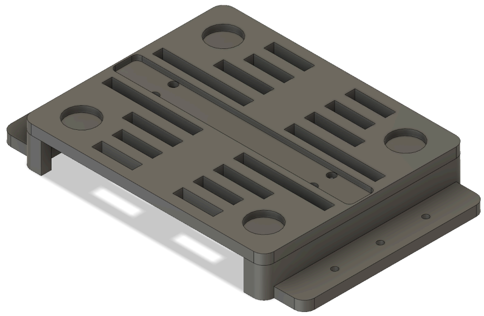
Bottom for horizontal mounting (the module at Waterloo container is mounted horizontally)

b. For Solar Panel configuration
    i. 1 Mounting_Plate
    ii. 1 Base_2_Plate
        1. This part has holes for M3 heat set inserts boards can be attached to it
    iii. 5 Middle_SS
    iv. 20 Connector
    v. 1 Top_Solar_Panel
    vi. 1 Lid_Solar_Panel_Holder
        1. This is to hold the solar panel in place
        2. Should be mounted with M3 screws to Top_Solar_Panel
    vii. 1 Lid_Solar_Panel_Mount
        1. This has a mounting spot for the solar panel that we purchased – see bill of materials (also linked here: https://www.adafruit.com/product/5366)
        2. Has holes for M3 heat sets in order to fasten Lid_Solar_Panel_Mount with M3 screws
        3. This part should be glued to Top_Solar_Panel
    viii. One of the following - depending if you are mounting horizontally or vertically – these can be mounted via wood screws to wooden surfaces
        1. Bottom_VertMount



Bottom for vertical mounting

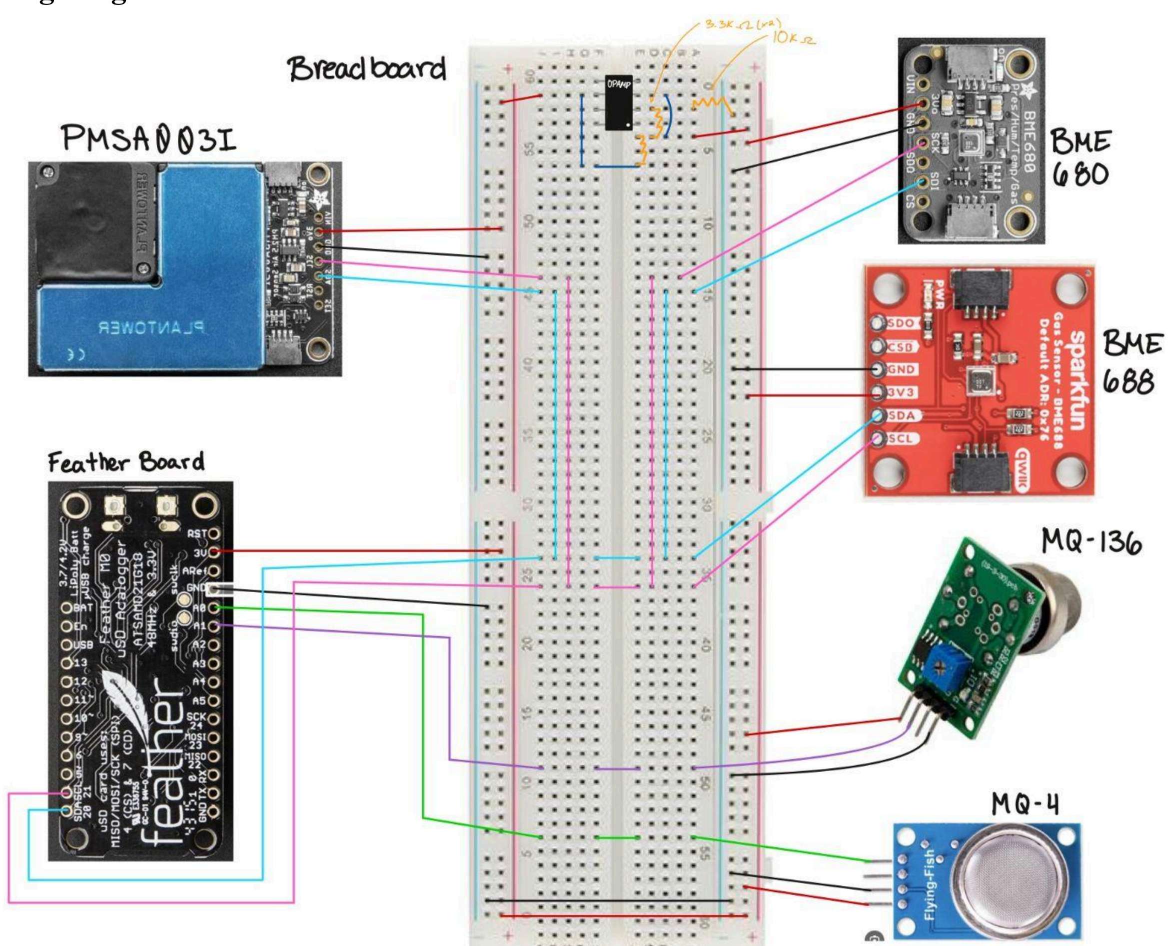
        2. Bottom_HorzMount



Bottom for horizontal mounting (the module at Waterloo container is mounted horizontally)

3. Print time should be around 30 hours total for either configuration
4. Once printed you can put everything together and glue where necessary. Please reach out to the owner of the folder if you have any questions or issues!

Wiring Diagram:



Software

Link to Software Folder:
https://drive.google.com/drive/folders/10azyncDescYgU5QaFbMo_eM5t8KJyMy5?usp=drive_link

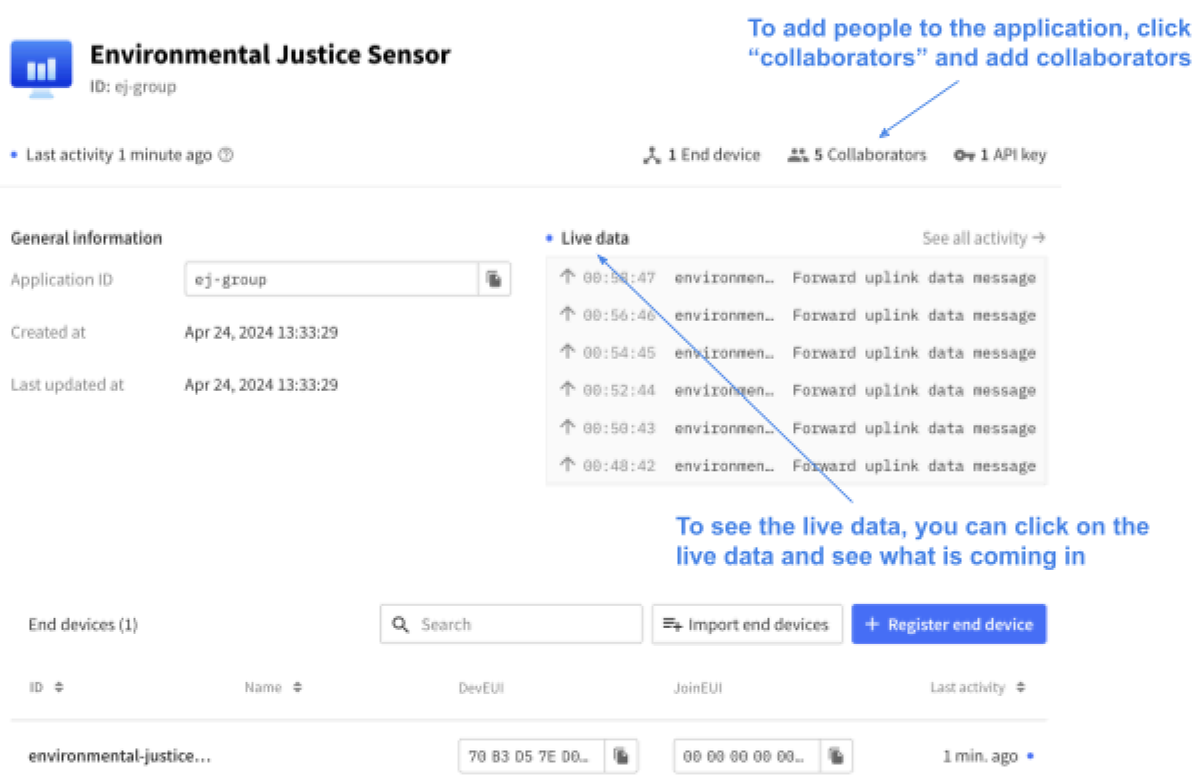
How to Use the Code:

1. Download all the files. Make sure the name of the folder containing all of the code is called ttn_otaa_highlevel, and that all the files are in the same folder.
2. Plug in the Featherboard into your computer. Use the program Arduino IDE to open the file ttn_otaa_highlevel.ino and upload the code onto the Featherboard by clicking first the verify button (the check mark on the upper left of the screen) and then upload (the right-pointing arrow next to the check mark on the upper left of the screen). This uploads the code into the Featherboard.
3. On TTN, check to see if the sensor has joined the network. If done correctly, data should begin to appear in the "Live Data"

Obtaining Data

Using TTN

If you are shared to the application, "*ej-group*" should appear as one of the applications. Once you open the application, you will see this screen which you can interact with. Theoretically, due to the data being downloaded onto a text file from the Python code (described below in the next section), there should not need to be much activity on TTN. However, for reference:



On the side panel in TTN, there are tabs called *End Devices*, *Payload Formatters, Integrations*, and *API keys*, which we either use or reference within the code to help the data transmission process occur. These should not need to be touched, but in the instance that they do:

- **End device:** talks to TTN to send and receive data wirelessly
- **Payload Formatters:** formats the raw data sent by the devices into a form that can be understood
    - **Uplink Code:**

```
function decodeUplink(input) {
    var buffer = new ArrayBuffer(input.bytes.length);
    var view = new DataView(buffer);

    input.bytes.forEach(function (byte, index) {
        view.setUint8(index, byte);
    });

    var data = {};

    data.temperature = view.getFloat32(0, true);  // From byte 0
    data.humidity = view.getFloat32(4, true);     // From byte 4
    data.pressure = view.getFloat32(8, true);     // From byte 8
    data.gas_resistance = view.getFloat32(12, true);  // From byte 12
    data.LPG = view.getFloat32(16, true);         // From byte 16
    data.pm25 = view.getFloat32(20, true);        // From byte 20
    data.h2sConcentration = view.getFloat32(24, true); // From byte 24

    // Calibration baselines and scale factors(if needed), require laboratory
environment with the chemical compounds to find the scale, and need to use clean
air research as a baseline.
    var baseline_LPG =; // Assumed baseline for fresh air LPG in parts per million
    var baseline_H2S =; // Assumed baseline for fresh air H2S in parts per million
    var scale_factor_LPG = ; //  scale factor for LPG
    var scale_factor_H2S = ; //  scale factor for H2S

    // Applying calibration
    data.LPG_corrected = (data.LPG - baseline_LPG) / scale_factor_LPG;
    data.h2sConcentration_corrected = (data.h2sConcentration - baseline_H2S) /
scale_factor_H2S;

    return {
        data: data,
        warnings: [],
        errors: []
    };
}
```

- **Integrations:** connects TTN to other apps or services, our sensor uses MQTT
- **API keys:** passwords that allow programs to talk to TTN to fetch data securely — it is used inside of the Python code used to get data onto a text file on the computer

To get data onto a text file on your computer:

1. Download the EJdata2text.py file and put it into the same folder as all of the software.
2. Follow the instructions for Installing Python, Python Basics, and Installing Paho. Make sure that when running Python code, you are in the correct folder before you run code. https://pages.github.coecis.cornell.edu/LPWAN-Training/training-labs/lab5
3. The name of the code file used to get data from our EJ Sensor is EJdata2text.py – run this test file, and it should say "Connected to broker" if it has been successfully run.
4. The data should be uploaded into a text file within the same ttn_otaa_highlevel folder with the name myData. This data can be uploaded into Google Drive, but utilizing software like Google Sync on the computer where the script is being run would be needed to sync the data automatically onto BluePrint Geneva's Google Drive.

Data Analysis

Understanding the Text File Data

The text file data is formatted specifically with a JSON object.

This is the example JSON object from the data we receive, the orange highlighted text will be the data that we need. The red highlighted text is the EST time, the orange highlight is time (UTC-0) and the data set.

2024-05-14 18:46:02 -
{"end_device_ids":{"device_id":"environmental-justice-mae4220","application_ids":{"application_id":"ej-group"},"dev_eui":"70B3D57ED0066FC4","join_eui":"0000000000000002","dev_addr":"260C7933"},"correlation_ids":["gs:uplink:01HXWMHG2TZFZJ2WYEXY15F0SZ"],"received_at":"2024-05-14T22:46:02.280464761Z","uplink_message":{"session_key_id":"AY9eUGRba09zePm4yU7ghg==","f_port":1,"f_cnt":3768,"frm_payload":"va+iQfrUnUIAQHhEAFjVRRLOaEIAAEBB7yOiQg==","decoded_payload":{"LPG":58.20124053955078, "LPG_corrected":5.770124053955078, "gas_resistance":6827, "h2sConcentration":81.07018280029297, "h2sConcentration_corrected":4.048509140014649, "humidity":78.91596984863281, "pm25":12, "pressure":993, "temperature":20.3358097076416},"rx_metadata":[{"gateway_ids":{"gateway_id":"ttn-ithaca-00-08-00-4a-3d-1f","eui":"00800000A00020DD"},"time":"2024-05-14T22:46:02Z","timestamp":1968216107,"rssi":-101,"channel_rssi":-101,"snr":9.25,"location":{"altitude":-1,"source":"SOURCE_REGISTRY"},"uplink_token":"CioKKAocdHRuLWl0aGFjYS0wMC0wOC0wMC00YS0zZC0xZhIIAIAAAKAAIN0Qq7DCqgcaCwiq04+yBhCFsIgiIPjPyZakngk=","received_at":"2024-05-14T22:46:02.071440389Z"}],"settings":{"data_rate":{"lora":{"bandwidth":125000, "spreading_factor":7, "coding_rate":"4/5"}}, "frequency":"904100000", "timestamp":1968216107},"received_at":"2024-05-14T22:46:02.075791003Z","consumed_airtime":"0.087296s","network_ids":{"net_id":"000013","ns_id":"EC656E0000000182","tenant_id":"ttn","cluster_id":"nam1","cluster_address":"nam1.cloud.thethings.network"}}}

Plotting Data (Python)

```
import csv
import re
from datetime import datetime
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
```

```
import pandas as pd

time_data = []
LPG_data = []

start_time = datetime.strptime('2024-05-13 21:00:37', '%Y-%m-%d %H:%M:%S')
end_time = datetime.strptime('2024-05-14 02:14:56', '%Y-%m-%d %H:%M:%S')

with open('myData1.txt', 'r', encoding='utf-8') as file:
    csv_reader = csv.reader(file)
    for row in csv_reader:
        time_field = None
        LPG_field = None
        for field in row:
            if 'received_at' in field:
                time_match = re.search(r'\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}', field)
                if time_match:
                    time_field = time_match.group(0).replace('T', ' ')
            elif 'LPG_corrected' in field:
                LPG_parts = field.split(':')
                if len(LPG_parts) == 2:
                    LPG_str = LPG_parts[1].strip().strip('}')
                    try:
                        LPG_field = float(LPG_str)
                    except ValueError:
                        print("Failed to convert LPG:", LPG_parts[1])
                        continue
        if time_field and LPG_field is not None:
            datetime_obj = datetime.strptime(time_field, '%Y-%m-%d %H:%M:%S')
            if start_time <= datetime_obj <= end_time:
                time_data.append(datetime_obj)
                LPG_data.append(LPG_field)
                print(datetime_obj, LPG_field)

average_LPG = sum(LPG_data) / len(LPG_data) if h2s_data else 0
print(f"Average LPG Concentration: {average_LPG} ppm")

if len(time_data) != len(LPG_data):
    print("wrong data")
else:
    plt.figure(figsize=(20, 5))
    plt.plot(time_data, LPG_data, marker='o')
    plt.axhline(y=average_LPG, color='r', linestyle='--', label=f'Average_LPG =
{average_LPG:.2f} ppm')
    plt.title('LPG Over Time')
    plt.xlabel('Time(utc-0)')
    plt.ylabel('LPG(ppm)')
    plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d %H:%M:%S'))
```

```
plt.gca().xaxis.set_major_locator(mdates.HourLocator(interval=1))
plt.gcf().autofmt_xdate()
plt.show()
```

Calibration Procedure

1. Setup and Initial Stabilization:
    a. Connect the sensors to the power supply and ensure all connections are secure.
    b. Place the sensors in a clean air environment for initial stabilization.
    c. Power on the sensors and allow them to stabilize for at least 48 hours for the BME680/BME68 sensors and 72 hours for the MQ-4 and SRAQ-G018 sensors.
2. Temperature, Humidity, and Pressure Calibration:
    a. Ensure the BME680/BME688 sensors are placed in an environment with known temperature, humidity, and barometric pressure.
    b. Operate the sensors in this environment for 48 hours to stabilize sensor output.
    c. Verify data consistency against known environmental standards.
3. Particulate Matter Calibration:
    a. Place the Adafruit PMSA003I sensor in an environment with known particle sizes and concentrations.
    b. Adjust the sensor's response to match the controlled concentrations by following the manufacturer's calibration instructions.
    c. Ensure the sensor readings align with the known particle concentrations.
4. Methane and Hydrogen Sulfide Calibration:
    a. Expose the MQ-4 Methane and SRAQ-G018 Hydrogen Sulfide sensors to clean air for 72 hours to achieve baseline stability.
    b. Use known concentrations of methane and hydrogen sulfide to create calibration curves.
    c. Record the voltage readings from the sensors and correlate these with the known gas concentrations to create a calibration curve.
    d. Use the calibration curve to determine a conversion factor for translating voltage differences into gas concentrations in parts per million (ppm).
5. Implementing the Calibration Factor:
    a. Update the .py payload formatter script with the calibration factor. The script should include a section where the raw voltage readings are converted to gas concentrations using the calibration curve.
6. Testing and Validation:
    a. Conduct initial testing of the calibrated sensors inside a controlled building to verify accuracy.
    b. Move the sensors outside the building and compare the readings to validate performance under different environmental conditions.
    c. Adjust the calibration factors if necessary to ensure data accuracy.
7. Ongoing Calibration:
    a. Periodically recalibrate the sensors to maintain accuracy over time.
    b. Update the conversion factor in the .py uplink code as needed based on new calibration data.

For integrating wind speed and wind direction sensors refer to this website:
https://www.elationsportstechnologies.com/post/emt-conduit-mounted-weather-station-wind-sensors

Data analysis code and sample data:
https://drive.google.com/drive/folders/17ZQuyl0P3eI9wqT2eIMCgCIeB1iT6W1d?usp=drive_link

Code:

---

ttn_otaa_highlevel.ino:

```
#ifdef COMPILE_REGRESSION_TEST
#define FILLMEIN 0
#else
#define FILLMEIN (#Don't edit this stuff. Fill in the appropriate FILLMEIN values.)
#warning "You must fill in your keys with the right values from the TTN control
panel"
#endif


#include <Arduino_LoRaWAN_ttn.h>
#include <lmic.h>
#include <hal/hal.h>
#include "keys.h"
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME680.h>
#include <Adafruit_PM25AQI.h>
#include <MQUnifiedsensor.h>



#define Board                     "MCCI Catena 4410"
#define Pin                       (A0)
#define Type                      "MQ-4"
#define Voltage_Resolution        5
#define ADC_Bit_Resolution        12
#define RatioMQ4CleanAir
MQUnifiedsensor MQ4(Board, Voltage_Resolution, ADC_Bit_Resolution, Pin, Type);

#define MQ136_PIN  A1
#define RatioMQ136CleanAir
MQUnifiedsensor MQ136("Adafruit Feather M0", 5, ADC_RESOLUTION, MQ136_PIN,
"MQ-136");


Adafruit_BME680 bme680;
Adafruit_BME680 bme688;
Adafruit_PM25AQI pm25;


uint64_t lastTime = 0;
uint32_t bufferLength = 8;
```

```
static uint8_t messageBuffer[8] = {0, 1, 2, 3, 4, 5, 6, 7};

struct __attribute__((__packed__)) pkt_fmt{
 float temperature;
 float humidity;
 float pressure;
 float gas_resistance;
 float LPG;
 float pm25;
 float h2sConcentration;
};

pkt_fmt myPkt;

#ifdef __cplusplus
extern "C"{
#endif

void myStatusCallback(void * data, bool success){
 }

#ifdef __cplusplus
}
#endif

class cMyLoRaWAN : public Arduino_LoRaWAN_ttn {
//Add the following to your cMyLoRaWan definition:
using Super = Arduino_LoRaWAN_ttn;
public:
 bool begin(const Arduino_LoRaWAN::lmic_pinmap& map);
 cMyLoRaWAN() {};

protected:
   // you'll need to provide implementations for each of the following.
   virtual bool GetOtaaProvisioningInfo(Arduino_LoRaWAN::OtaaProvisioningInfo*)
override;
   virtual void NetSaveSessionInfo(const SessionInfo &Info, const uint8_t
*pExtraInfo, size_t nExtraInfo) override;
   virtual void NetSaveSessionState(const SessionState &State) override;
   virtual bool NetGetSessionState(SessionState &State) override;
   virtual bool GetAbpProvisioningInfo(Arduino_LoRaWAN::AbpProvisioningInfo*)
override;

};

cMyLoRaWAN myLoRaWAN {};
```

```
const cMyLoRaWAN::lmic_pinmap myPinMap = {
    .nss = 8,
    .rxtx = cMyLoRaWAN::lmic_pinmap::LMIC_UNUSED_PIN,
    .rst = 4,
    .dio = { 3, 6, cMyLoRaWAN::lmic_pinmap::LMIC_UNUSED_PIN },
    .rxtx_rx_active = 0,
    .rssi_cal = 0,
    .spi_freq = 8000000,
};

void setup() {

   {
     uint64_t lt = millis();
   }
   myLoRaWAN.begin(myPinMap);
   lastTime = millis();


   float calcR0 = 0;

   if (!bme680.begin(0x76)) {
   }

   if (!bme688.begin(0x77)) {
   }
   bme688.setGasHeater(320, 150);

   if (!pm25.begin_I2C()) {
   }


   MQ4.init();

   for (int i = 1; i <= 10; i++) {
     MQ4.update();
     calcR0 += MQ4.calibrate(RatioMQ4CleanAir);
   }
   MQ4.setR0(calcR0 / 10);

   MQ136.init();

   for (int i = 1; i <= 10; i++) {
     MQ136.update();
     calcR0 += MQ136.calibrate(RatioMQ136CleanAir);
   }
```

```
   MQ136.setR0(calcR0 / 10);
   if (isinf(calcR0)) {
     while (1);
   }
   if (calcR0 == 0) {
     while (1);
   }
   MQ136.serialDebug(true);

   myLoRaWAN.SendBuffer((uint8_t *) &myPkt, sizeof(myPkt), myStatusCallback, NULL,
false, 1);
}

void loop() {

   myLoRaWAN.loop();
   if (millis() - lastTime > 90000) {
       messageBuffer[0]++;

       if (!bme680.performReading()) {
       } else {
           float temperature = bme680.temperature;
           memcpy(&myPkt.temperature, &temperature, 4);
           float humidity = bme680.humidity;
           memcpy(&myPkt.humidity, &humidity, 4);
           float pressure = bme680.pressure / 100.0;
           memcpy(&myPkt.pressure, &pressure, 4);
       }

       if (bme688.performReading()) {
           if (bme688.gas_resistance > 0) {
               float gas_resistance = bme688.gas_resistance;
               memcpy(&myPkt.gas_resistance, &gas_resistance, 4);
           } else {
           }
       }

       PM25_AQI_Data data;
       if (pm25.read(&data)) {
           float pm25 = data.pm25_env;
           memcpy(&myPkt.pm25, &pm25, 4);
       } else {
       }

       MQ4.update();
       MQ4.setA(3811.9); MQ4.setB(-3.113);
       float LPG = MQ4.readSensor();
```

```
        memcpy(&myPkt.LPG, &LPG, 4);

        MQ136.update();
        MQ136.setA(36.737);
        MQ136.setB(-3.536);
        float h2sConcentration = MQ136.readSensor();
        memcpy(&myPkt.h2sConcentration, &h2sConcentration, 4);

        delay(30000);

        myLoRaWAN.SendBuffer((uint8_t *)&myPkt, sizeof(myPkt), myStatusCallback,
NULL, false, 1);
        lastTime = millis();
   }
}


// this method is called when the LMIC needs OTAA info.
// return false to indicate "no provisioning", otherwise
// fill in the data and return true.
bool
cMyLoRaWAN::GetOtaaProvisioningInfo(
   OtaaProvisioningInfo *pInfo
   ) {
     if (pInfo){
        memcpy_P(pInfo->AppEUI, APPEUI, 8);
        memcpy_P(pInfo->DevEUI, DEVEUI, 8);
        memcpy_P(pInfo->AppKey, APPKEY, 16);
     }
   return true;
}

void
cMyLoRaWAN::NetSaveSessionInfo(
   const SessionInfo &Info,
   const uint8_t *pExtraInfo,
   size_t nExtraInfo
   ) {
   // save Info somewhere.

   //Do the rest here!
}

void
cMyLoRaWAN::NetSaveSessionState(const SessionState &State) {
   // save State somwwhere. Note that it's often the same;
   // often only the frame counters change.
```

```
}

bool
cMyLoRaWAN::NetGetSessionState(SessionState &State) {
   // either fetch SessionState from somewhere and return true or...
}

bool
cMyLoRaWAN::GetAbpProvisioningInfo(Arduino_LoRaWAN::AbpProvisioningInfo* Info){
 //either get ABP provisioning info from somewhere and return true or...
 SessionState temporaryState;

 SessionState State;
 if (!NetGetSessionState(State)) return false;
 Info->FCntUp = State.V1.FCntUp;
 Info->FCntDown = State.V1.FCntDown;
 return false;
}

bool cMyLoRaWAN::begin(const Arduino_LoRaWAN::lmic_pinmap& map) {
   return Super::begin(map);
}
```

---

keys.h:

```
#include "stdint.h"
#ifndef KEYS_H
#define KEYS_H
const uint8_t PROGMEM APPEUI[8] = {};
const uint8_t PROGMEM DEVEUI[8]= {};
const uint8_t PROGMEM APPKEY[16]= {};
#endif
```

EJdata2text.py:

```
import paho.mqtt.client as mqttClient
import time
import os
import datetime
import pytz


def on_connect(client, userdata, flags, rc):
```

```
    if rc == 0:

        print("Connected to broker")

        global Connected                #Use global variable
        Connected = True                #Signal connection

    else:

        print("Connection failed")

def on_message(client, userdata, message):
    print("")
    print("Message received: "  + str(message.payload))

    file_name = ''

    utc_time = datetime.datetime.utcnow()
    utc_time = utc_time.replace(tzinfo=pytz.utc)
    eastern = pytz.timezone('US/Eastern')
    local_time = utc_time.astimezone(eastern)

    formatted_time = local_time.strftime('%Y-%m-%d %H:%M:%S')

    with open(file_name,'a+') as f:
        message_str = str(message.payload.decode('utf-8'))
        f.write(f"{formatted_time} - {message_str}\n")

Connected = False   #global variable for the state of the connection

file_name = ''

file_path = os.path.abspath(file_name)
print(file_path)

broker_address= ""  #host
port = 1883                         #Broker port
user = "" #<--  Put your TTN V3 app here                    #Connection username
password = #<--  Put your TTN V3 API key in quotes           #Connection password

client = mqttClient.Client("Python")               #create new instance
client.username_pw_set(user, password=password)    #set username and password
client.on_connect= on_connect                      #attach function to callback
client.on_message= on_message                      #attach function to callback
client.connect(broker_address,port,60) #connect
client.subscribe(f"v3/{user}/devices/+/up") #subscribe
client.loop_forever() #then keep listening forever
```

WORKS CITED

"Environmental Justice and Urban Agriculture." Welcome!, sites.google.com/blueprintgeneva.org/welcome/project-areas/environmental-justice-and-urban-agriculture?authuser=0

EPA, Environmental Protection Agency, www.epa.gov/ej-research/epa-research-environmental-justice-and-air-pollution#:~:text=Clean%20air%20is%20important%20for,color%20and%20low%2Dincome%20communities

EPA Researchers Develop New Air Monitoring Technology to Understand Leaks and Irregular Emissions from Sources | US EPA. (2023, October 4). US EPA. https://www.epa.gov/sciencematters/epa-researchers-develop-new-air-monitoring-technology-understand-leaks-and-irregular

"Geneva, New York (NY) Poverty Rate Data Information about Poor and Low-Income Residents." Geneva, New York (NY) Poverty Rate Data - Information about Poor and Low-Income Residents Living in This City, www.city-data.com/poverty/poverty-Geneva-New-York.html

"Health Impacts." World Health Organization, World Health Organization, www.who.int/teams/environment-climate-change-and-health/air-quality-energy-and-health/health-impacts#:~:text=Air%20pollution%20is%20a%20risk,(household%20air%20pollution%20only)

Jabbar, W. A., Subramaniam, T., Ong, A. E., Shu'Ib, M. I., Wu, W., & De Oliveira, M. A. (2022). LORAWAN-Based IoT System Implementation for Long-Range Outdoor Air Quality Monitoring. *Internet of Things*, *19*, 100540. https://doi.org/10.1016/j.iot.2022.100540

"May 6: Environmental Justice & Air Quality." May 6: Environmental Justice & Air Quality | AirNow.Gov, AirNow.gov, U.S. EPA, www.airnow.gov/aqaw-2021/environmental-justice/#:~:text=Environmental%20injustice%20predominately%20affects%20minority,air%20quality%20awareness%20and%20work

Mckinley, Jesse, and Lauren Petracca. "Why a Landfill as Tall as the Statue of Liberty May Rise Even Higher." The New York Times, The New York Times, 17 Sept. 2023, www.nytimes.com/2023/09/17/nyregion/new-york-landfill-seneca-meadows.html

"Methane and the Environment: Socalgas." SoCalGas, A Sempra Energy Utility, www.socalgas.com/stay-safe/methane-emissions/methane-and-the-environment#:~:text=Because%20it%20is%20able%20to,than%20are%20those%20other%20gases

Methane General Information,

assets.publishing.service.gov.uk/media/5c34c0b240f0b6445ac3e198/Methane_PHE_general_information__070119.pdf

U.S. Census Bureau Quickfacts: Waterloo Town, Seneca County, New York, www.census.gov/quickfacts/fact/table/waterlootownsenecacountynewyork/PST045223

Supreme Court of the State of New York county of Albany. Seneca Lake Guardian, INC., Seneca Falls Environmental Action Committee, Waterloo Contractors, INC., d/b/a Waterloo Container Company, Absolute Auto Repair, INC., Valerie Sandlas, and Heather Bonetti V Seneca Meadows, INC. and The New York State Department of Environmental Conservation. . 25 Mar. 2024.

Warde, Bryan. “Why Race Still Matters 50 Years After the Enactment of the 1964 Civil Rights Act.” *Journal of African American Studies*, vol. 18, no. 2, 2013, pp. 251–259., doi:10.1007/s12111-013-9264-3

“Working-Age Americans Are Struggling to Pay for Health Care, Even Those with Insurance, Report Finds.” CBS News, CBS Interactive, www.cbsnews.com/news/health-care-costs-unaffordable-even-for-insured-americans-commonwealth-fund/

Public Health England. “Methane General Information.” 2019. Public Health England, https://assets.publishing.service.gov.uk/media/5c34c0b240f0b6445ac3e198/Methane_PHE_general_information__070119.pdf

NASA Earth Observatory. "Atmospheric Methane." 2005. NASA Earth Observatory, https://earthobservatory.nasa.gov/images/5270/atmospheric-methane.

Occupational Safety and Health Administration (OSHA). "Hydrogen Sulfide: Hazards." OSHA, https://www.osha.gov/hydrogen-sulfide/hazards.