Dyson School of Design Engineering | MEng Design Engineering

Sensing & IoT
DE4-SIOT

Module Exam: Dyson Smell Station

Student CID: 01559850

December 17, 2021

Presentation URL: https://bit.ly/3qo7FKX

Dashboard URL: https://colinlaganier.com/siot

Code & Data: https://github.com/colinlaganier/DE4-SIOT

1 Coursework 1: Sensing

1.1 Introduction and objectives

As the number of students returning to the Imperial College London campus increased and inched its way ever closer to the pre-COVID level, some disturbing changes started to appear in our student environment. As a final year student, I have been away from the Dyson Building for over 6 months as I was conducting my placement in a company. The most glaring change came in the form of the strong smell associated with the Level 2 shared space. A communal space loved by students from every year group, where many projects end their life, where countless reports have been written, and above all where innumerable meals have been eaten.

This seemingly sudden change raised many questions in my mind, and while many hypothesis were formulated and many potential culprits were identified, no in-depth research was undertaken. To attempt to put an end to the speculation and identify once and for all what was at the root of this troubling issue.

Among the potential root causes of this issue, three hypotheses were put forward as most likely as well as more easily quantifiable using data acquired from the space. These were:

1. The bins in the space are not emptied often enough by the cleaning team, therefore leading to fermentation and the release of unpleasant smells over the course of the organic material's decomposition.

2. The heating up and the consumption of food at lunchtime releases gases which give the room a bad smell.

3. The students themselves are coming to the Dyson Building smelling bad, or are releasing foul smelling gases, therefore giving the space itself an unpleasant smell.

Coming into the study with prior experience monitoring air quality using more complex automated air monitoring solutions (Airbeam2 [1]), I was aware of the issues related with monitoring large spaces. For this reason, as the number of sensors which could be used was restricted by the budget of the project, the system was designed to be flexible with the possibility of expanding upon it throughout the building at a later time to obtain better results and a better understanding of the global air quality.

The results identified

1.2 Data Sources and sensing set-up

To streamline the study and to verify the validity of the three hypothesis that were formulated, the data acquisition was divided and studied as three sub-sections: the air-quality monitoring station, the bin-volume monitoring and occupancy.

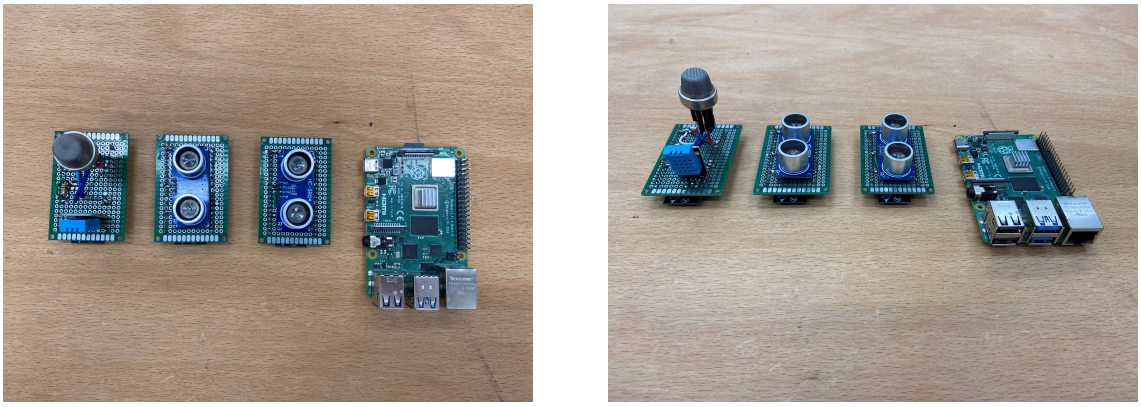

Figure 1: Sensors used for Level Two monitoring, from left to right: air quality sensor, bin 1 volume sensor, bin 2 volume sensor, occupancy scraping computer

1.2.1 Air Quality Monitoring

Air quality monitoring could have been implemented in multiple ways using a variety of different sensors to obtain a very clear idea of which gases were present and in what concentration in the room. The first

method explored was to integrate an Airbeam2, a con summer grade particulate matter sensor. This sensor is of relatively good quality and would have given quite accurate results in the form of particle count. However, due to the requirement of a constant Bluetooth with a phone and no easy access to the data after the monitoring, this option was not practical for this automated implementation. Instead, single target gas sensors were explored due to their easy access as well as their relatively low cost compared to complex automated systems. Among the different sensors, the MQ-136 Hydrogene Sulfide Sensor was selected. This sensor was chosen as Hydrogen Sulfide, also known as the "sewer gas" [2], is the smell of rotten eggs. As the main goal of this research was to identify cause of the smell of Level 2, this sensor seemed very well suited. Unfortunately, this sensor is far more expensive than typical low-cost ones at £40, and for this reason could not be coupled with any other gas sensor. This was indeed the original goal, by coupling it with Ammonia and Methane sensors to have a broader scope. Instead, the DHT11 temperature and humidity sensor was repurposed from a past project. This sensor would also be useful to learn more about the current environment on Level 2.

Figure 2: Air monitoring sensors taped to an elevated pole on Level 2

Both sensors were soldered to a strip board with the necessary resistors to reduce the voltage of the 5 V output signal for our 3.3 V microcontroller. The ESP32 was used due to its low cost as well as on-board 2.4 GHz WiFi which will be needed to transmit the data as the sensors will be spread apart on Level 2. This installation only required power, supplied from a repurposed phone charger plugged in the wall. To fully match the olfactory sensation obtained when walking in the room, the sensor was taped on a pole as to elevate it and match the level of the nose.

1.2.2 Bin-volume Monitoring

The bin-volume monitoring was done using an ultrasound sensor, emitting a frequency and calculating how filled the bin is by measuring the delay between the impulse and the received signal. This value could then be transformed into a distance by knowing the speed of sound or in the form of a percentage representing how filled up the bin is based on preliminary readings when empty and full. Alternative methods were considered such as using a scale under each bin and monitoring these changes over the course of the day. However, the implementation using an ultrasonic sensor was more cost effective and easier to install. The ultrasound sensor was soldered to a strip board and wired to the ESP32 to transmit the data. As this sensor was to be taped on the inside of the bin lid, the sensor package needed to be flush and as thin as possible.

1.2.3 Occupancy Monitoring

The final data source of the system is the occupancy monitoring system. This element was crucial to see how the number of people varied over the course of the day. It was originally planned to use the card scanners on both of the entrances of the common space to count the number of people coming in and out. However, such a system would have been very complicated to implement and would have required other sensors to be constantly running during the day. Two implementations were considered, first an infrared sensor on top of the door frame, triggered by the opening and closing of the door, and second a computer vision algorithm which would have monitored the change in colour of the card reader. While both of these implementations could have been developed, a third option was identified. By using the Lone Rooftop space utilisation platform, developed by Hubstar and which Imperial College has implemented, a live

Figure 3: Ultrasonic sensors taped to the inside of the lid of the bins to monitor its content

approximation of the occupation of the space was available. This software uses IP addresses connected to the access points, eliminating the ones too close to one-another to determine how many students are in the space. While the results from the platform could be more accurate, this created a reliable option to have an approximation of the occupancy of the space. Unfortunately, while I was granted access from the department to the online platform, I could not get the access to the underlying API to periodically fetch the data. As an alternative, a scraping algorithm was developed using Python and the Selenium library to obtain the data by regularly signing in to the platform and returning the value from the loaded web page. As real time data was required and as no past data was saved on the platform, real time scraping was the only option. Additionally, as the scraping script was developed, it was noticed that the website had a security measure put in place, disabling the rendering of the page in headless mode, therefore requiring a display and GUI to load the web page. As the system was designed to rely on a central Raspberry Pi running headless Debian which was already setup and ready to start monitoring, this required another Raspberry Pi to be dedicated exclusively to the scraping.

1.2.4 Setup

As the study revolved around monitoring at Imperial College, several challenges arose during the setup of the system on campus. The sign-in process to the WPA-EAP was tedious on Linux but was resolved after several hours of trying different login protocols. This process was repeated with the ESP32's Arduino code with more ease. All in all 3 ESP32 were used as well as two Raspberry Pis.

1.3 Data collection and storage process

1.3.1 MQTT Network Protocol

As flexibility and the prospect of expanding this system at a later date was a self-imposed goal, the system was designed to be able to integrate new sensors with little to no modifications to the software or hardware. For this reason, an MQTT system was created, creating an MQTT broker running on a central Raspberry Pi, with many MQTT clients running on lower powered ESP32 devices. Each ESP32 is connected to internet and has the ability to verify the time periodically by synchronising itself with a Network Time Protocol (NTP) server, and basing the data monitoring from that. When the data is acquired, the ESP32s simply publish the sensor data to their pre-assigned topic, which is the received by the MQTT Broken. The system diagram in Fig. 4 explains the communications between each component of the central broker.

As mentioned previously, the goals of this implementation is to become a lot bigger with many more clients. While the MQTT network works very well with the 4 clients, the real value of MQTT can only be observed in far larger implementations with many more clients. Never the less, the MQTT framework brings many advantage for such a system, especially in the form of reduced strain on the WiFi network thanks to the lightweight and efficient nature of the communication protocol.

1.3.2 Data Collection

Prior to the deployment of the system, an informal study was conducted to see how long individuals would stay on Level 2 and how they would interact with the space. It was found that it was a relatively slow

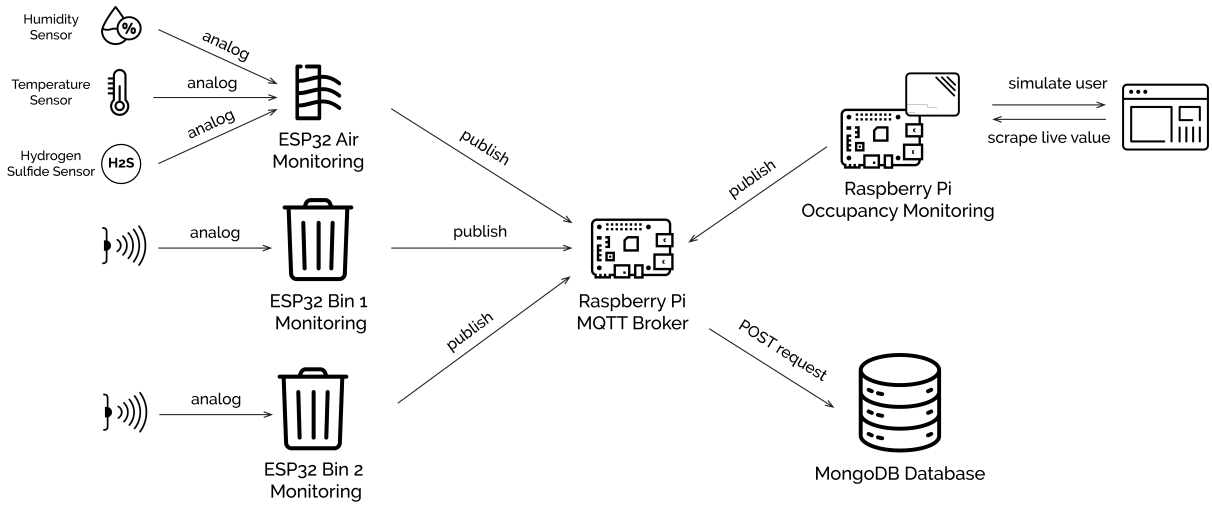

Figure 4: MQTT System Diagram of the Dyson Smell Station

changing environment, with most students staying for extended periods of time in the space. Additionally, while it was possible that faster moving actors coming in and out of the common space would affect the air quality and more specifically the smell of the room, it was decided to ignore these. Finally, all of the other studied parameters (temperature, humidity, H2S concentration and bin volume) are very slow changing and their variations can be seen as semi permanent: temperature takes a considerable time to rise back after a sudden drop, volume of trash can only increase, etc. For these reasons, a sampling rate of once every 10 minutes was determined to be satisfactory. This created suitable granular data while not creating an excessive amount of data to sift.

To ensure that the readings from the sensors were accurate and to reduce the impact of sampling anomalies, each sensor reading was done 5 times separated out by a few seconds and then averaged. Additionally, if the value of too many of the samples seemed out of the expected bracket, the data would be thrown away and the measurement redone. This mitigated for example for the unlikely possibility of the sensor reading taking place when a user is throwing something in the bin.

1.3.3 Storage

When all of the sensor readings have arrived to the MQTT broker, or when the 3 minute timeout period has elapsed since the first received measurement, the MQTT broker packages the data as a JSON and sends it to the database. The database used for this project is MongoDB, a NoSQL database with free tiers. The data is sent to the database using a POST request using the MongoClient Python library. The schema of the JSON object to be added to the *sensorData* collection is:

```
sensorData = {
        _id = objectId,
        timestamp = UNIX timestamp (int),
        bin1 = double,
        bin2 = double,
        H2S = integer,
        temperature = double,
        humidity = integer,
        occupancy = integer,
    }
```

1.3.4 Issues

As there were quite a large number of individual components running at the same time, issues were expected to arise as soon as the system was deployed. Several ones manifested themselves right from the start, while others were lingering issues during the duration of the sensing. The main issue with the implementation was the inability to obtain a static IP address. Because of the periodic and unexpected shuffling of the IP address by the access points, the MQTT network was down repeatedly as the MQTT

clients required the IP address of the broker to publish sensor data. This required a manual update of the Arduino code with the new IP Address of the broker. To alert me of when this occured, an automated email was sent to me when the IP verification failed. In larger implementations, this should be easily solved by communicating with the Imperial IT service. As two of the sensors were placed in the bins, a sign was placed next to it to ensure no-one was to unplug the micro controller over the period of the sensing. Unfortunately, this was misunderstood by the cleaning staff, which did not empty the bin for two days following the installation of the sensors. This is when the highest levels of monitoring of the bins were registered. Finally, a power issue from one of the AC to DC converters was found. Because of the unstable power produced by the plug, as well as the relatively high current draw from the MQ-136 to keep the sensor at a constant temperature, this caused the ESP32 to periodically shut down. When this issue was identified, the power was then provided from a laptop over the course of the day as it was to late to purchase a new plug. This caused several outages and took down the monitoring of the H2S, temperature and humidity.

1.4 Basic characteristics of the end-to-end systems set up and data

The system was designed to run and collect data without human supervision. Never the less, the MongoDB live dashboard was a useful tool to see how the system was doing and to track the progress of the data collection.

Initial data collection and study seemed promising, however the gaps in the data when the system failed were quite worrying, see Fig. 5.

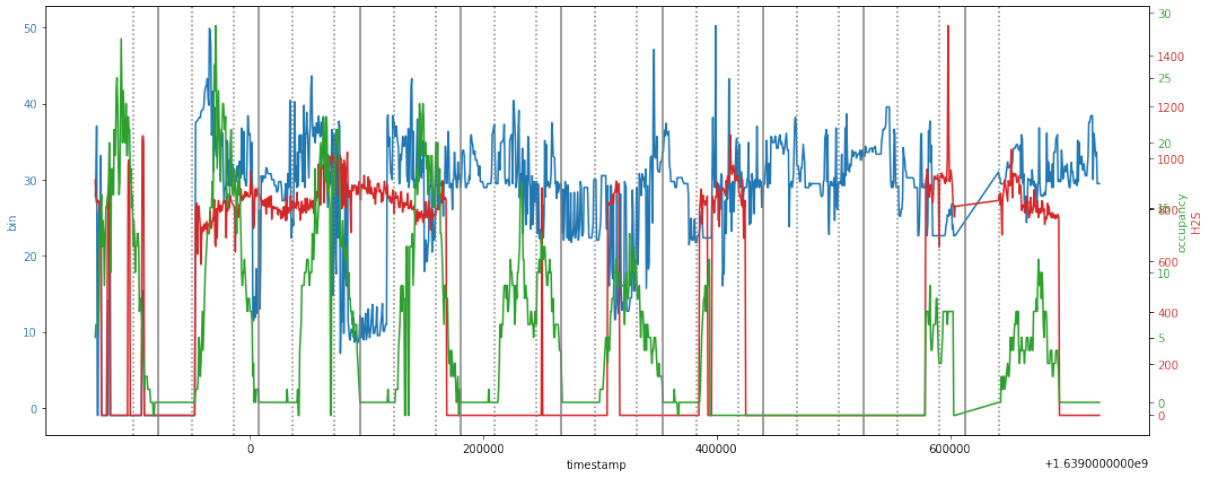

Figure 5: Initial plot of the values before the data analysis

2 Coursework 2: Internet of Things

2.1 Data interaction & visualisation platform

Once over a week's worth of data was collected, the current sensor readings were then fetched from the database to be processed and cleaned up for data visualisation. The data cleaning was done in a Jupyter notebook to easily visualise the data at every step of the way. The sensing implementation could not be left running over the Christmas break as I needed the Raspberry Pi. Therefore, the data could be processed at once and exported as a csv for the visualisation as there was no computational advantage to process the raw data in real time on the dashboard.

The dashboard was made using Google Data Studio, a powerful solution which can create dashboards and report combining multiple sources of data. This was in turn integrated in my personal website, hosted on Firebase, for ease of access. The dashboard enables the user to interact with the data when hovering on top of the data points as well as choosing how many data streams are displayed on each plot. This dashboard is very functional to display the cleaned up sensor data and could allow for real time fetching of the data.

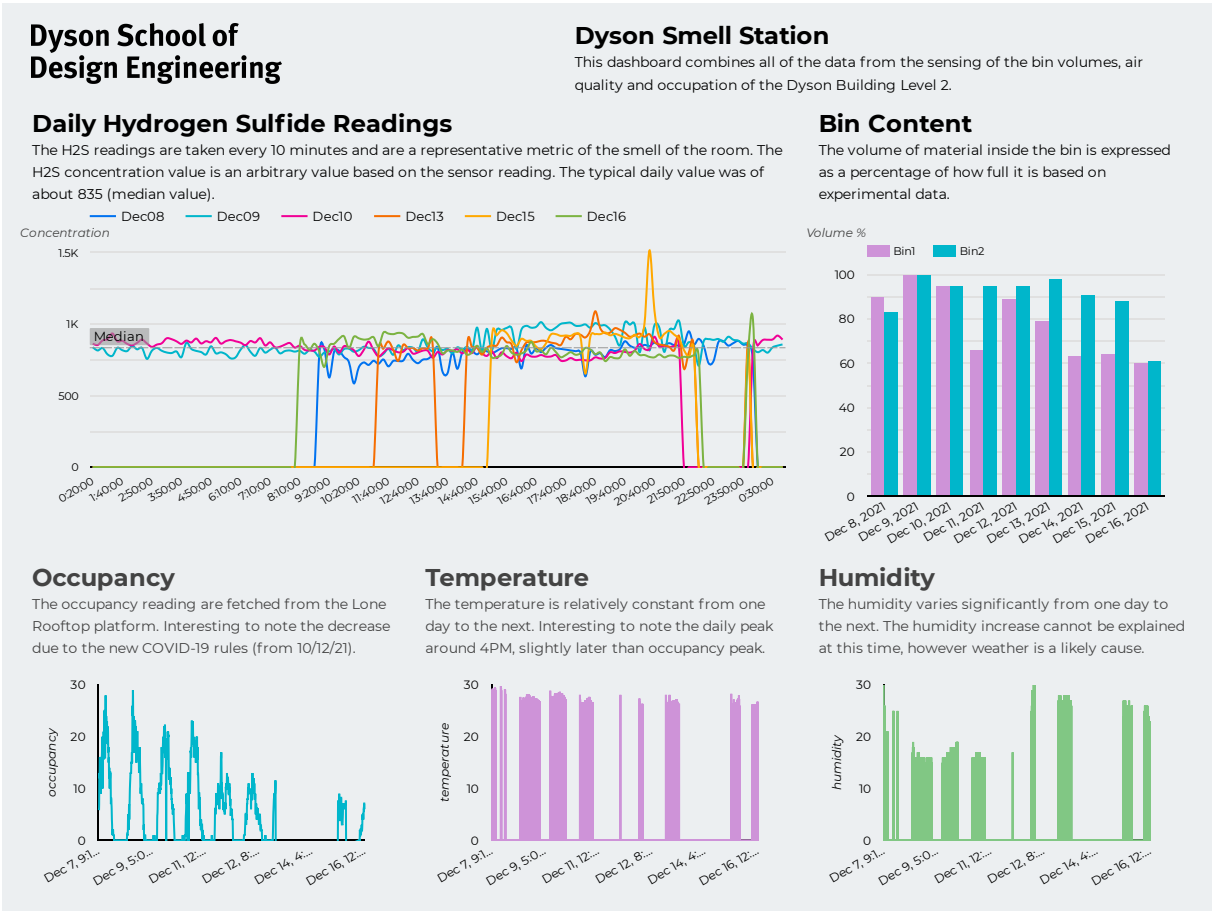

Figure 6: Data visualisation dashboard showing the key sensor data from the Dyson Smell Station

2.2 Data analytics, inferences and insights

2.2.1 Hypothesis 1

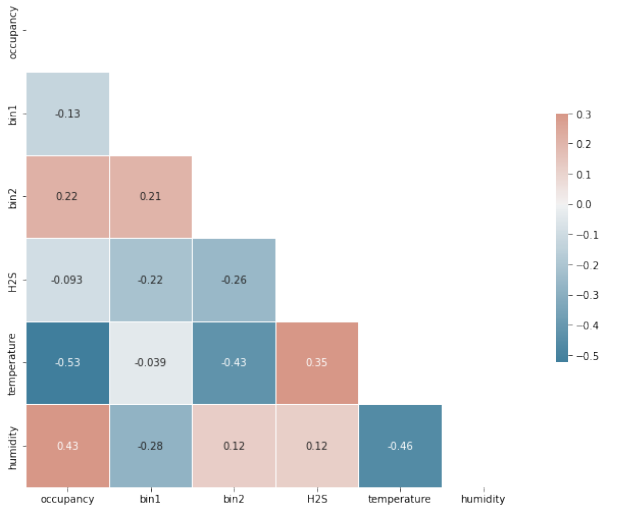

Figure 7: Correlation matrix of a week's worth of data

To verify if the bins were responsible for the smell of the room, the data was analysed to identify any potential correlations. By plotting the correlation matrix of using a week's worth of data (removing days with no data or insufficient data), the relationship between each variable can be observed. As Fig. 7

shows, there is only very little correlation between the data, with a correlation of -0.22 and -0.26 between the measured Hydrogen Sulfide levels and the bin volume from Bin 1 and Bin 2 respectively. The same poor correlation between the sensor data is related by the plotting of the normalised values against each other as seen in Fig. 8.

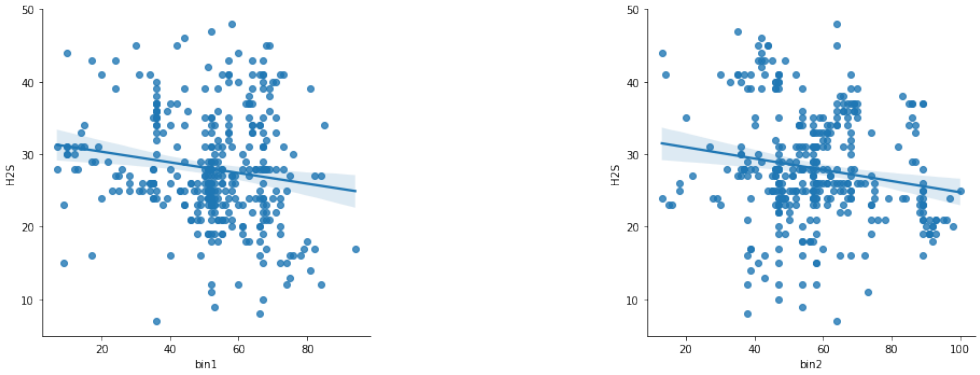

Figure 8: Point cloud with regression model fit line of H2S levels and the bin volumes

Based on the data acquired from the week of sensing, the correlation between the bin volume and the H2S concentration in the room is not very strong. However, it is interesting to note that more than half (4/7) of the peaks in the readings from the MQ-136 sensor (see Fig. 9, highlighting the highest smell levels of the day, are in the evening. This could potentially be a result of the start of the fermentation process releasing gases from the food wastes from lunch. This could be studied in further depth by placing gas sensors directly in the bins at a later time.

2.2.2 Hypothesis 2

To verify if the food the students are bringing to level 2 to consume is responsible for the space's smell, the maximum Hydrogen Sulfide reading were studied. If the food brought in by the students was at the root of the smell problem, a peak would be expected around lunch time, as only very few students have dinner in the common space. By taking a look at the maximum values seen every day on Fig. 9 and Table. 1, it can clearly be seen seen that these peaks do not line up with the time of day where students eat food.

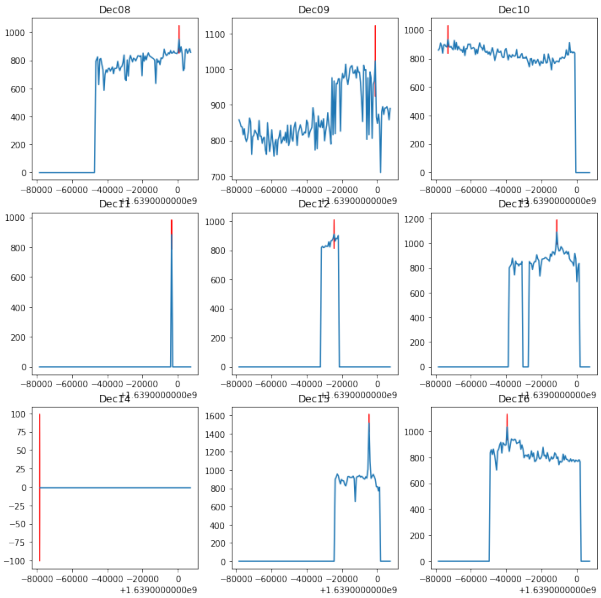

Table 1: Daily H2S max value and timestamp

Date	Time	Value
December 8	22:00	950
December 9	21:30	1023
December 10	01:30	933
December 11	N/A	N/A
December 12	15:00	910
December 13	18:40	1090
December 14	N/A	N/A
December 15	20:30	1515
December 16	10:50	1033

Figure 9: Daily H2S reading with peak detection

Unfortunately, due to the lack of full day monitoring with no cutoffs, it is hard to predict if these identified peaks are truly representative of the highest values seen during the day. This experiment could

be repeated when the system is tweaked to be more stable and to provide more reliable data throughout the day.

2.2.3 Hypothesis 3

As it can be seen in the correlation matrix of Fig. 7, the correlation between the H2S concentrations and the occupancy of the room is very weak with a value of just under 0.1. This lack of data highlighting a correlation between the two variables seem to indicate that the number of users does not affect the smell of the common space.

2.3 Discussions on the important aspects of the project

While the implementation of the system was a relative success as every part of the system produced data and was collated using the MQTT protocol, the sensor data obtained was not optimal for the analysis and to conclusively prove one of our hypotheses. Among the major issues, two seem like the most problematic. First, the sensitivity of the MQ-136 sensor is too low for our application, therefore creating very mediocre data from which very few conclusions can be drawn from. Secondly, the reliability of the system was an issue due to hardware failure (power plug) and unstable WiFi. Never the less, the system as a whole can be studied in further depth.

Overall no conclusions can be drawn from the data as to identify a relationship between the smell of the room, and either the bin volume or occupancy. Never the less, the implementation still has a strong potential for reuse as well as to become a larger system. While the upfront cost of the MQTT system seems a little bit more expensive than direct communication from the ESP32 directly to the database, this extra equipment opens the door to many advantages when scaling. Indeed, with more collected data and a structured pipeline, real time data processing could be done directly on the Raspberry Pi, which in turn could be transformed into an actuation of a mechanism regulating the air-flow to the space. This implementation would otherwise be more complicated, requiring a server to process the data in real time and send the information back to the microcontrollers. Based on the prices of most cloud computing services, this in turn will become a more expensive alternative.

2.4 Avenues for future work and potential impact

2.4.1 Security

While the sensor data used in this implementation is not particularly sensitive as it does not contain any discernible features which could be traced back to any user, the security of the system could be improved. Indeed, MQTT uses the Transmission Control Protocol (TCP) transport protocol as the default, which does not use an encrypted communication. This could be upgraded at a later stage to use the Transport Layer Security (TLS) communication protocol.

2.4.2 Power Consumption

A more power conscious implementation could be done, by putting the ESP32 to sleep and having it wake up periodically to broadcast before going back to sleep. However this would required most robust implementation as the MQTT server login periodically failed for no apparent reason as well as the WiFi login process. For this reason, to mitigate the potential points of failure, this sleep state was not implemented in this initial implementation. Additionally, the use of a second Raspberry Pi for the scrapping of the data was not a long term solution. The MQTT Broker Python script could call the live occupancy data REST API and receive the live count from the platform, effectively removing one point of failure and complexity of the system, as well as a very power hungry solution.

2.4.3 Sensor Implementation

The addition of more accurate sensors as well as more of them would make a real difference to the monitoring system. Indeed, as the sensor could only pick up gases from very close range, the results were far from representative of the real air quality in the space. It is interesting to note that the noticeable peak which occurred on December 15th at 8:30PM was from a student eating sushi very close to the sensor. Additionally, the purchase of higher quality ultrasound sensors could very well improve the accuracy of the bin content monitoring and resolve the drift issue noticed after extended use.

References

"AirBeam - Buy It Now." [Online]. Available: https://www.habitatmap.org/airbeam/buy-it-now

"Hydrogen Sulfide :: Washington State Department of Health." [Online]. Available: https://www.doh.wa.gov/CommunityandEnvironment/Contaminants/HydrogenSulfide