ECE 445

Senior Design Laboratory

Fall 2022

Final Report

# Microcontroller-based Occupancy Monitoring (MOM)

Team 7: Franklin Moy, Vish Gopal Sekar, John Li

TA: Hanyin Shao

12/07/2022

# Abstract

Microcontroller-based Occupancy Monitoring (MOM) is a project which shows the current occupancy of a space to its users. The MOM device monitors Wi-Fi traffic density to estimate the number of people in a space in real-time. This estimation is presented on a web application for the public to use. By using the web application, users can know if a location is likely to have available seats without needing to physically visit the location. Using a custom algorithm with finely-tuned parameters, MOM is able to estimate the number of people in a large, enclosed space with at least 80% accuracy and update the web application every five minutes or less.

# Table of Contents

# 1 Introduction

## 1.1 Problem

With the campus returning to normalcy from the pandemic, most, if not all, students have returned to campus for the school year. This means that more and more students will be going to the libraries to study, which also means that the limited space at the libraries will be filled up with the many students who are now back on campus. Even in the semesters during the pandemic, many students have entered libraries such as Grainger to find a place to study, only to leave 5 minutes later because there are no open seats. With the closing of the Undergraduate Library in the Spring 2022 semester, it is likely that this problem will get worse.

Libraries play an essential role in the academic side of the college experience. In a 2015 research survey conducted by Gensler [1], libraries were ranked as the top places for students to study alone and work in a group, with providing quiet space for work ranked as its important resource. However, when there is a lack of study spaces for students, especially during midterm exam periods, this could have a negative impact on students. It could not only reduce the amount of study time available to a student, but it may also reduce motivation to study. As mentioned in an article written by the Daily Bruin, the official student newspaper at the University of California, Los Angeles [2], students have attributed the lack of open study spaces as a cause of unnecessary stress which may affect academic performance.

## 1.2 Solution

Our solution utilizes a fleet of custom devices that will scan for nearby Wi-Fi and Bluetooth network signals in different areas of a building. Since students nowadays will be using phones and/or laptops that emit Wi-Fi signals, scanning for Wi-Fi signals is a good way to estimate the fullness of a building. In an IEEE research article about the effectiveness of using Wi-Fi probe requests to estimate the presence of people [3], the authors found that their proposed method indicated a very strong correlation with the actual number of people present in their environment of study. Our custom devices, which can be deployed in any location near a wall outlet, will be able to scan for these connections. They will then compile occupancy data for their assigned sectors and feed this data into an Internet of Things (IoT) core in the cloud. This IoT core will send information to a cloud database. The database will connect with our web application which students will use to locate open study spaces without aimlessly searching around campus.

It should be mentioned that other occupancy monitoring products exist, such as the Waitz [4] system from the University of California, San Diego. However, those products rely on the use of hardware that are currently experiencing supply chain issues [5], [6]. Our solution will be using microcontrollers and parts which are inexpensive and readily available. Each one of our devices will also have a battery backup, which many existing solutions do not have.

## 1.3 Visual Aid

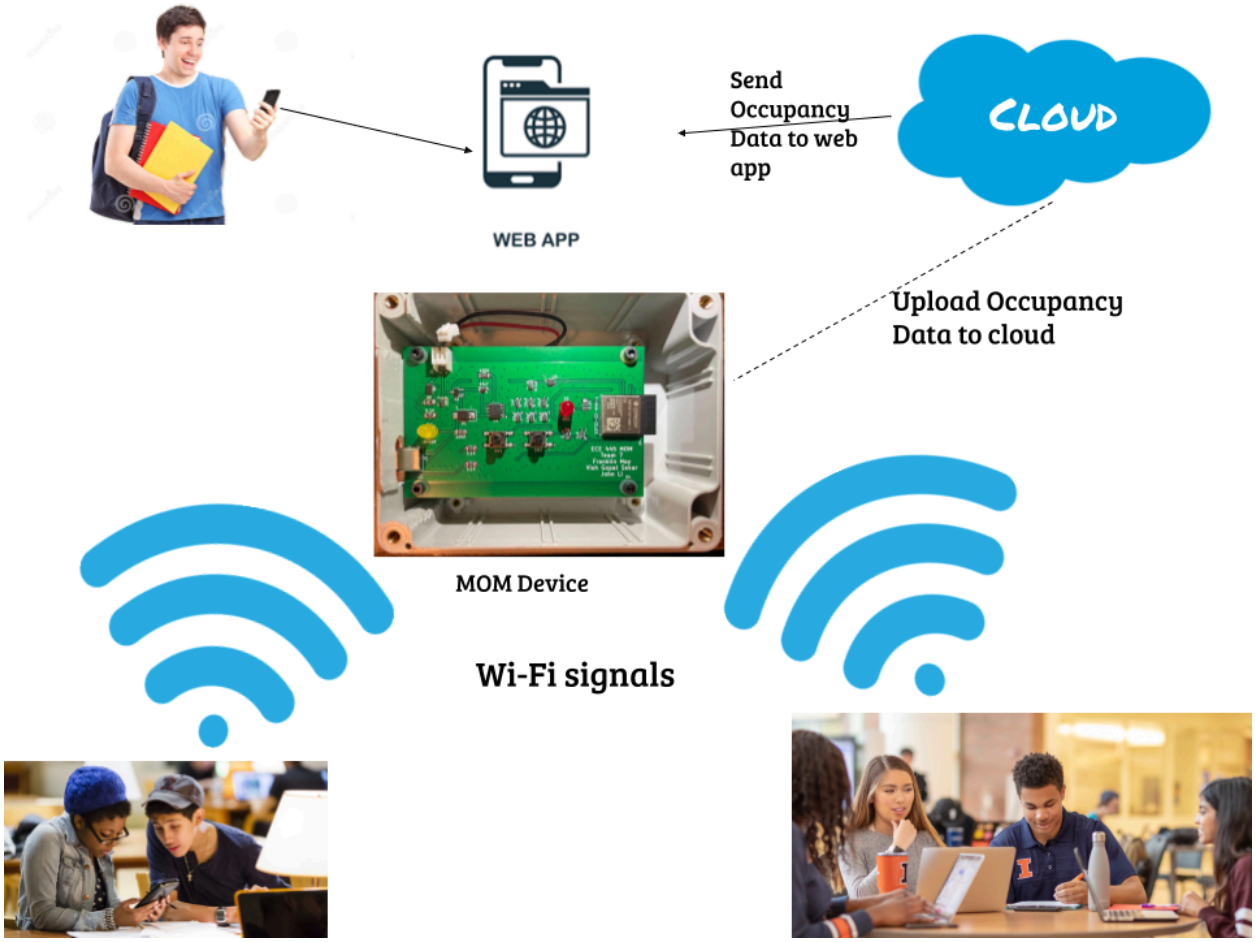



Figure 1: Interactions between MOM Device, Student Devices, and Web Application

## 1.4 High-Level Requirements

- **Estimation Accuracy:** For users to have confidence in the presented data, each MOM device must be able to estimate sector occupancy with at least 80% accuracy.
- **Battery Life:** Each MOM device must be able to run indefinitely while plugged into wall power and for at least one hour when unplugged.
- **Frequent Updates:** The MOM device must consistently send occupancy data such that the web application shows data no more than 5 minutes old.

## 1.5 Subsystem Overview

There are four subsystems that make up the MOM project. Three of these subsystems make up a physical MOM device: the Power Supply, Radio Scanner Suite, and Control Unit. The fourth subsystem, the IoT Cloud, is a purely software-based subsystem that resides in the cloud. Figure 2 below shows the components of each subsystem and how the subsystems interact with each other. An explanation of each subsystem is provided on the following page.

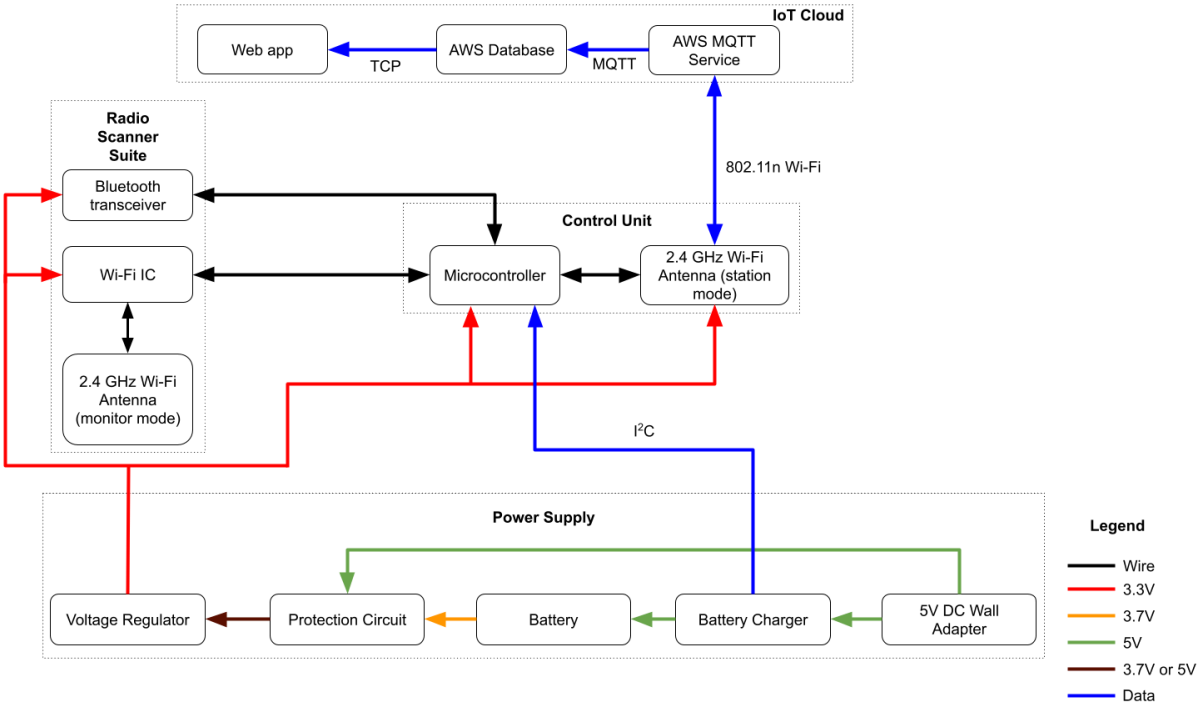



Figure 2: Block Diagram

### 1.5.1 Control Unit

The Control Unit receives information about Wi-Fi traffic density from the Radio Scanner Suite. It will then compile and process this data to estimate the occupancy of the sector that it is monitoring. Once it finishes processing this data, it will connect to the internet and send the data to the IoT Cloud subsystem.

### 1.5.2 Radio Scanner Suite

The Radio Scanner Suite is the subsystem that scans for Wi-Fi traffic. Since most, if not all, modern personal devices contain Wi-Fi radios that communicate in the 2.4 GHz frequency band, we will be monitoring this Wi-Fi frequency to estimate occupancy. The Radio Scanner Suite consists of the necessary components for monitoring these types of RF transmissions. For each packet of Wi-Fi traffic the Radio Scanner Suite picks up, it will send it to the Control Unit for occupancy data processing.

### 1.5.3 Power Supply

The Power Supply delivers power to the Radio Scanner Suite and Control Unit, allowing them to perform their respective functions. The Power Supply consists of a standard 5V USB DC power supply and a lithium-polymer backup battery. If power from the wall outlet is interrupted, the power supply will be able to instantly switch and draw power from the backup battery.

### 1.5.4 IoT Cloud

The IoT Cloud subsystem is a purely software-based subsystem. This subsystem leverages Amazon Web Services (AWS) for web application hosting, occupancy data retrieval, and database management. It receives occupancy data from all MOM devices, compiles them, and then displays them on the web application for users to see. The web application is available to the public and can be accessed on any device with a web browser.

# 2 Design

## 2.1 Control Unit

### 2.1.1 Design Procedure

The Control Unit of each MOM device only requires a microcontroller to process occupancy data and a Wi-Fi transceiver to send this data to the IoT Cloud. Thus, the design procedure for the Control Unit was to find a suitable System on a Chip (SoC) with both of these features. There are many SoC manufacturers who develop products meeting our requirements, including Nordic Semiconductors and Silicon Labs. However, the products from these manufacturers are not only expensive, but many were, and still are, out of stock. However, one SoC platform stands out from the rest for our project. Espressif's ESP32 family of SoCs are not only widely available, but they are also inexpensive and popular within the Internet of Things industry.

### 2.1.2 Design Details

The ESP32-S3-MINI-1 SoC was selected as the SoC for MOM devices due to its low cost, small form factor, and robust development platform. It consists of a ESP32-S3 microcontroller and a programmable Wi-Fi antenna, which makes integrating the Radio Scanner Suite simple. Additionally, we included an LED which connects to one of the microcontroller's General-Purpose Input/Output (GPIO) pins to indicate when the monitoring program is active.

In order to program the microcontroller, there needs to be an interface to power it and connect it to a development computer. The ESP32-S3-MINI-1 has a native USB interface for this purpose, so connecting the USB GPIO pins to the corresponding pins on a USB connector is enough to create this interface. Two buttons used for restarting and setting the SoC into program mode are also present in the design. The schematic for the Control Unit can be seen in Figure 3 on the following page.

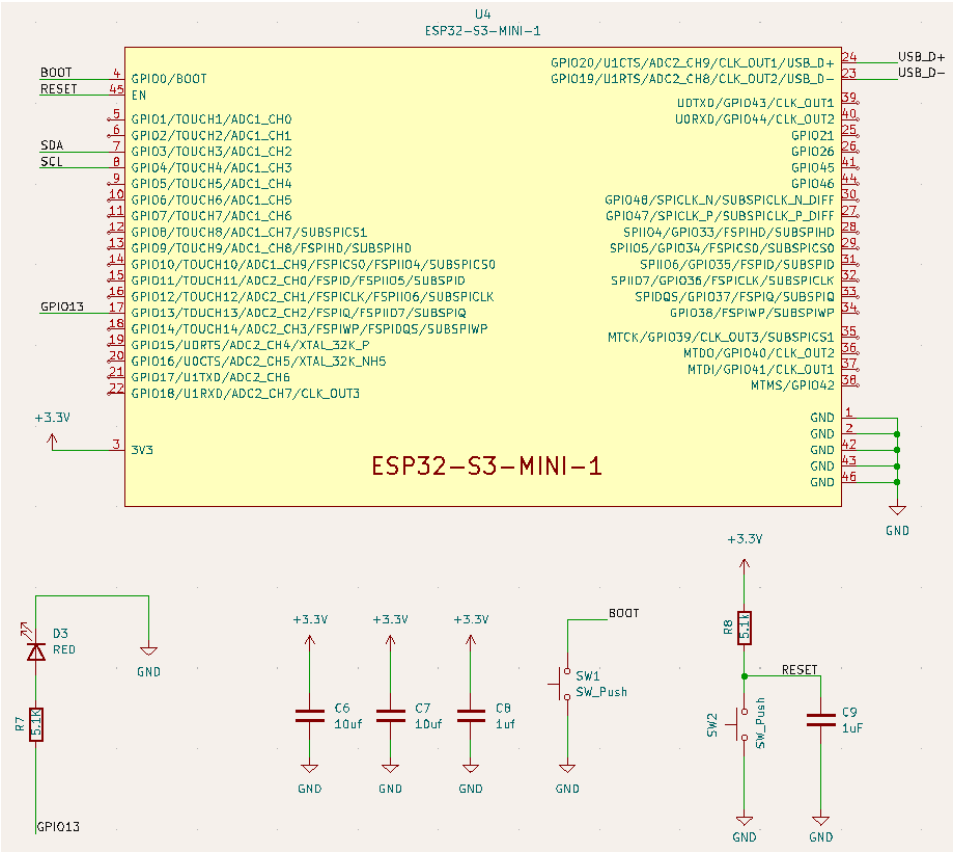



Figure 3: Schematic of the Control Unit

## 2.2 Radio Scanner Suite

### 2.2.1 Design Procedure

Since it will be monitoring Wi-Fi traffic, the Radio Scanner Suite must have an antenna and Wi-Fi controller capable of operating under Wi-Fi 4, also known as the IEEE 802.11n protocol, or newer. One of the initial designs of a MOM device involved creating the Radio Scanner Suite from scratch, which would involve purchasing and programming a separate companion chip and connecting it to antennas. This would not only be complex and time consuming, but this design later became redundant since the ESP32-S3-MINI-1 already has a Wi-Fi interface with an antenna. Ultimately, the Radio Scanner Suite was implemented using the Wi-Fi interface on the ESP32-S3-MINI-1.

### 2.2.2 Design Details

The Wi-Fi interface in the Radio Scanner Suite is fully controlled by the microcontroller in the Control Unit. Although the Wi-Fi interface can switch between the 11 available channels under the 2.4 GHz frequency band, we decided to only monitor Channel 1. This is because probe requests, the type of traffic we are targeting, are transmitted on every Wi-Fi channel to search for available access points. This ultimately means that it does not matter which channel the Radio Scanner Suite is targeting.

After collecting occupancy data for a set period, the data needs to be transmitted to the IoT Cloud to keep the web application updated. The microcontroller can programmatically switch the Wi-Fi interface between monitor mode for occupancy monitoring and station mode for data transmission. Thus, the microcontroller was programmed to switch to station mode every one or two minutes, send the occupancy data, and switch back to monitor mode when done.

## 2.3 Power Supply

### 2.3.1 Design Procedure

Ideally, MOM devices deployed in a building would be plugged into a wall outlet. Since a USB interface already exists for the purpose of powering and programming the microcontroller, other components on the device can utilize the same bus to draw power. This means that any off-the-shelf USB wall plug and USB-C cable can be used to power a MOM device.

In addition, we wanted each MOM device to have the ability to instantly switch to backup battery power so that each device can still operate in the event of a wall power disruption. While 18650 batteries would be able to deliver the necessary power, lithium-polymer batteries were chosen as the battery type due to the slimmer form factor. To notify us of the state of charge of the backup battery, we selected the MAX17048G as the battery fuel gauge since it requires minimal configuration, is designed specifically for lithium-polymer batteries, and is able to communicate with the Control Unit to report this information. We also thought it would be a good idea to charge the backup battery when the device is powered by the wall outlet.

### 2.3.2 Design Details

The ESP32-S3-MINI-1 SoC, which houses the Control Unit and Radio Scanner Suite, requires a voltage of 3.3 volts for safe operation [7]. To step the input voltage down to 3.3 V, we use an AP2112 low-dropout voltage regulator. It is able to step down both the five-volt input from USB and the average of 3.7 V from the backup battery to 3.3 V.

However, only one power source should be used to power the device at a time. For a MOM device, this means deciding between wall power or battery power. Since we only want the

battery to be used when wall power is not available, we use a P-Channel MOSFET in conjunction with a Schottky diode as a decision circuit. As illustrated in Figure 4 on the next page, when wall power is present, the P-Channel MOSFET prevents power from the battery going to the device. When wall power is not present, the P-Channel MOSFET acts as a short, and the battery is used to power the device, with the Schottky diode blocking the battery power from going to the USB line.

Since the device needs to be able to run on battery power for at least one hour, the capacity of the backup battery needs to be at least a certain value. To calculate the minimum battery capacity, we used Equation 1 below. The ESP32-S3-MINI-1 has a current draw of 500 mA [7]. Since we want each device to run for at least one hour, the battery capacity must be at least 500 mAH. During our testing, the MOM device was able to run for about five hours while powered by the battery. The results of the battery life test can be seen in Figure 13 in Appendix D.

$$Capacity\ [mAH] = I_{VDD}\ [mA]\ *\ Runtime\ [hours] = 500\ [mA]\ *\ 1\ [hour] \qquad (1)$$

To charge the battery while connected to wall power, the USB power line is also connected to a battery charger, which consists of an MCP73831 lithium-polymer battery charge controller. The charging rate of the MCP73831 can be programmed by using a programming resistor, allowing the charging rate to be optimized to the specific battery the controller is charging. Equation 2 shows the formula to calculate the resistance of the programming resistor to optimize the charging rate [8]. Since the backup battery will have a capacity of 500 mAH, this means the regular charge current, or the one-cell charging rate, is 500 mA. From the formula, we can see that the resistance of the programming resistor should be 2 kΩ. The charging rate of the battery charger can be seen in Figure 14 in Appendix D, and the circuit schematic can be seen in Figure 5 on the next page.

$$R_{prog}[\Omega] = \frac{1000\ V}{I_{reg}\ [A]} = \frac{1000\ V}{0.5\ A} = 2\ k\Omega \qquad (2)$$

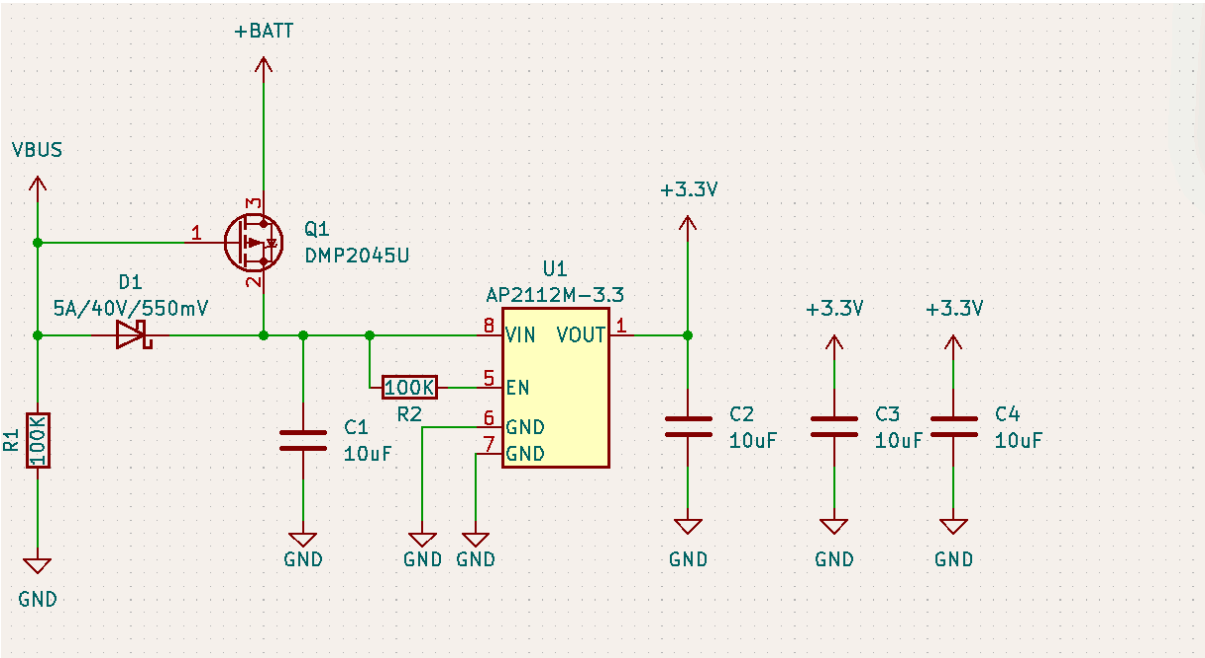



Figure 4: Schematic of the Power Source Decision Circuit

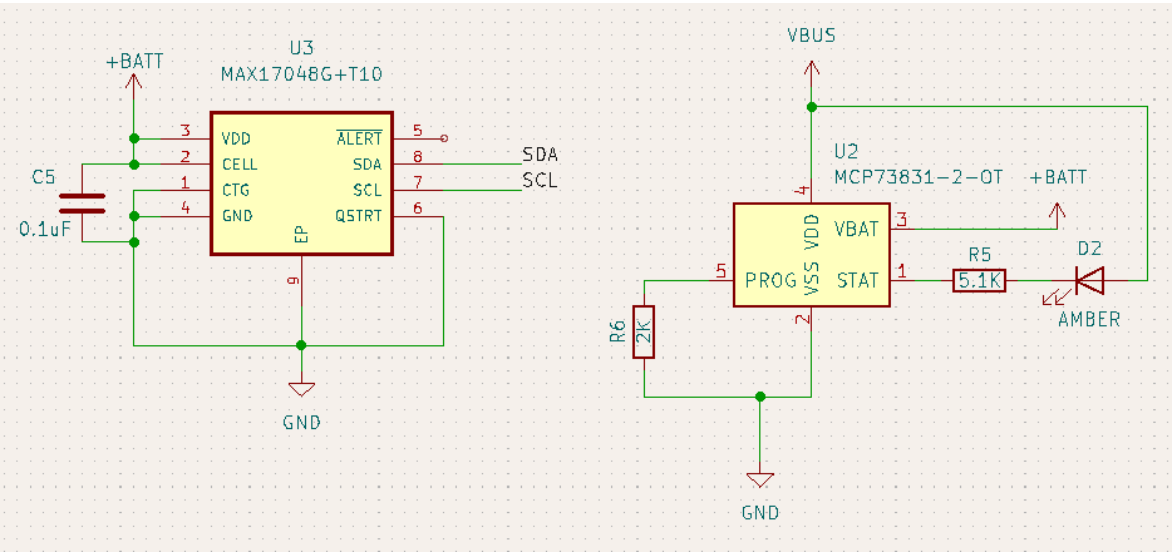



Figure 5: Schematic of the Battery Charger and Fuel Gauge

## 2.4 IoT Cloud

### 2.4.1 Design Procedure

The IoT Cloud subsystem consists of the web application itself as well as the infrastructure to host it, store occupancy data, and receive messages from MOM devices. We selected AWS as our cloud provider of choice since it has all of the cloud services we need such as a database, IoT core, and web application hosting platform. AWS also has ways to seamlessly integrate these different services together with minimal setup.

For the web application, we needed a simple full-stack application that was both deployable on AWS and capable of communicating with the AWS database. Today, there are a plethora of publicly available and open-source web development frameworks that are capable of both. However, we needed a framework that was minimal since we wanted our web application to be fast and scalable.

### 2.4.2 Design Details

Since we needed a web development framework that was easy to learn and performant, we chose to develop it under the Python Flask framework. Python Flask is a full-stack micro framework that is easy to develop under and configure. It can be deployed on AWS using AWS Elastic Beanstalk and, just like most other Python frameworks, can utilize the vast collection of Python modules available. One such module is Amazon's official module to interact with AWS, Boto3.

For the underlying AWS infrastructure, we designed a flow that enables data from the MOM device to be sent to the web application for users to see, which can be seen in Figure 6 below. First, each MOM device will use the MQTT protocol to publish occupancy data to the AWS IoT Core. Then, we have the IoT Core receive each message and store them in the occupancy metrics database. From there, whenever a user accesses the web application, the backend of the web application will pull the data from the database and send it to the user interface. The web application user interface can be seen in Appendix B.

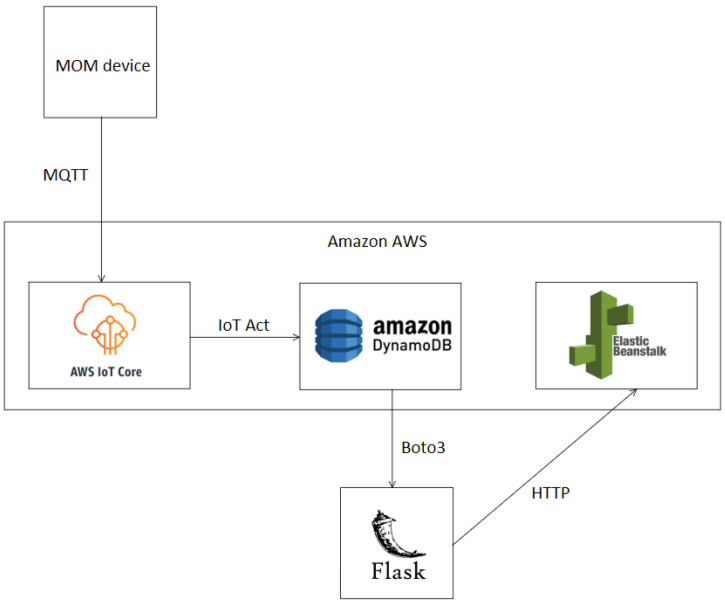



Figure 6: Block Diagram of the AWS Infrastructure

## 2.5 Occupancy Estimation Algorithm

### 2.5.1 Design Procedure

The simplest approach to occupancy estimation by monitoring Wi-Fi traffic would be simply counting the number of devices seen within set periods. With enough fine-tuning, this approach would be sufficient to determine the relative occupancy of a space (i.e. how full the space is represented as a percentage). However, the primary goal of the project is to estimate the exact number of people with at least 80% accuracy.

Monitoring Wi-Fi traffic to estimate the exact number of people in an enclosed area requires careful observation of multiple factors. In addition to the number of Wi-Fi enabled devices seen, it also requires observing the signal strength of each seen device, the dimensions of the enclosed area, the timestamps of when each device was seen, and devices that may be part of the environment (e.g. wireless routers, Wi-Fi enabled printers, digital signage). By accounting for these factors, we are able to mitigate the possibility of overcounting the number of people in the area the device is deployed in.

### 2.5.2 Design Details

The aggregation of these factors allows for a more accurate estimation of the number of people in the area. Since we are using all of these factors, we came up with a model that determines the probability that a seen device contributes to exactly one occupant in the room. This model focuses on three possibilities. The first is if the seen device belongs to a person in the target area

and not outside the area, the second is if the seen device could be the person's second device, and the third is if the seen device belongs to the environment as stated previously. These are illustrated in the flowchart in Figure 7 below.

To calculate the probabilities, conditional checks are used in conjunction with the timestamps and signal strength of each seen device's probe request. Devices which haven't been seen after a set period, fall outside of the defined signal strength range, or are part of the environment are given a probability of zero. The stronger the device signal and the more recently the device has been detected, the higher the probability. The probabilities are recalculated multiple times in one second, so as time goes on, the probability assigned to a device may decrease.

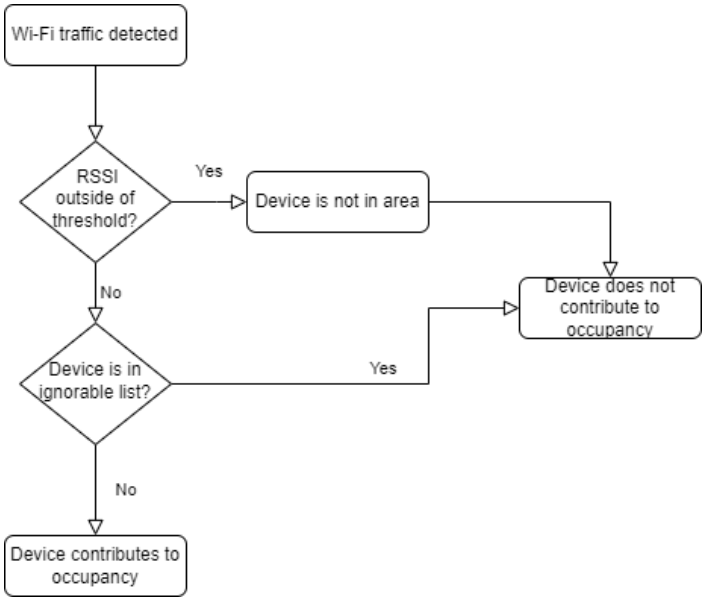



Figure 7: Probability Model Flowchart

# 3 Design Verification

The Requirements & Verification Tables can be found in Appendix A.

## 3.1 Control Unit

- The microcontroller on the MOM device powers on and is able to be programmed by a development computer over USB.

This requirement confirms that the device not only works, but is also able to be programmed with the monitoring program. After running through the verification of the Control Unit, we were able to connect it to a development computer to program it with the verification program and power it, verifying this requirement.

## 3.2 Radio Scanner Suite

There were no requirements to verify for the Radio Scanner Suite as it only consisted of the included Wi-Fi interface of the ESP32-S3-MINI-1 SoC.

## 3.3 Power Supply

### 3.3.1 Battery Charger

- The battery charger must be able to charge the battery to at least 3.7 V, but no more than 4.3 V, in 12 hours or less.

In the event of intermittent wall power disruptions, the battery needs to be as charged as possible in order to power the device for at least one hour. As we can see in Table 1 below, this requirement was met. The battery was charged to 3.7 V within 20 minutes, and charged to 4.19 V within 80 minutes. After being plugged in for an additional 100 minutes, the battery charging circuit stopped so that the battery did not overcharge, verifying this requirement.

**Table 1: Battery Voltage While Charging Over Time**

| Time (minutes) | Battery Voltage (V) |
|---|---|
| 0 | 3.09 |
| 20 | 3.71 |
| 80 | 4.19 |
| 180 | 4.1 |

- The battery charger must be able to report the battery level to the control unit with at least 80% accuracy.

This requirement is important so that if there was an issue with the device, we could determine if it is a battery issue or not by checking the battery's state of charge. During the verification, the multimeter reported a battery voltage of 4.15 V, while the battery charger reported a voltage of 4.14 V, which is within the tolerance of 20%.

### 3.3.2 Power Source Decision Circuit

- The decision circuit must be able to switch to wall power and supply a stable voltage of 3.3 V ± 0.3V.

This requirement ensures the safe transition from battery power, or no power at all, to wall power. After performing the verification steps, the multimeter probe read 3.32 V while the device was connected to wall power. 3.32 V is within the specified tolerance, verifying this requirement.

- The decision circuit must be able to switch to battery power and supply a stable voltage of 3.3V ± 0.3V.

The transition from wall power to battery power needs to keep the SoC at a safe voltage level. After performing the verification steps, the multimeter probe read 3.31 V while the device was powered only by the battery. 3.31 V is within the specified tolerance, verifying this requirement.

## 3.4 IoT Cloud

### 3.4.1 MQTT Service

- Each MOM device should be able to connect to AWS servers with a round-trip latency of no greater than 500 milliseconds.

If the round-trip latency from the device to AWS is greater than 500 milliseconds, that may be indicative of a weak connection. This requirement enforces a reliable connection before sending occupancy data. As we ran the ping tests as part of the verification for this requirement, the average round-trip latencies were all less than 30 milliseconds while connected to IllinoisNet.

- Each MOM device should be able to post three messages to the MQTT service in no more than 10 seconds.

We want to ensure the stability of the cloud infrastructure should multiple devices publish to the MQTT service in rapid succession. As part of the verification for this requirement, a MOM device attempts to publish three messages all within one second. The MQTT service was able to

handle all three messages and relay them in slightly more than one second, which is less than ten seconds.

### 3.4.2 Web Application

- The user interface of the application loads in less than 10 seconds.

An application which loads quickly provides an enhanced user experience. As part of the verification for this requirement, we simply access the web application and see how long it takes to load. On every occasion, it loads in less than ten seconds.

- The user interface must be responsive (i.e. scalable and readable on smartphones, tablets, laptops, and computer monitors).

Responsive web applications enhance the user experience by making it easy to read the content no matter what device is being used. The verification for this requirement is simply making sure that the contents of the web application are properly scaled to the screen size of the device it is accessing. This requirement was verified, and screenshots of this can be seen in Appendix B.

# 4 Cost and Schedule

## 4.1 Parts

The parts list for a single MOM device can be seen in Table 2 below. Many of the parts are optional, such as the enclosure, and many of them can be repurposed from other products, such as the USB wall adapter and cable. Omitting optional and repurposed parts from the cost would greatly reduce the cost of a single MOM device. If one is planning on building multiple devices, additional discounts are given by purchasing these parts in bulk from distributors such as Digi-Key or Mouser Electronics.

**Table 2: Cost Breakdown Table**

| Part (With Purchase Link) | Quantity (per node) | Extended Price (per node) |
|---|---|---|
| ESP32-S3-MINI-1-N8 SoC | 1 | $3.41 |
| GCT USB 2.0 Type-C Receptacle | 1 | $0.94 |
| AP2112 (3.3V) Linear Voltage Regulator | 1 | $0.42 |
| MCP73831-2ACI Battery Charge Controller | 1 | $0.80 |
| MAX17048G+T10 Battery Fuel Gauge | 1 | $3.34 |
| 3.7V 500mAh Li-Po Battery (JST-PH) | 1 | $7.95 |
| 2-pin JST-PH Header | 1 | $0.17 |
| DMP2045UQ MOSFET | 1 | $0.53 |
| 5A/40V Schottky Diode | 1 | $0.54 |
| Surface Mount Chip Resistor 100KOhm | 2 | $0.20 |
| Surface Mount Chip Resistor 5.1KOhm | 5 | $0.50 |
| Surface Mount Chip Resistor 2KOhm | 1 | $0.10 |
| Surface Mount Capacitor 0.1uF | 1 | $0.36 |
| Surface Mount Capacitor 10uF | 7 | $2.16 |
| Surface Mount Capacitor 1uF | 2 | $0.72 |
| Through-Hole Red LED | 1 | $0.35 |

(continued on next page)

| | | |
|---|---|---|
| Through-Hole Yellow LED | 1 | $0.15 |
| Tactile Button Switch | 2 | $0.36 |
| USB Wall Adapter | 1 | $5.00 |
| USB-C to USB-A cable | 1 | $5.89 |
| PCB | 1 | $5.00 |
| LP-51F Polycase (Optional) | 1 | $6.13 |
| Total Parts Cost | | $45.02 |

## 4.2 Labor

The average salary of a computer engineering graduate from UIUC is **$105,352** [9]. Assuming that a full-time engineer has a 40-hr work week and the engineer works an average of 48 weeks per year, then the engineer works 1,920 hours per year. This equates to $54.87 per hour. Our project timeline from here on out is 12 weeks. Assuming we spend 10 hours on this project per week then we would be spending a total of 120 hours on this project to complete it. We can calculate the labor cost per team member using Equation 3 below:

$$(\$54.87/hour) * 2.5 * 120\ hours = \$16,461 \quad (3)$$

From Equation 3, we can see that the labor cost for one team member would be $16,461. Since all three members of our team are computer engineers, then the total labor cost for the team would be $49,383.

# 5 Conclusion

## 5.1 Accomplishments

We believe that this project was an overall success. All high-level and subsystem requirements were met, with the MOM device estimating occupancy with an accuracy of more than 80% during our testing. Additionally, the IoT Cloud infrastructure is scalable should we make more MOM devices and deploy them to monitor additional additional areas.

## 5.2 Uncertainties

Unfortunately, creating a general model that works for any kind of room or enclosed space proved difficult. Trying to detect if a person has multiple Wi-Fi devices is extremely difficult to detect. This is due to the privacy features of modern devices and the fact that using signal strength values to estimate distance yields the radius of a sphere. To make things simple, our algorithm runs under the assumption that the signals it is catching are on the same floor of the building it is deployed on. Even though it is likely to capture signals from devices on adjacent floors, we were able to fine-tune the parameters of our algorithm to overcome these difficulties when deployed in larger rooms. However, when the device is deployed in a smaller room, the accuracy of the algorithm decreases dramatically.

## 5.3 Ethical Considerations

One of the most significant ethical considerations we had to take into account was privacy, which directly relates to section I-1 of the IEEE Code of Ethics [10]. The project was designed with maintaining the privacy of individuals in mind. MOM devices do not read or collect data contained within Wi-Fi packets themselves. The only identifiable data in probe requests are Wi-Fi MAC addresses, which themselves are public information. Wi-Fi MAC addresses can only identify a particular device, but cannot be used to identify an individual.

Pursuant to Section I-2 of the IEEE Code of Ethics [10], students and others in public spaces have the right to know when and where this monitoring is occurring and how our devices work. Therefore, if we ever put our MOM devices to production in any study spaces, there would be a clear sign to indicate to the occupants of the room the intention of the device and how it collects data.

Finally, the health and safety of the public as described in Section I-1 of the IEEE Code of Ethics [10] is paramount. As each MOM device will have lithium-polymer batteries, it is important to ensure that they stay in safe operating conditions. We have performed extensive testing to make sure the battery cannot be overcharged, the results of which can be seen in Appendix E.

## 5.4 Future Work

There are various improvements and applications we can derive from our initial prototype of the MOM device. Though our current model is successful, we can develop a stronger and more accurate model to improve the overall efficacy of the occupancy approximation determination. For instance, we can deploy several MOM devices (as opposed to just one) over an extended stretch of time, accruing more data. This could help fine tune the precision of the room's occupancy metrics.

Additionally, our MOM device relied on probing Wi-Fi signals to estimate occupancy. We can expand this by probing Bluetooth signals as well. Then we can compute a "weighted average" between cumulative Bluetooth and Wi-Fi traffic to better capture how busy a specific room is.

Right now, our project only supports one room. However, our AWS and web application infrastructure was set up in a manner where it can be easily scalable if needed. Therefore, if we can expand this project to a wider scale, then we could put a MOM device in each study room that we would like to monitor occupancy data in. The web application would also be updated appropriately to support this and would have a page navigation where the user is able to browse through and select the room of interest to them.

Another future improvement could be to include a graph of the most popular times when a room is busy and usually open. This graph would go right below our occupancy pie chart on our web application. The graph would forecast when a room is most available based on historical occupancy data that we would have collected and stored in our database. Using this approach we could get a rough estimate of the number of people for a given future time. To forecast future occupancy, we could also use more sophisticated models such as an LSTM (Long Short-Term Memory) model which is a type of recurrent neural network that has vast applications when given time series data as inputs [11]. It can predict pretty accurately when given proper data points and trained appropriately.

# References


[1] M. Thaler, C. Barber, and T. Pittman, "Students on Libraries - Student Views on the Academic Library of Today and Tomorrow," Gensler, 2015. Accessed: Sep. 14, 2022. [Online]. Available: https://www.gensler.com/doc/students-on-libraries

[2] A. Raychawdhuri. "Lack of study spaces at UCLA imposes unnecessary stress on students." The Daily Bruin, Feb. 28, 2019. Accessed: Sep. 14, 2022. [Online]. Available: https://dailybruin.com/2019/02/28/lack-of-study-spaces-at-ucla-imposes-unnecessary-stress-on-students

[3] L. Oliveira, D. Schneider, J. De Souza, and W. Shen, "Mobile Device Detection Through WiFi Probe Request Analysis," in IEEE Access, vol. 7, pp. 98579-98588, 2019, doi: 10.1109/ACCESS.2019.2925406.

[4] Occuspace, Inc. "Waitz," 2019. Accessed: Sep. 12, 2022. [Online]. Available: https://waitz.io/

[5] LiveOverflow, Berlin, Germany. Student Finds Hidden Devices in the College Library - Are They Nefarious?, Nov. 9, 2018. Accessed: Sep. 12, 2022. [Online Video] Available: https://www.youtube.com/watch?v=UeAKTjx_eKA

[6] E. Upton, "Production and supply-chain update," Raspberry Pi, Apr. 4, 2022. Accessed: Sep. 12, 2022. [Online]. Available: https://www.raspberrypi.com/news/production-and-supply-chain-update/

[7] "ESP32-S3-MINI-1 ESP32-S3 MINI-1U Datasheet", Espressif, Sep. 2022. Accessed: Sep. 12, 2022. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-mini-1_mini-1u_datasheet_en.pdf

[8] "MCP73831/MCP73832 Data Sheet," Microchip Technology, June 2020, Accessed: Nov. 2, 2022. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73831-Family-Data-Sheet-DS20001984H.pdf

[9] The Grainger College of Engineering, “Computer Engineering,” University of Illinois Urbana-Champaign. Accessed: Sep. 25, 2022. [Online]. Available: https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/computer-engineering

[10] IEEE Board of Directors, "IEEE Code of Ethics," Institute of Electrical and Electronics Engineers, 2020. Accessed: Sep. 14, 2022. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[11] J. Brownlee. “How to Develop LSTM Models for Time Series Forecasting,” Machine Learning Mastery. Accessed: Nov. 22, 2022. [Online]. Available: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

# Appendix A: Requirements & Verification Tables

**Table 3: Control Unit RV Table**

| Requirements | Verification | Verified |
|---|---|---|
| 1. The microcontroller on the MOM device powers on and is able to be programmed by a development computer over USB. | 1A. Connect the device with a USB port on a development computer.<br>1B. Create a simple "Hello World!" program that involves blinking the LED on the microcontroller's GPIO13 pin.<br>1C. Flash the program onto the microcontroller. Verify that the program is executing correctly and the LED on GPIO13 is blinking. | Y |

**Table 4: Battery Charger RV Table**

| Requirements | Verification | Verified |
|---|---|---|
| 1. The battery charger must be able to charge the battery to at least 3.7V, but no more than 4.3V, in 12 hours or less. | 1A. Allow the battery to discharge over time until it reads 3.1V.<br>1B. Plug in the device to wall power. Monitor battery voltage using a multimeter periodically. When the battery voltage reads 3.7V through an oscilloscope or multimeter, record the charge cycle time and verify that it is less than 12 hours.<br>1C. Allow the battery to continue charging until it has reached around 4.2V. Measure the battery voltage using either an oscilloscope or multimeter and ensure that the battery reads 4.2V ± 0.1V.<br>1D. Allow the battery to continue charging for an additional hour. Measure the battery voltage using either an oscilloscope or multimeter and ensure that the battery reads no more than 4.3V. | Y |

(continued on next page)

| | | |
|---|---|---|
| 2. The battery charger must be able to report the battery level to the control unit with at least 80% accuracy. | 2A. Write a simple program to print the voltage of the battery. Flash this program onto the microcontroller.<br>2B. While the program is running, measure the battery voltage by probing it using a multimeter or oscilloscope.<br>2C. Ensure that the value printed by the microcontroller is within 80% of the voltage read by the oscilloscope or multimeter. | Y |

**Table 5: Power Source Decision Circuit RV Table**

| **Requirements** | **Verification** | Verified |
|---|---|---|
| 1. The decision circuit must be able to switch to wall power and supply a stable voltage of 3.3V ± 0.3V. | 1A. Connect the device to wall power and boot it.<br>1B. Using a multimeter or oscilloscope, probe the Vout pin of the voltage regulator with the positive probe and a GND pin with the negative probe.<br>1C. Confirm that the voltage read is 3.3V ± 0.3V. | Y |
| 2. The decision circuit must be able to switch to battery power and supply a stable voltage of 3.3V ± 0.3V. | 2A. Ensure that the battery backup is connected and disconnect the device from wall power. Boot the device if it hasn't already been booted.<br>2B. Using a multimeter or oscilloscope, probe the Vout pin of the voltage regulator with the positive probe and a GND pin with the negative probe.<br>2C. Confirm that the voltage read is 3.3V ± 0.3V. | Y |

**Table 6: AWS MQTT Service RV Table**

| **Requirements** | **Verification** | Verified |
|---|---|---|
| 1. Each MOM device should be able to connect to AWS servers with a round-trip latency of no greater than 500 milliseconds. | 1A. Create a simple program which can ping AWS servers and report the round-trip latency.<br>1B. Connect the microcontroller to the development computer and flash the program to the microcontroller.<br>1C. Run the ping program and confirm that the round-trip latency is no greater than 500 milliseconds. | Y |
| 2. Each MOM device should be able to post three messages to the MQTT service in no more than 10 seconds. | 2A. Create a simple program which pushes three separate messages to the MQTT Service.<br>2B. Connect the microcontroller to the development computer and flash the program to the microcontroller.<br>2C. Set up the MQTT service to process occupancy information events.<br>2D. Run the program. Once the program finishes, confirm that all three events have been processed within 15 seconds ± 5 seconds in the AWS MQTT Message Monitor. | Y |

**Table 7: Web Application RV Table**

| **Requirements** | **Verification** | Verified |
|---|---|---|
| 1. The user interface of the application loads in less than 10 seconds. | 1A. Open a new tab on any device with a web browser that has a strong connection to the internet.<br>1B. Enter the address of the web application and start a stopwatch.<br>1C. Confirm that the web application's user interface loads in less than 10 seconds. | Y |
| 2. The user interface must be responsive (i.e. scalable and readable on smartphones, tablets, laptops, and computer monitors). | 2A. Open a new tab on a smartphone that has a strong connection to the internet.<br>2B. Enter the address of the web application.<br>2C. Confirm that the user interface is readable and properly scaled to the size of the smartphone.<br>2D. Open a new tab on a laptop that has a strong connection to the internet.<br>2E. Enter the URL/address of the application.<br>2F. Confirm that the user interface is readable and properly scaled to the size of the laptop's display. | Y |

# Appendix B: Web Application User Interface

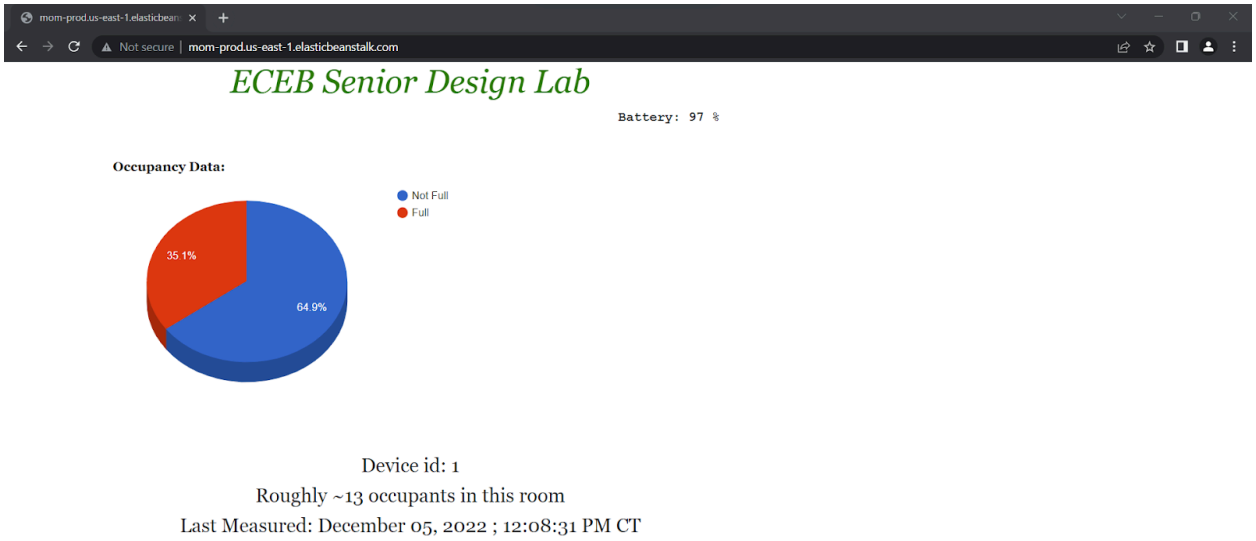
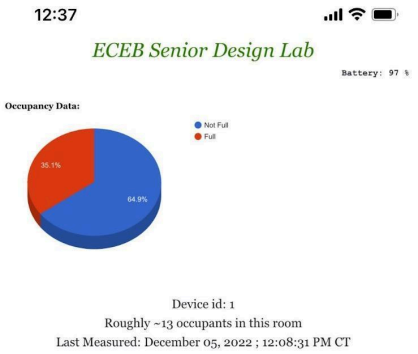



Figure 8: Web Application User Interface on Laptops and Desktops



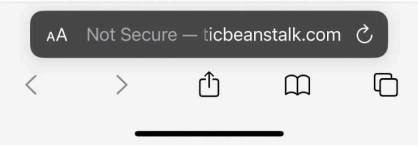




Figure 9: Web Application User Interface on an iPhone 12

# Appendix C: Schematic, PCB, and Final Product

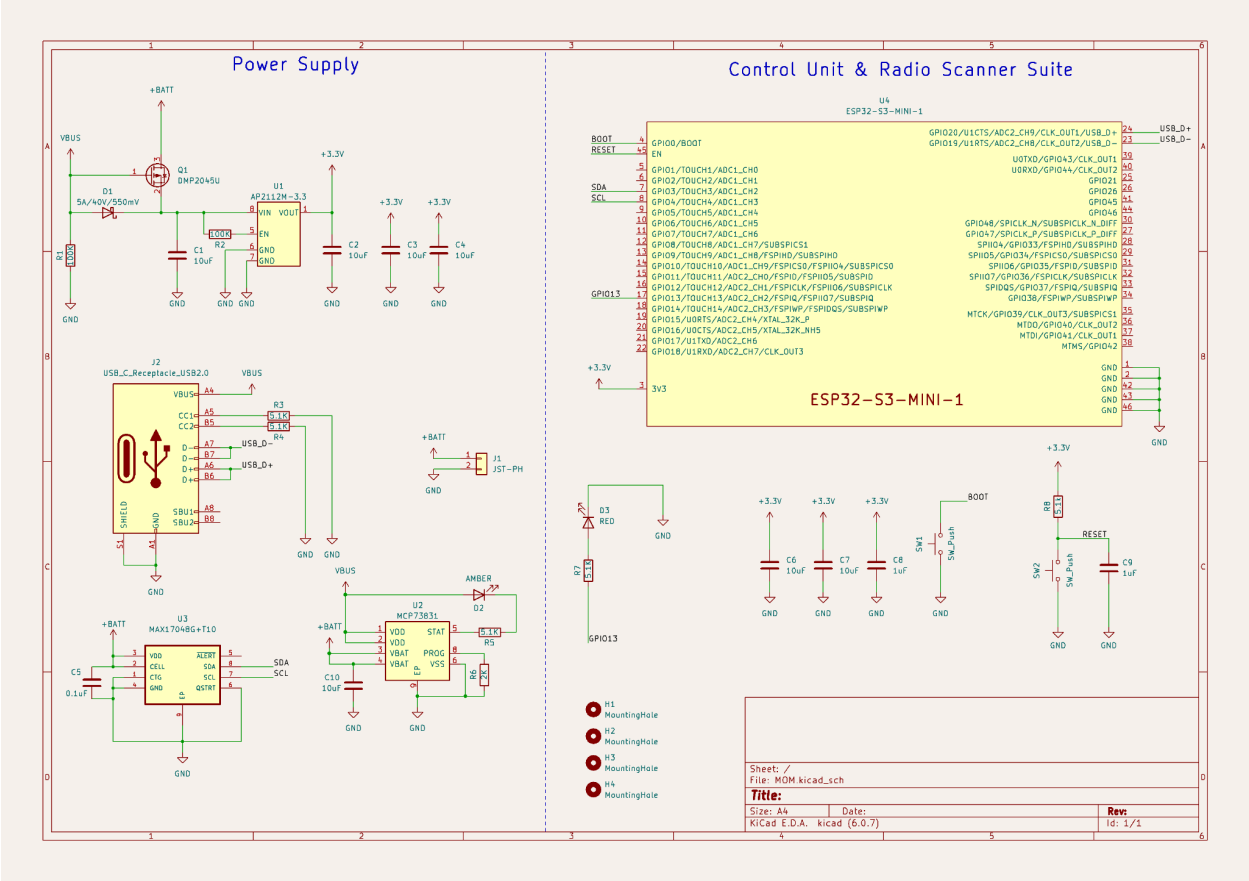



Figure 10: MOM Device Full Schematic

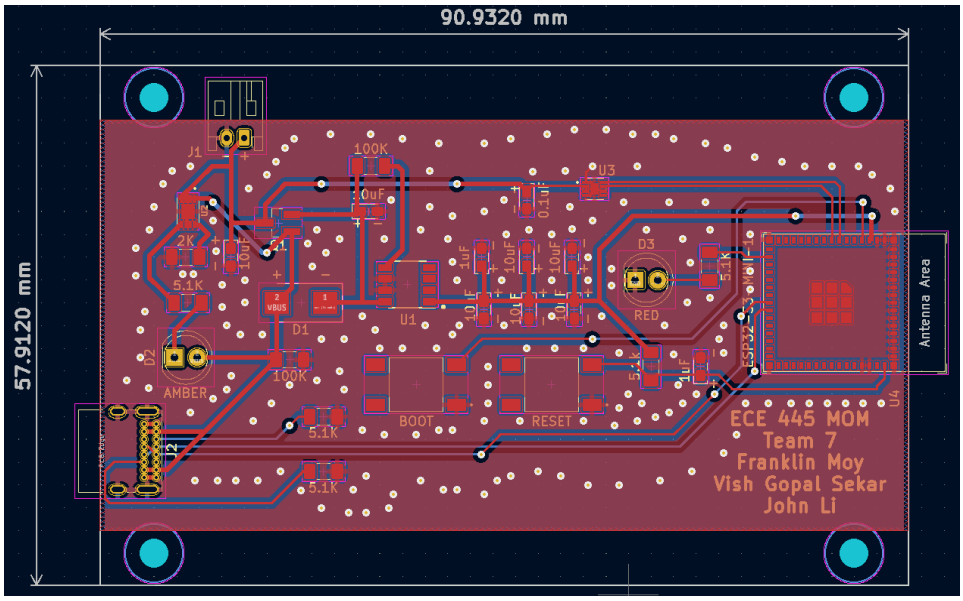
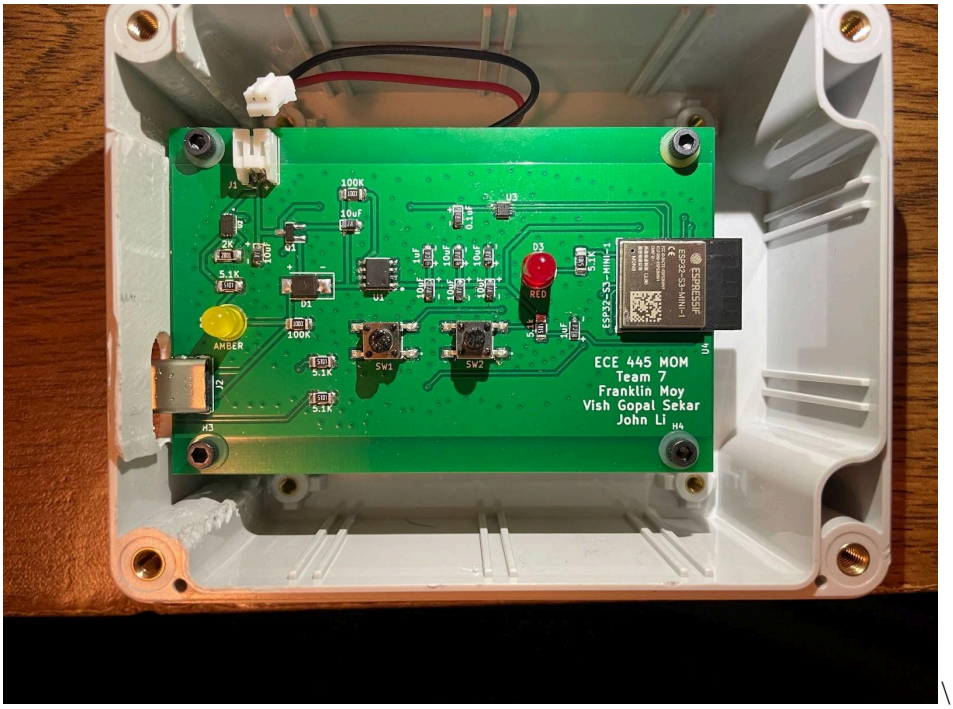



Figure 11: MOM Device PCB Layout



\

Figure 12: MOM Device in Enclosure

# Appendix D: Battery Test Results

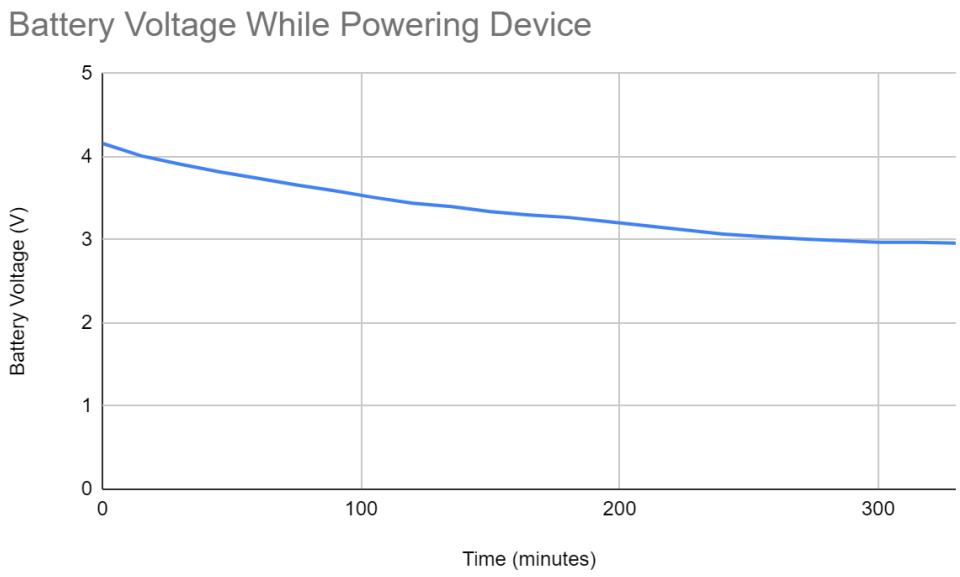



Figure 13: Battery Voltage While Powering Device During Testing

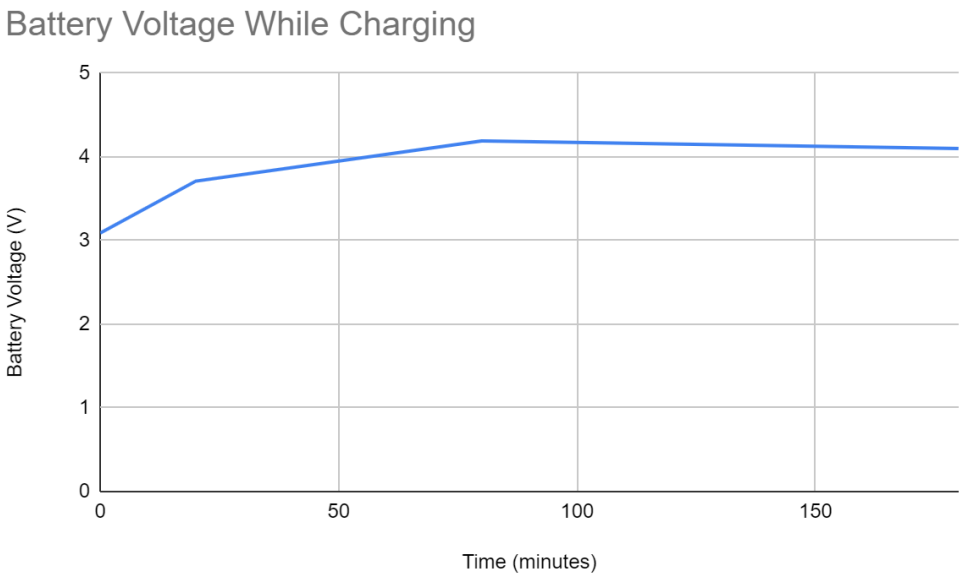



Figure 14: Battery Voltage While Charging During Testing

# Appendix E: Battery Charger Overcharge Test

An additional current cutoff test was conducted to ensure the battery charger would not overcharge the backup battery. Once a lithium-polymer battery reaches about 4.2 V, the charging circuit should cut off power to the battery to maintain the health of the battery. Overcharging a lithium-polymer battery can lead to swelling and fire.

A circuit imitating a battery was created and connected to the battery header of the device. The voltage of the imitation battery can be adjusted using a potentiometer. An ammeter was connected in series between the charge output of the battery charger and the positive terminal of the imitation battery to measure the charge current at various voltages. The results of the test can be seen in Table 8 below, which shows that the battery charger properly cuts off power when the battery voltage goes above 4.2 V.

**Table 8: Mock Battery Charging Test Readings**

| Mock Battery Voltage (V) | Charge Current (A) |
|---|---|
| 3.31021 | 0.524102 |
| 3.72014 | 0.523611 |
| 4.06172 | 0.523125 |
| 4.21920 | 0.000004 |
| 4.30182 | 0.000004 |

# Appendix F: Monitoring Program Flow

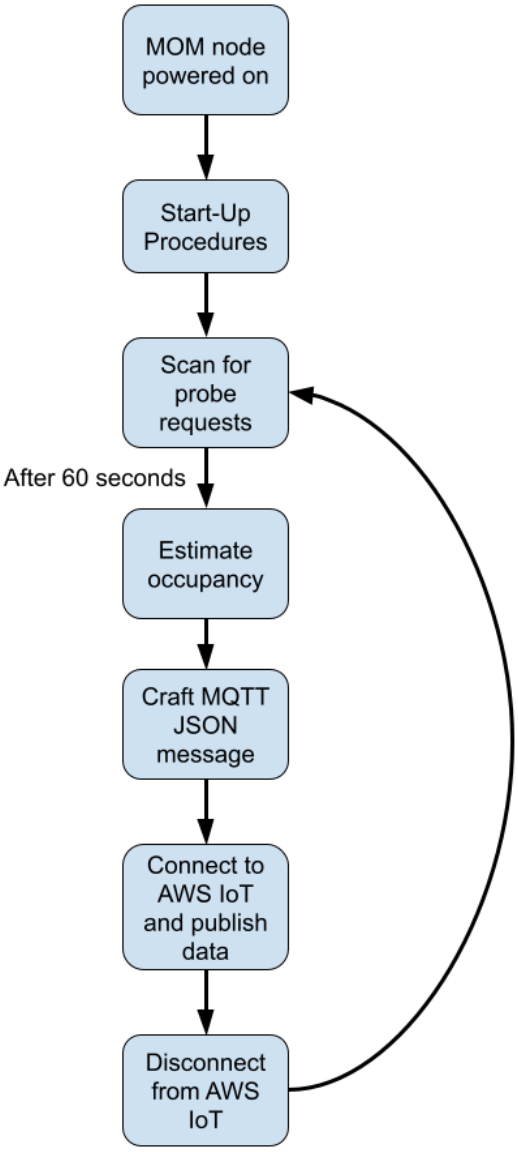



Figure 15: Flow of the Monitoring Program