ECE 445

Senior Design Laboratory

Fall 2022

Design Document

Microcontroller-based Occupancy Monitoring (MOM)

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

TA: Hanyin Shao

9/29/2022

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 and Bluetooth signals, scanning for Wi-Fi and Bluetooth 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 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

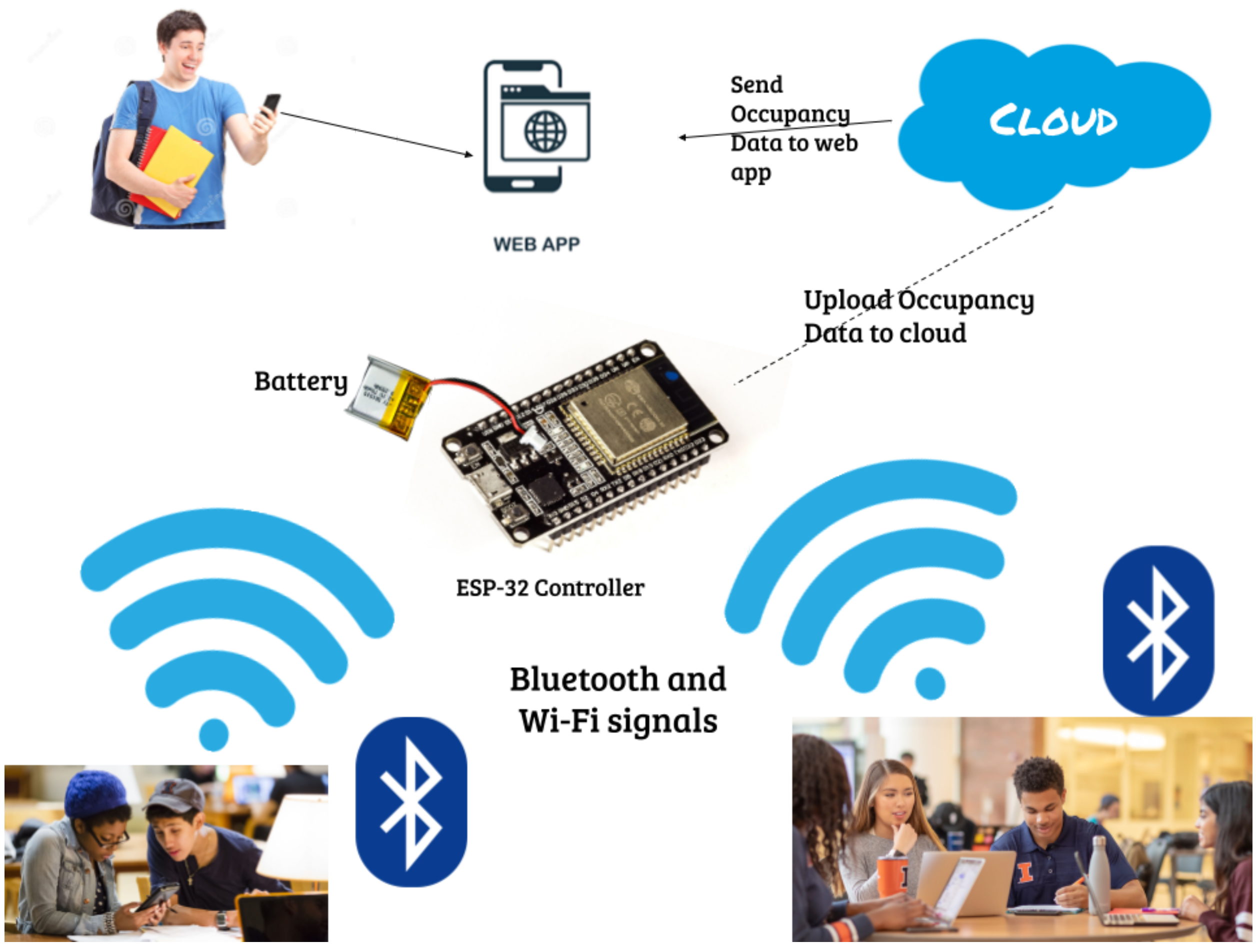

Figure 1: Interactions between MOM device, student devices, and web application

1.4 High-Level Requirements

- Each device must be able to estimate sector occupancy with at least 80% accuracy.
- Each device must be able to run indefinitely while plugged into wall power and for at least one hour when unplugged.
- The web app must consistently update such that it shows data which is no more than 5 minutes old.

2 Design

2.1 Block Diagram

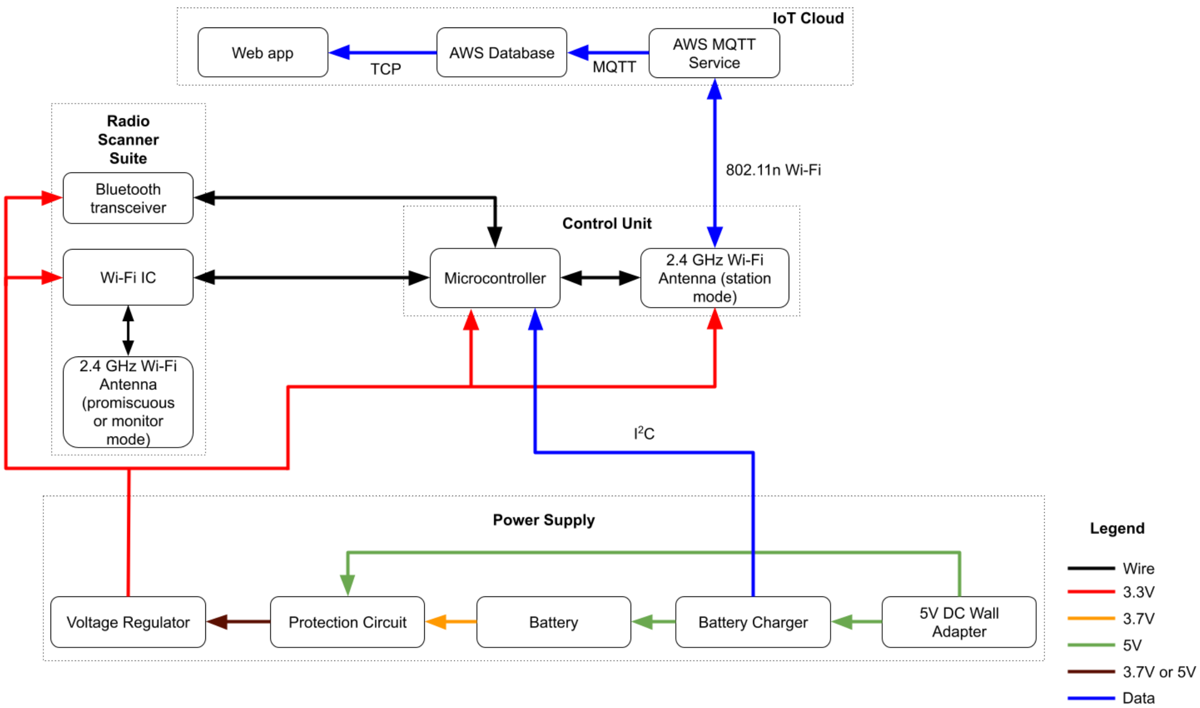

Figure 2: Block Diagram

2.2 Subsystem Overview

2.2.1 Control Unit

The Control Unit receives information about Wi-Fi and Bluetooth traffic from the Radio Scanner Suite. It will then do some preliminary processing on 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.

2.2.1.1 Microcontroller (ESP32-S3)

The Espressif ESP32 family of microcontrollers are mature products that are well-known in the IoT space. For this project, an ESP32-S3 was selected as the microcontroller of choice. It is easy to program and is a very capable microcontroller for its price. It is also known for its low power consumption and has native USB support, meaning a separate USB-to-UART bridge is not required for it to communicate with a computer over USB. It can also be packaged with an integrated 2.4 GHz Wi-Fi module, Bluetooth Low Energy module, and low-profile RF antenna. These components can be used as part of the Radio Scanner Suite, as well. Since it has integrated Wi-Fi, it can connect to the IoT Cloud subsystem over the internet and send occupancy information to the web application.

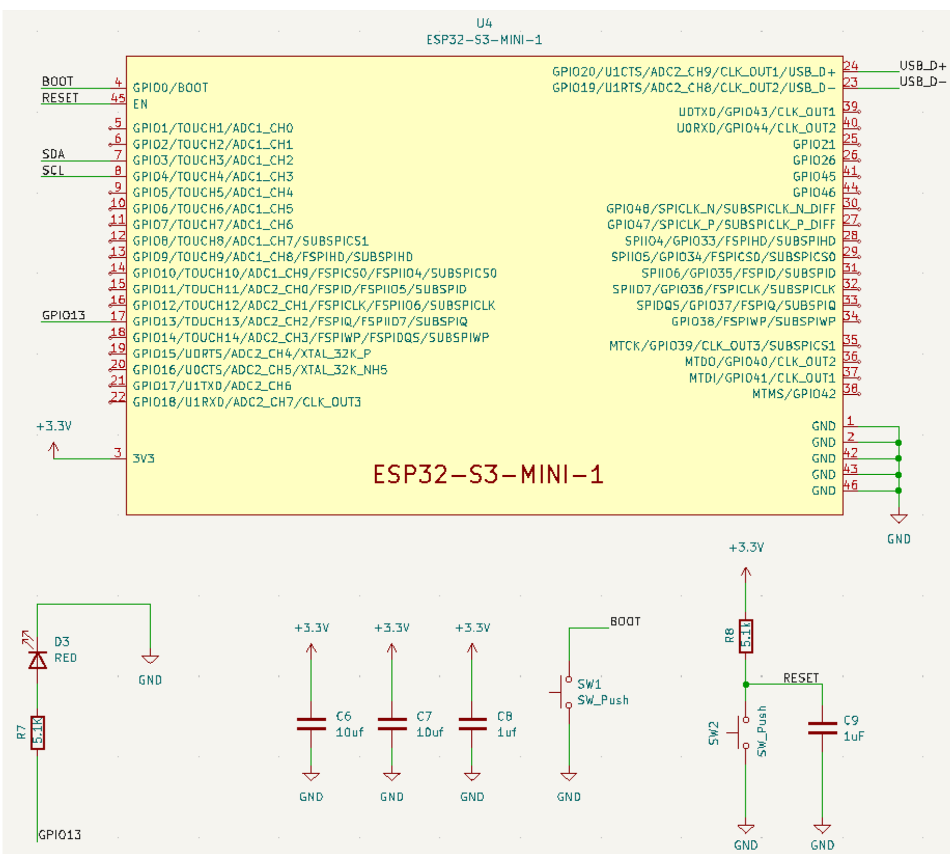

Figure 3: Control Unit Circuit Schematic

Requirements	Verification
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. 1B. Create a simple "Hello World!" program that involves blinking the LED on the microcontroller's GPIO13 pin. 1C. Flash the program onto the microcontroller. Verify that the program is executing correctly and the LED on GPIO13 is blinking.

Table 1: Microcontroller RV Table

2.2.2 Radio Scanner Suite

The Radio Scanner Suite is the main subsystem that will be used to measure the occupancy of a sector. Since most, if not all, modern personal devices contain Wi-Fi and Bluetooth modules, we will be monitoring those protocols 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 and Bluetooth traffic the Radio Scanner Suite picks up, it will send it to the Control Unit for occupancy data processing. Since the ESP32-S3 can be packaged with Wi-Fi and Bluetooth modules and antennas, we will be utilizing these included modules in our Radio Scanner Suite.

2.2.2.1 Integrated ESP32-S3 2.4 GHz Wi-Fi Module

To scan for Wi-Fi traffic, each MOM device will be using the ESP32's integrated 2.4 GHz Wi-Fi module and antenna. The ESP32 can be set to monitor Wi-Fi traffic in promiscuous or monitor mode, and there are official library functions that our monitoring software can utilize as part of our Wi-Fi monitoring algorithm. The Wi-Fi Module and antenna are directly connected to the microcontroller, so no external hardware or connections are required.

Table 12: Wi-Fi RF Standards

Name		Description
Center frequency range of operating channel [1]		2412 ~ 2484 MHz
Wi-Fi wireless standard		IEEE 802.11b/g/n
Data rate	20 MHz	11b: 1, 2, 5.5 and 11 Mbps 11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps 11n: MCS0-7, 72.2 Mbps (Max)
	40 MHz	11n: MCS0-7, 150 Mbps (Max)
Antenna type		PCB antenna, external antenna via the connector [2]

[1] Device should operate in the center frequency range allocated by regional regulatory authorities. Target center frequency range is configurable by software.

[2] For the modules that use external antennas, the output impedance is 50 Ω. For other modules without external antennas, the output impedance is irrelevant.

Figure 4: ESP32-S3 Wi-Fi Specifications as seen in [7]

2.2.2.2 Integrated ESP32-S3 Bluetooth Module

To scan for Bluetooth traffic, each MOM device will be using the ESP32's integrated Bluetooth Low Energy (BLE) module. There are official library functions created by the developers of the ESP32 that assist in the scanning of Bluetooth traffic. Our monitoring software can utilize these as a part of our Bluetooth scanning algorithm. The Bluetooth Low Energy module and antenna are directly connected to the microcontroller, so no external hardware or connections are required.

4.6 Bluetooth LE Radio

Table 18: Bluetooth LE Frequency

Parameter	Min (MHz)	Typ (MHz)	Max (MHz)
Center frequency of operating channel	2402	—	2480

4.6.1 Bluetooth LE RF Transmitter (TX) Specifications

Table 19: Transmitter Characteristics - Bluetooth LE 1 Mbps

Parameter	Description	Min	Typ	Max	Unit
RF transmit power	RF power control range	–25.00	0	20.00	dBm
	Gain control step	—	3.00	—	dB
Carrier frequency offset and drift	Max $\|f_n\|_{n=0,\ 1,\ 2,\ ..k}$	—	2.50	—	kHz
	Max $\|f_0 - f_n\|$	—	2.00	—	kHz
	Max $\|f_n - f_{n-5}\|$	—	1.40	—	kHz
	$\|f_1 - f_0\|$	—	1.00	—	kHz
Modulation characteristics	$\Delta\ f1_{avg}$	—	249.00	—	kHz
	Min $\Delta\ f2_{max}$ (for at least 99.9% of all $\Delta\ f2_{max}$)	—	198.00	—	kHz
	$\Delta\ f2_{avg}/\Delta\ f1_{avg}$	—	0.86	—	—

Figure 5: ESP32-S3 Bluetooth Low Energy Specifications as seen in [7]

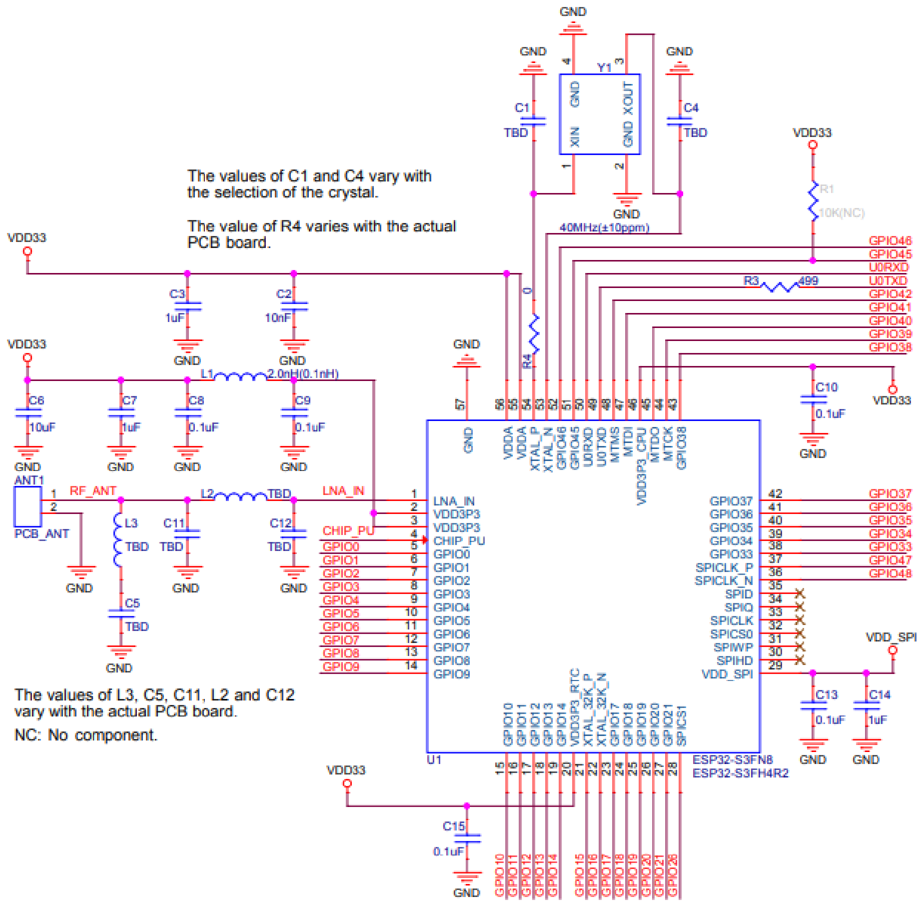

Figure 6: Official ESP32-S3-MINI-1 SoC Circuit Schematic with Wi-Fi and BLE antenna as seen in [7]

2.2.3 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 battery backup, which are used so that a MOM device can continue gathering data even if its wall plug is removed.

2.2.3.1 Lithium-Polymer Battery (Rechargable)

A battery serves as a power source when the device is not plugged into wall power. If the device is plugged in, the battery will be charging to ensure that it has enough charge to power the device if it becomes unplugged. For each MOM device, a Lithium-Polymer battery with an operating voltage range of 3.0V-4.2V will be able to deliver sufficient power. Its compact form also makes it simple to keep components organized when inside of an enclosure. The ESP32-S3 datasheet [7] states that it requires a current of at least 500mA and a voltage of 3.0V-3.6V from the power supply to operate. Using the formula and calculation below, we determined that we need to get a battery with a capacity of at least 500 mAH:

$$Capacity\ [mAH] = I_{VDD}\ [mA] * Runtime\ [hours]$$

$$Capacity = 500\ mA * 1\ hour = 500mAH$$

4.2 Recommended Operating Conditions

Table 8: Recommended Operating Conditions

Symbol	Parameter		Min	Typ	Max	Unit
VDD33	Power supply voltage		3.0	3.3	3.6	V
I_{VDD}	Current delivered by external power supply		0.5	—	—	A
T_A	Operating ambient temperature	85 °C version	–40	—	85	°C
		105 °C version			105	

Figure 7: ESP32-S3 Recommended Operating Conditions as seen in [7]

2.2.3.2 Battery Charger

All rechargeable batteries will need to be recharged after being used. When the device is connected to wall power, the battery charger will take some of the wall power and use it to charge the battery. Our battery charging circuit also consists of a battery fuel gauge, which will send battery status information to the microcontroller over I^2C to notify us if the charge on a battery backup is running low. Other safeguards are added to ensure that the Lithium-Polymer battery does not get overcharged.

Requirements	Verification
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. 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. 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. 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.
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. 2B. While the program is running, measure the battery voltage by probing it using a multimeter or oscilloscope. 2C. Ensure that the value printed by the microcontroller is within 80% of the voltage read by the oscilloscope or multimeter.

Table 2: Battery Charger RV Table

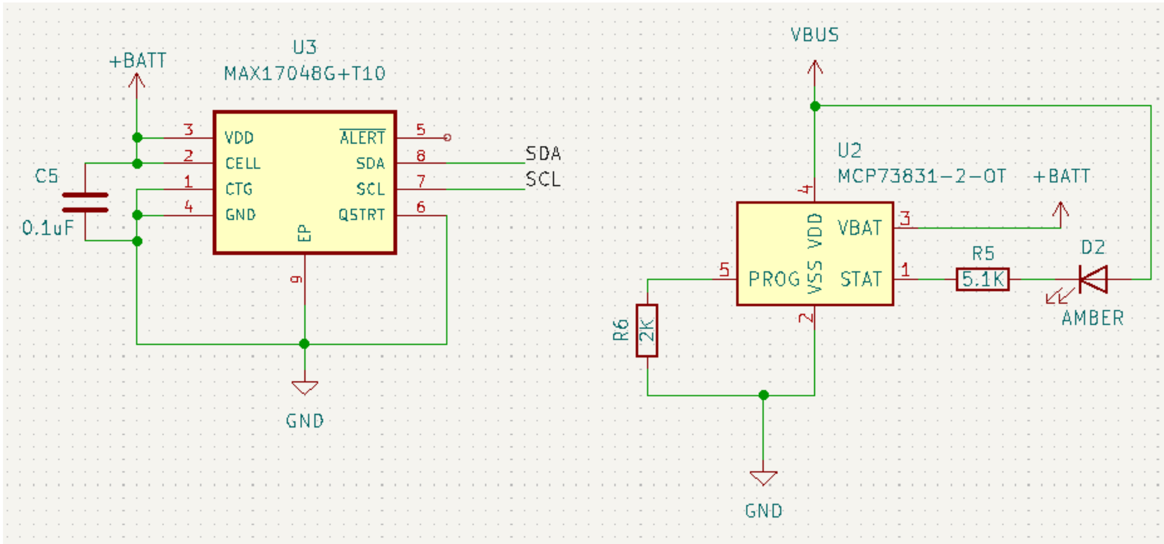

Figure 8: Circuit schematic of the Battery Charger

2.2.3.3 Voltage Regulator

Since the ESP32 has a standard operating voltage of 3.3V [7], and Lithium-Polymer batteries have a normal voltage of 3.7V, the voltage will need to be stepped-down to prevent damage to the ESP32. The Voltage Regulator is combined with a battery backup switchover circuit to allow the device to continue running when the wall power is disconnected.

Requirements	Verification
1. The voltage regulator must be able to convert from wall power and supply a stable voltage of 3.3V ± 0.3V.	1A. Connect the device to wall power and boot it. 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. 1C. Confirm that the voltage read is 3.3V ± 0.3V.
2. The voltage regulator must be able to convert from 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. 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. 2C. Confirm that the voltage read is 3.3V ± 0.3V.

Table 3: Voltage Regulator RV Table

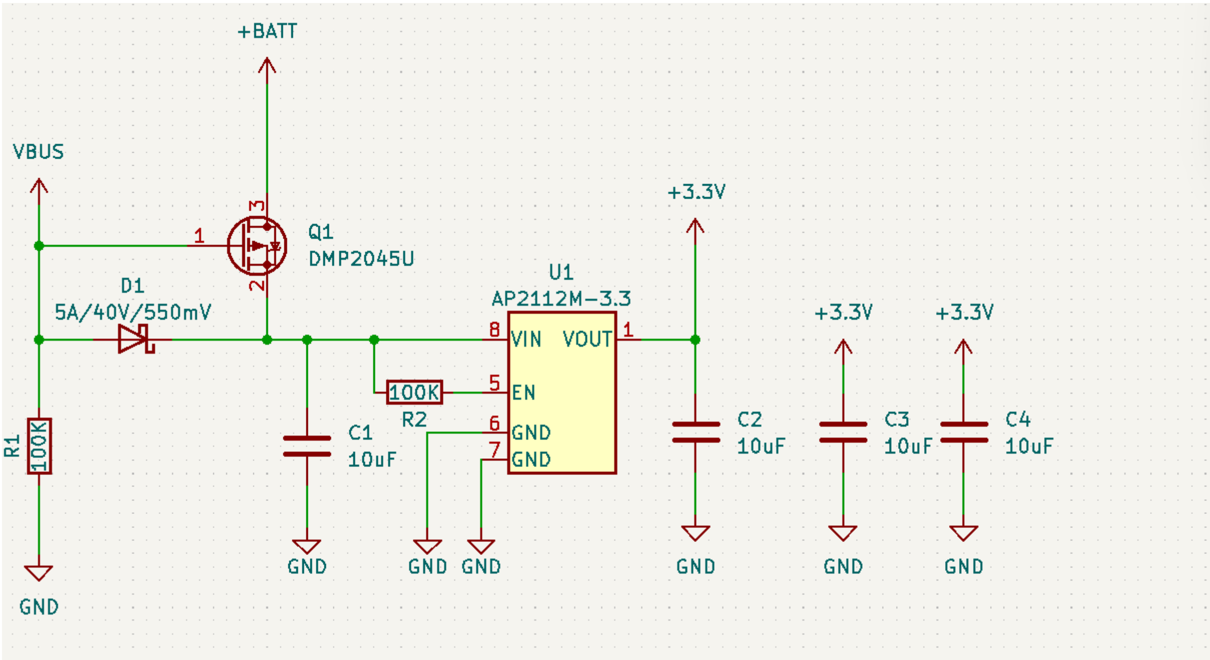

Figure 9: Circuit schematic of the Voltage Regulator with a battery backup switchover circuit

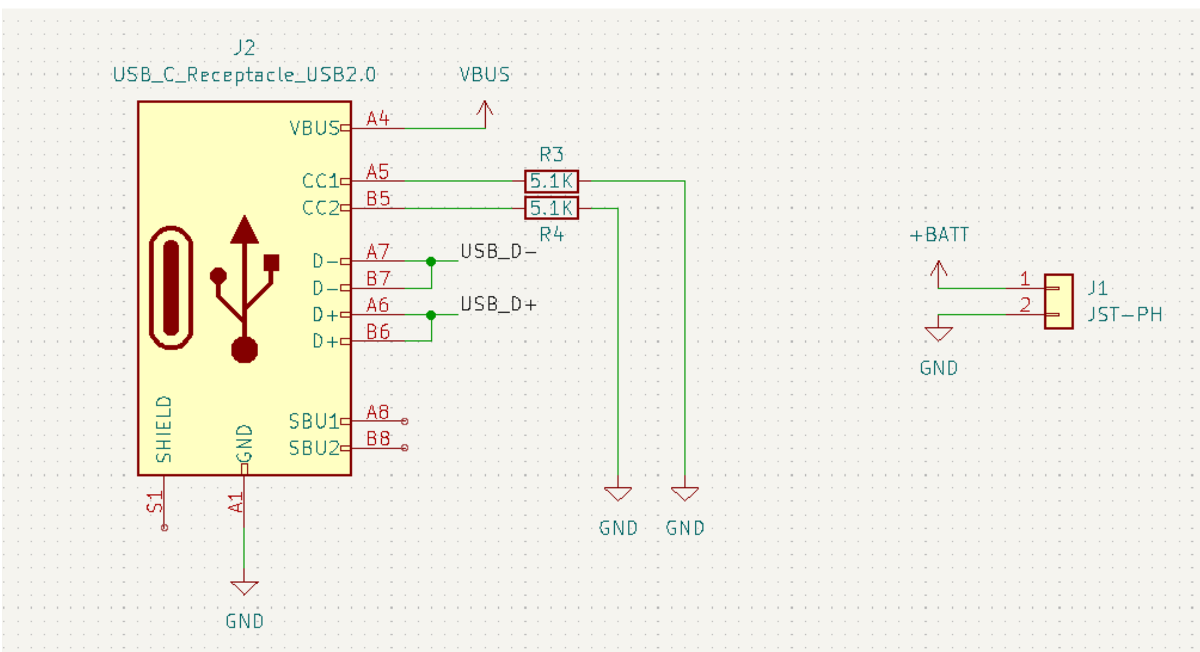

Figure 10: Circuit schematic of the device's USB receptacle and battery header

2.2.4 IoT Cloud

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

2.2.4.1 AWS MQTT Service

Each one of our MOM devices will connect to AWS IoT Core's MQTT Service. The MQTT Service allows the MOM devices to publish messages, such as occupancy information events, and data to the various other AWS services, such as the database, that will be used for the web application.

Requirements	Verification
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. 1B. Connect the microcontroller to the development computer and flash the program to the microcontroller. 1C. Run the ping program and confirm that the round-trip latency is no greater than 500 milliseconds.
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. 2B. Connect the microcontroller to the development computer and flash the program to the microcontroller. 2C. Set up the MQTT service to process occupancy information events. 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.

Table 4: AWS MQTT Service RV Table

2.2.4.2 AWS Database

To store and persist the occupancy data for the web application to use, we will need to keep the data in a database. AWS makes this easy by offering a plethora of database management systems that are interconnected with other AWS services, such as the AWS IoT MQTT Service. We plan on keeping stale occupancy data for a particular client device for no more than 24 hours. The web application can directly query this database to retrieve the most recent occupancy data transmitted by each MOM device.

2.2.4.3 Web Application

The web application is what the users of our project will be interacting with. The frontend of the web application will display the occupancy data, while the backend will query the AWS database for the occupancy data to display to the user. It will need to display the information quickly to the user, and its user interface will be scalable and readable on many types of devices.

Requirements	Verification
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. 1B. Enter the address of the web application and start a stopwatch. 1C. Confirm that the web application's user interface loads in less than 10 seconds.
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. 2B. Enter the address of the web application. 2C. Confirm that the user interface is readable and properly scaled to the size of the smartphone. 2D. Open a new tab on a laptop that has a strong connection to the internet. 2E. Enter the address of the web application. 2F. Confirm that the user interface is readable and properly scaled to the size of the laptop's display.

Table 5: Web Application RV Table

2.3 Tolerance Analysis

The most challenging part of our design is the process of estimating the occupancy of a sector via the Radio Scanner Suite. As Wi-Fi and Bluetooth are wireless communication protocols, it is possible that a MOM device that is assigned to one sector of a building could receive traffic that actually originated from a different sector. One example of this is in a populated, multi-level apartment complex. Not only would a device in one unit be able to detect Wi-Fi traffic that is in a different unit on the same floor, but it can also detect traffic originating from clients on different floors. Our occupancy estimation algorithm would need to account for this in order to keep estimations as accurate as possible. One possible metric that could be used is the Received Signal Strength Indicator (RSSI), which can be read from the integrated antenna. We can estimate RSSI and its relationship to distance as described in [8] with the following formula:

$$RSSI = A - 10n \log(\frac{d}{d_0}) - X_\sigma$$

In this formula, A is equal to the RSSI measured at a reference distance $d_0 = 1\ m$, n is the path loss exponent of the environment where the device is deployed, d_0 is the reference distance, and X_σ is the Gaussian distribution cover factor. If we take a reference distance of $d_0 = 1\ m$ and take multiple measurements for a single client, we can use the following formula to estimate distance:

$$d = 10^{(A-RSSI)\ /\ 10n}$$

As we can physically measure the maximum distance between a client and one of our devices in a sector, we can determine a minimum RSSI value which indicates if a detected wireless packet originated from the device's assigned sector. For example, say that we assigned a MOM device to a 10 meter x 10 meter room and placed it on the ground in the middle of the room with no obstructions. If we place a client device against a wall (such that it is 5 meters away from the MOM device) and there is some form of line-of-sight between the MOM device and client device, the path loss exponent will be about $n = 1.8$. If we measured the reference RSSI at $d_0 = 1\ m$ to be $A = \quad 25\ dBm$, then we would find that the minimum RSSI is about -38 dBm. Any traffic received that has RSSI readings lower than -38 dBm can be filtered out as they are likely not from the same room.

2.4 Cost Analysis

The average salary of a computer engineering graduate from UIUC is **$105,352** [12]. 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. Using the below formula, we can calculate the labor cost per team member:

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

So, according to the formula given in the rubric, 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.

In addition to labor costs, parts would need to be purchased to build each MOM device. The cost of parts for each device would be reduced if we were to buy these components in bulk. However, since we are just making a single prototype for now, the cost of the parts are listed as per their MSRP for a single unit of each part. The below table contains the bill of materials for a single MOM device combined with the team labor cost.

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	6	$2.16
Surface Mount Capacitor 1uF	2	$0.72
Through-Hole Red LED	1	$0.35
Through-Hole Yellow LED	1	$0.15
USB wall adapter	1	$5.00
USB-C to USB-A cable	1	$5.89
PCB	1	$5.00
Total Parts Cost		$33.53
Labor (Computer Engineer)	3	$49,383.00
Parts + Labor		$49,416.53

Table 6: Cost Breakdown Table

2.5 Tentative Schedule

Week	Vish	Franklin	John
9/19	Finalize parts research and create bill of materials	Finalize parts research and create bill of materials	Finalize parts research and create bill of materials
9/26	Order parts for the first prototype	Order parts for the first prototype	Order parts for the first prototype
10/3	Create a simple Bluetooth monitoring program that can count the number of unique Bluetooth MAC addresses	Set up AWS services and environment	Create Wi-Fi monitoring program that can count the number of unique Wi-Fi MAC addresses
10/10	Work on getting AWS IoT Cloud set up with the MQTT protocol	Work on the radio scanner components	Work on power supply setup with LiPo battery
10/17	Write API script that will receive pings from cloud when database updates	Hardware unit testing and verification	Validate battery output values and discharge rates of it
10/24	Backend web app development	Backend web app development	Frontend web app development
10/31	Resolve any issues with microcontroller accessing database and web app	Resolve any issues with microcontroller accessing database and web app	Resolve any issues with microcontroller accessing database and web app
11/7	Mock testing of subsystems	Mock testing of subsystems	Mock testing of subsystems
11/14	Finalize assembly and test for any bugs	Finalize assembly and test for any bugs	Finalize assembly and test for any bugs
11/21	Fall Break	Fall Break	Fall Break
11/28	Demo	Demo	Demo
12/5	Final Presentation	Final Presentation	Final Presentation

Table 7: Tentative Schedule of Project Development

3 Ethics and Safety

One ethical concern we foresee is installing or deploying our project in public spaces (e.g. ECEB, Grainger Library) without people being aware that the device is in operation and "tracking" people situated in the public space. This might violate section 1.6 in ACM's Code of Ethics [9]. We plan to address this concern by explicitly labeling our devices (with the name of our project, what class and section this device was built for, points of contact, etc.). This gives people who have physically seen our devices to know the purpose of the device as well as who to contact if there are any concerns. We also don't want to give off the impression that we are recording people's movements without their knowledge and consequently violating their personal privacy. Thus, we may also introduce an online survey that individuals can fill out anonymously to report any questions or concerns they may have, as well as post signs notifying people of our project.

In addition, while collecting our occupancy data, we plan on collecting no more than the signal strength and originating MAC addresses of transmitted traffic our microcontrollers detect. This is in accordance with section 1.6 in the ACM Code of Ethics [9] to respect privacy by only collecting the minimum amount of personal information necessary to perform the project's function. As we don't need to store MAC address information to estimate occupancy, this information will be retained for no more than 24 hours after it has become stale and will be safely disposed of after that time.

As our project is currently in its very early stages of development, it is imperative that we stay in accordance with section I-5 of the IEEE Code of Ethics [10] and acknowledge and consider all feedback offered to us. This includes feedback offered by other students, teaching assistants, professors, Machine Shop staff, and other professionals in our discipline. As we continue through the development of this project, it is also important to remain honest and realistic in presenting our project's results, capabilities, limitations, and data.

We will make sure to adhere to all the safety regulations and precautions put in place when building the electrical components/portions of our project. That means strictly following the rules and precautions enforced in all of the ECE labs's (ECE Open Lab, etc.) and outlined in the university's laboratory safety training. For instance, always wearing safety goggles and face coverings when soldering or performing other sorts of direct electrical work.

As our design includes the use of a rechargeable battery, we must also test and ensure that the power supply subsystem remains within safe operating parameters. This includes making sure that the rechargeable battery does not overcharge, which could lead to battery explosion [11] and physical harm to others. This directly relates to section I-1 of the IEEE Code of Ethics [10] to protect the health of the public. To prevent this and possible physical harm to others from happening, we will ensure that our battery charger contains a battery charge management controller to properly manage the charge of the battery.

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] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, "A Location Estimation Algorithm Based on RSSI Vector Similarity Degree," in International Journal of Distributed Sensor Networks, vol.10, issue 8, 2014, doi: 10.1155/2014/371350.

[9] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," Association for Computing Machinery, 2018. Accessed: Sep. 14, 2022. [Online]. Available: https://www.acm.org/code-of-ethics

[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] National University of Singapore College of Design and Engineering, "Why do Lithium-Ion batteries explode?," National University of Singapore. Accessed: Sep. 14, 2022. [Online]. Available: https://cde.nus.edu.sg/cert/2018/06/22/why-batteries-explodes/

[12] 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