Event Attendance Tracker

Team 13

Anand Sunderrajan (anands3), Eric Layne (ealayne2), Mason Edwards (masonae2)

ECE 445 Design Document - Fall 2020

TA: Dhruv Mathur (dmathur2)

Table of Contents

1. Introduction

1.1. Problem and Solution Overview

Every year, there are hundreds of trade shows, career fairs, expos, and other large events wherein there are hundreds of presenters (i.e. exhibition booths) and thousands of attendees. Attendees see many booths, try to talk with and leave an impression on presenters who may interact with hundreds of attendees, hopefully remember information about each booth, and create a significant amount of waste from brochures and pamphlets that are thrown out shortly after the event. To solve all these issues, the Event Attendance Tracker will provide a way for attendees to recall booths they stopped at, and learn more information about them even after the event, including contact information to communicate with the presenters in a more memorable setting - all without needing to remember the name of every booth they stopped at.

In the United States, events such as Maker Faires and CES are prime examples of where such a solution could be extremely beneficial. CES 2019 attracted 175,212 verified attendees [1] compared and 4,500 exhibitors [2], while Maker Faire 2017 had approximately 125,000 attendees to its 1,200 exhibitors [3]. One can even look at an even smaller scale, specifically at the University of Illinois, and notice the presence of such events through the Career Fair, Quad Day, and Engineering Open House. The issues our solution addresses can be found at all these events.

Our solution proposes a standalone, battery powered device which will continuously monitor for booth attendants via Bluetooth Low Energy. A companion app will monitor for these devices located at each booth; upon detecting a booth, the app and device will sync and the app can provide information about that booth. This information could be a website link, contact information, an introduction to the booth itself, or whatever the exhibitor wishes to provide. The smartphone app would choose to log this data based on the attendant's relative distance to the device itself and the amount of time they were near the booth. Additionally, the app can include functionality to share information about the attendee to the booth presenter, which could include contact information, links to personal websites, or LinkedIn pages. As the event progresses, the device at each booth would display the number of attendees who were classified as visited for each booth, and the battery charge on a small display. This would serve to show the presenter their attendance count and indicate when the device needs to be charged. The attendant count could be very useful for recruiters as it would allow them to consistently measure attendance without intrusive methods such as sign-up sheets or less reliable methods like manual counting.

Current alternative solutions include apps like Scantrakk, which works on the premise that an attendee would have enough time and space to walk up to the booth itself and scan a QR code that would be at the booth [4]. This would register them as attended on the app. While the app presents an alternative solution, it is based on a subscription model, which leads to an extremely high recurring cost. To be able to have 5000 (+500 bonus) scans, one would have to pay $549, while opting for the unlimited scan package would lead to a recurring charge of $399/month. Additionally, solutions that require scanning a QR code could suffer from problems with attendees crowding around the code. Another method that is commonly used is WiFi monitors in conjunction with heatmaps. The WiFi monitors pick up attendee movement and the information is graphically depicted via heatmaps, however, this leads to the issue that the information provided is for general areas, rather than specific booths. Furthermore, it solely provides estimated metrics of attendance in the general area to the booth presenters, and does not provide information to attendees.

1.2. Visual Aid

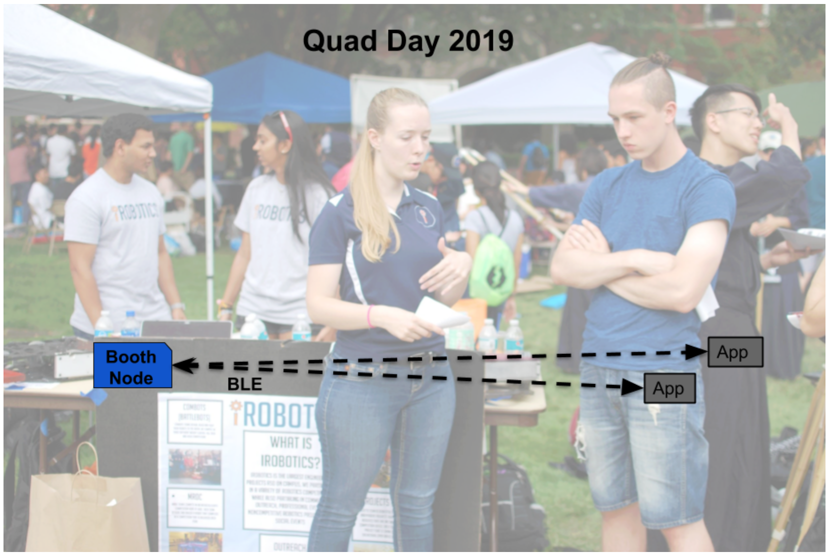

Figure 1: Event Attendance Tracker Usage Visual Aid

Figure 1 shows the operation of the device. It will be self contained and be placed at each booth and a smartphone app that attendees download will exchange information with it via BLE.

1.3. High-Level Requirements

- The device's battery will need to be able to provide sufficient power for at least four hours of normal operation per charge.
- The interface between the ESP32 and the Smartphone itself must be able to log attendees within 3 meters [5] from the booth they are at provided that they meet the minimum time requirement for the booth, which would be user configurable from 1-15 minutes. Additionally, they must be at least 6 meters [5] away from the device present at another booth.
- The transfer of user data between the smartphone and the device should be completed in under 250 milliseconds after initial connection. This is set to ensure the system of the device and the smartphone app of all attendees can be serviced, and would allow the typical 4 attendees actively interacting with a booth presenter [6] to be serviced each second, in addition to servicing other smartphones' asynchronous communication once their timer expires for having "attended" the booth.

2. Design

2.1. Block Diagram

This system was designed to meet the high-level requirements. Specifically, we have designed the structure of the power supply subsystem and power switch to ensure we can safely use a battery cell. The system will not have any high-current components and so we can select from a multitude of inexpensive and readily available batteries designed for low-power commercial electronics. The control unit subsystem is designed around a microcontroller which is capable of wireless BLE communication to satisfy the requirement to detect which booth a smartphone app is nearest, and we have experience with it and are confident in its ability to operate fast enough to handle exchanging data with attendees' smartphones in a timely manner.

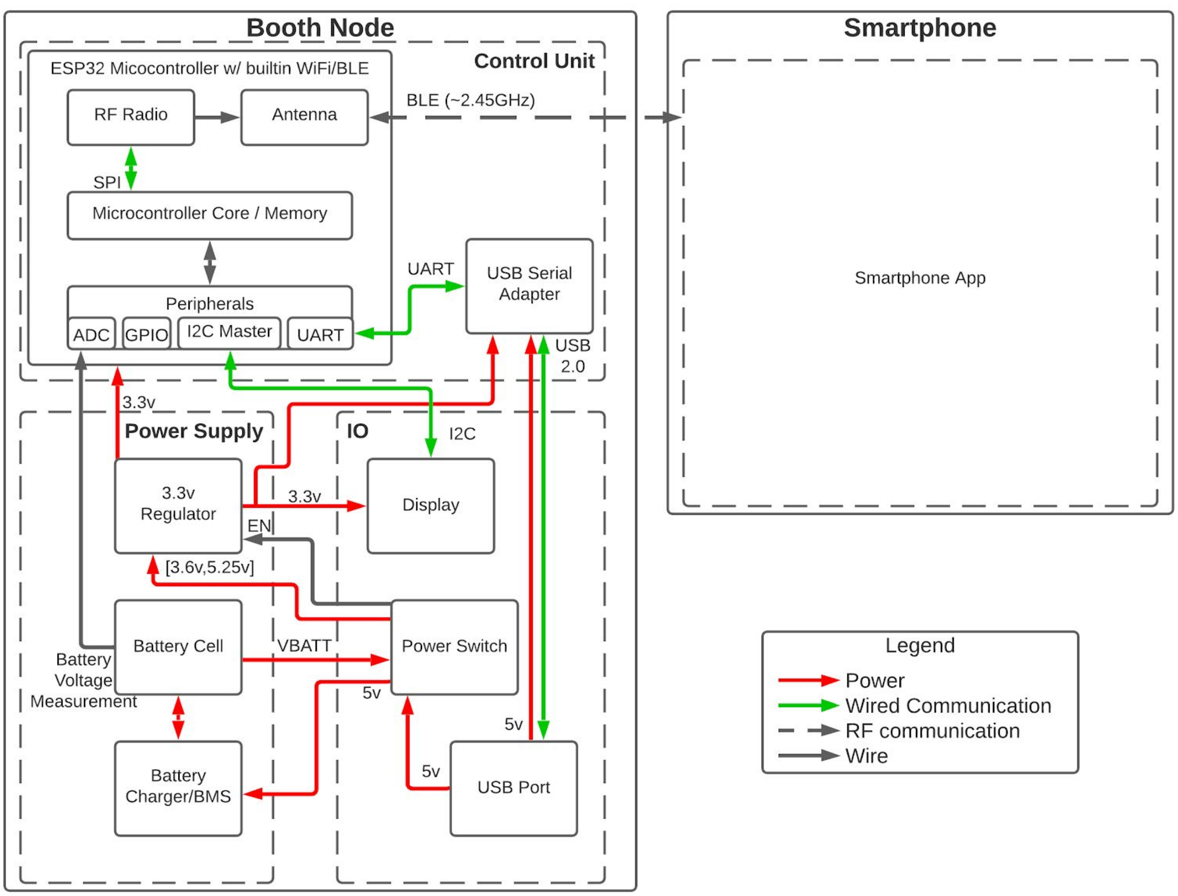

Figure 2: Block diagram for Event Attendance Tracker System

2.2. Physical Design

The physical design is driven by the dimensions of the chosen battery cell as well as estimates to the PCB dimensions based on a preliminary layout. The battery is kept separated from the PCB and other electronics in a space in the bottom of the design, with the PCB held above it, far enough to allow the design to include a separator to ensure no connections on the PCB short circuits on the case of the battery cell. On the left side of the physical design, there are two cutouts, one for the power switch of the device, and the other for a USB mini B port to allow charging the device and interacting with the microcontroller. The top has an angled front which allows the OLED display to be easily read by a booth presenter standing near the device.

Beyond the physical design, the device interacts with smartphones over BLE, the radio for which is contained at the rear of the design to reduce the interference and metal blocking the RF signals (avoiding for example, the metal in the OLED display and its high frequency communication).

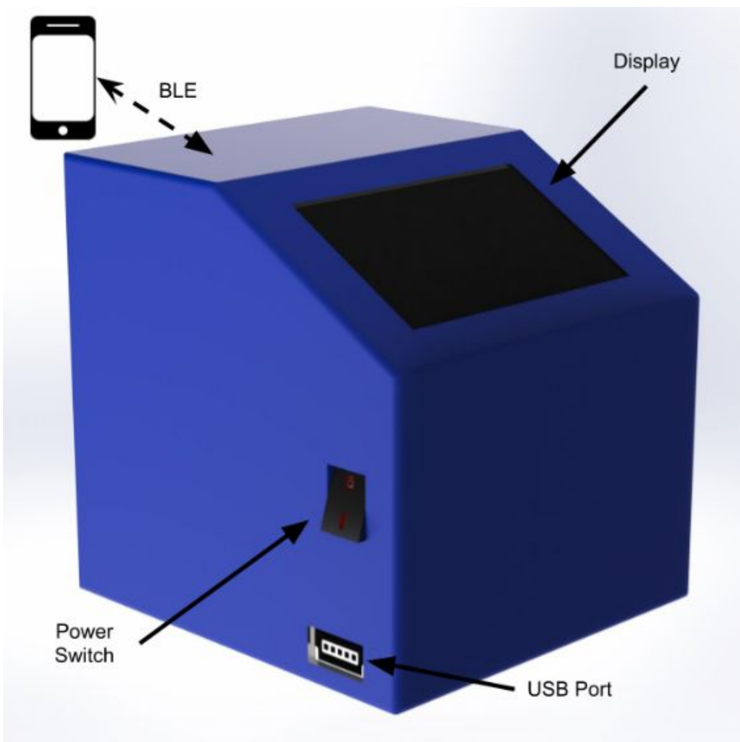

Figure 3: Event Attendance Tracker Physical Design Render

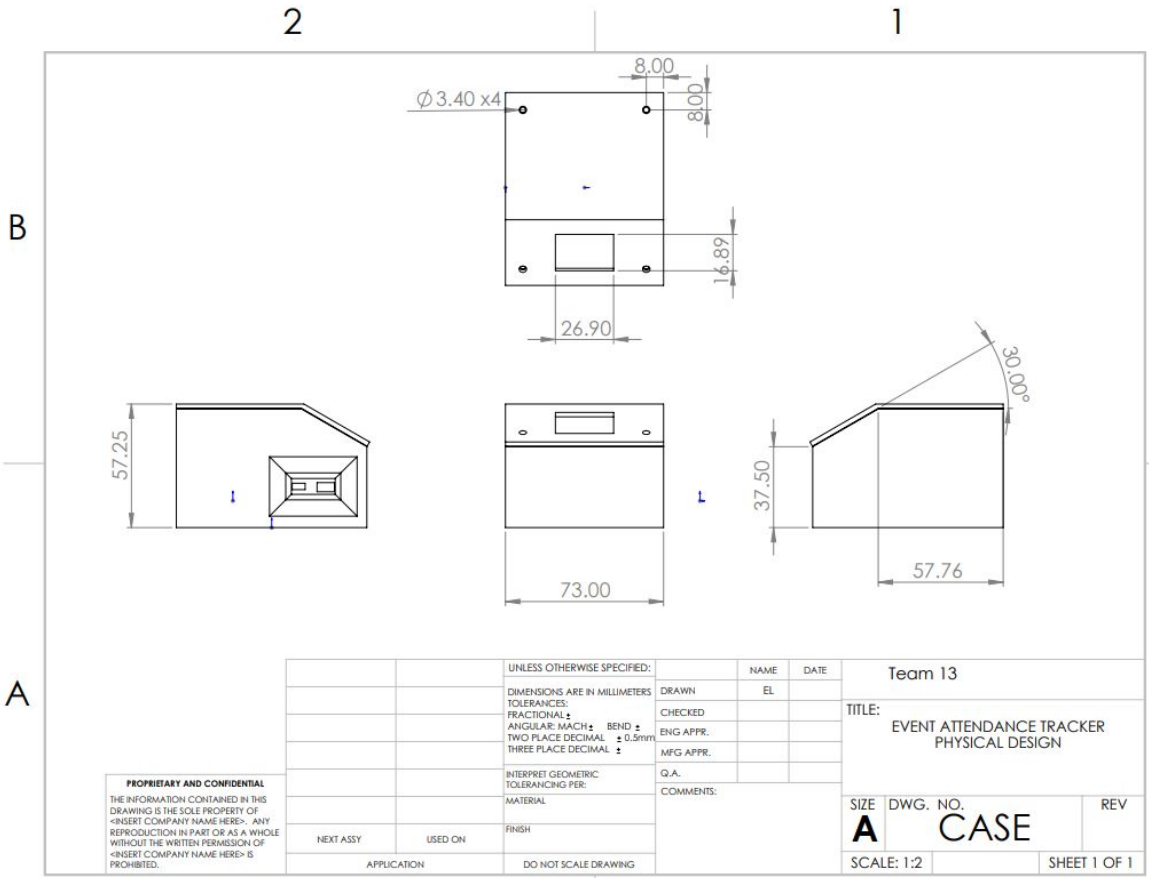

Figure 4: Event Attendance Tracker Physical Design Dimensions

2.3. Power Supply

The power supply subsystem manages taking in a range of input voltages; when enabled it regulates the voltage to 3.3V to power a majority of the device's components including the microcontroller, USB serial adapter, and display. Further, this subsystem monitors and safely charges the lithium-polymer battery cell. This battery voltage is exposed to the other subsystems for additional monitoring and allowing control of where power is delivered based on the device's current operating mode.

2.3.1. 3.3V Regulator

The device could be expected to draw up to ~200mA when powered (microcontroller: 80mA [7], display: 108mA [8], USB serial adapter: 13.7mA [9]) from the regulated 3.3V supply, and based on the recommended current supply for the microcontroller of 500mA to power during bursts of current draw, the regulator must be able to support at least this much current (with additional overhead to ensure reliability) and deliver it directly to the control unit and I/O subsystems. To leave some overhead room, we will require the regulator to support up to 750mA continuous of regulated voltage. The power supplied to the regulator will be delivered from the power switch which will control if the device is powered on or is inactive.

The regulator also will have an enable signal, which will disable the device if the battery voltage drops too low. Without this, the device could "brown-out" and crash from insufficient voltage/current supply, specifically when the supply voltage drops below 3.0V [6]; in an even worse case this could otherwise cause the battery cell to be over-discharged and become a safety risk. To ensure this block stops current from flowing from the battery when the battery voltage is too low, the enable pin will be connected through a voltage divider to the input voltage to the regulator. With the ADP7156 regulator chosen, we can implement this functionality by connecting the enable pin to the regulator input voltage through a voltage divider. These divider calculations can be seen below with equations and device characteristics from the regulator datasheet [10].

$$R_{up} = R_{down} * (V_{en} - 1.22V) / 1.22V \quad (1)$$

With a $V_{batt-min} = 3.6V$, we can use approximate resistances of $R_{up} = 100K$ & $R_{down} = 47K$ (Equivalent to a $V_{en} = 3.816V$ with a $V_{hysteresis} = 0.281V$)

Which means the device will shut down typically at $V_{batt-typ} = 3.535V$, but potentially as high as $V_{batt-max} = 3.816V$.

This divider may need to be slightly adjusted during assembly to trim the cut-off voltage to be reliably close to the ideal 3.6V (since the device manufacturing variation itself we cannot account for, only trim to fix in the final product).

2.3.2. Battery Cell

The device will use a single cell lithium-polymer battery as the battery, which will dictate the charge parameters and limits, as well as the voltage range that the subsystem must be able to regulate while powering the rest of the device. This battery will provide power to the device when it is switched on and not connected to a USB power source. This battery cell is connected directly to the I/O subsystem to control the mode of operation of the device, and it is connected to the battery charger/BMS block in this subsystem for both charging and monitoring. The battery voltage is also broken out to the control unit subsystem for estimating the battery capacity remaining.

To fulfill the 4 hour runtime, we can calculate the minimum battery capacity to meet the requirement. We will calculate the capacity required for both the average expected current, as the peak current is rarely seen [11] (and is true from prior experience with the device).

$$Capacity[Ah] = Current[A] * Runtime\,[hours] \quad (2)$$

$$Capacity[Ah] = 0.2A * 4h = 0.8Ah \quad (3)$$

From this minimum capacity of 0.8Ah, we will choose a battery cell which has at least this much capacity (and add overhead room to ensure the capacity is sufficient to cover the initial variable bursts of current draw during the booting process).

We have chosen a single cell lithium-polymer battery cell with a capacity of 1.5Ah; this will certainly meet our minimum criteria for the battery life. The extra capacity will increase the battery run-time, and allow the device to operate with less intervention by the booth presenters. The cell is slightly larger than the minimum requirement, but is more easily available and still fits within a reasonable footprint for the device itself.

2.3.3. Battery Charger/BMS

This block must safely charge the battery cell and monitor it to ensure it is not overcharged to an unsafe level [12]. This is required to safely use a lithium-based battery cell because it otherwise could degrade the battery cell or even cause it to become inoperable (or worst case, lead to thermal runaway) [12]. This connects to the power switch within the I/O subsystem when the device is not active (i.e. in charge mode) and is always connected to the battery cell for monitoring and charging.

We have chosen to use an MCP73831T, which provides the functionality of charging the battery cell, and is specifically designed to charge lithium polymer batteries, which will ensure it operates safely. The IC has built in hardware for monitoring the charging process, and will "trickle-charge" the battery cell once it is fully charged, to ensure it stays fully charged without overcharging the cell. The "PROG" pin of the IC determines the maximum charging current, and the current is given by the following expression [13]:

$$I_{reg} = 1000V / R_{prog} \quad (4)$$

In this equation, I_{reg} has units of milliamperes, R_{prog} has units of kilo-ohms. With a minimum capacity of 0.84Ah and the requirement to have a maximum charge rate of 0.5C, we can set the charge current to be up to 0.42A. Solving the equation for R_{prog} we find that $R_{prog} \geq 2.4k\Omega$, and with resistor value tolerances of $\pm 5\%$ the value resistor we must use is at least $2.5k\Omega$. Rounded up to standard resistor values, we chose to use a $2.7k\Omega$ to ensure we do not exceed the charge current limit for any battery chosen that meets the minimum capacity criteria.

Power Supply	
Requirements	**Verifications**
The subsystem must be capable of outputting a regulated 3.3V ± 0.1V at 750mA.	Provide the subsystem with 4.20V from a bench power supply, and connect the regulated output to an resistive load pulling 750mA, and at the same time measure the steady state output voltage using an oscilloscope.
The subsystem must be able to operate normally (fulfilling all other requirements) when the power in voltage is anywhere in the range (3.6V ~ 5.25V)[12][14].	Repeat the maximum load test to confirm the subsystem can output a regulated voltage at 750mA, but instead slowly sweep the input voltage from 3.6V up to 5.25V using the same bench power supply. Record the output voltage after allowing time to stabilize throughout the sweep.
The subsystem, provided a steady USB voltage (Hub voltage: 5V ± 0.25V [14]), has to charge the battery cell at a rate of no more than 0.5C, and without the voltage rising above the max voltage of 4.2V ± 0.1V [12] anytime during or after the charging (even if left plugged into USB power).	Connect a bench power supply providing 5.0V to the subsystem, and monitor the current drawn from/displayed on the power supply, as well as the voltage of the battery using an oscilloscope. Both the current and voltage requirements must be met, while the battery is charging, and after it is "fully charged;" record pass or fail of the test.

Table 1: Power Supply Subsystem Requirements and Verifications

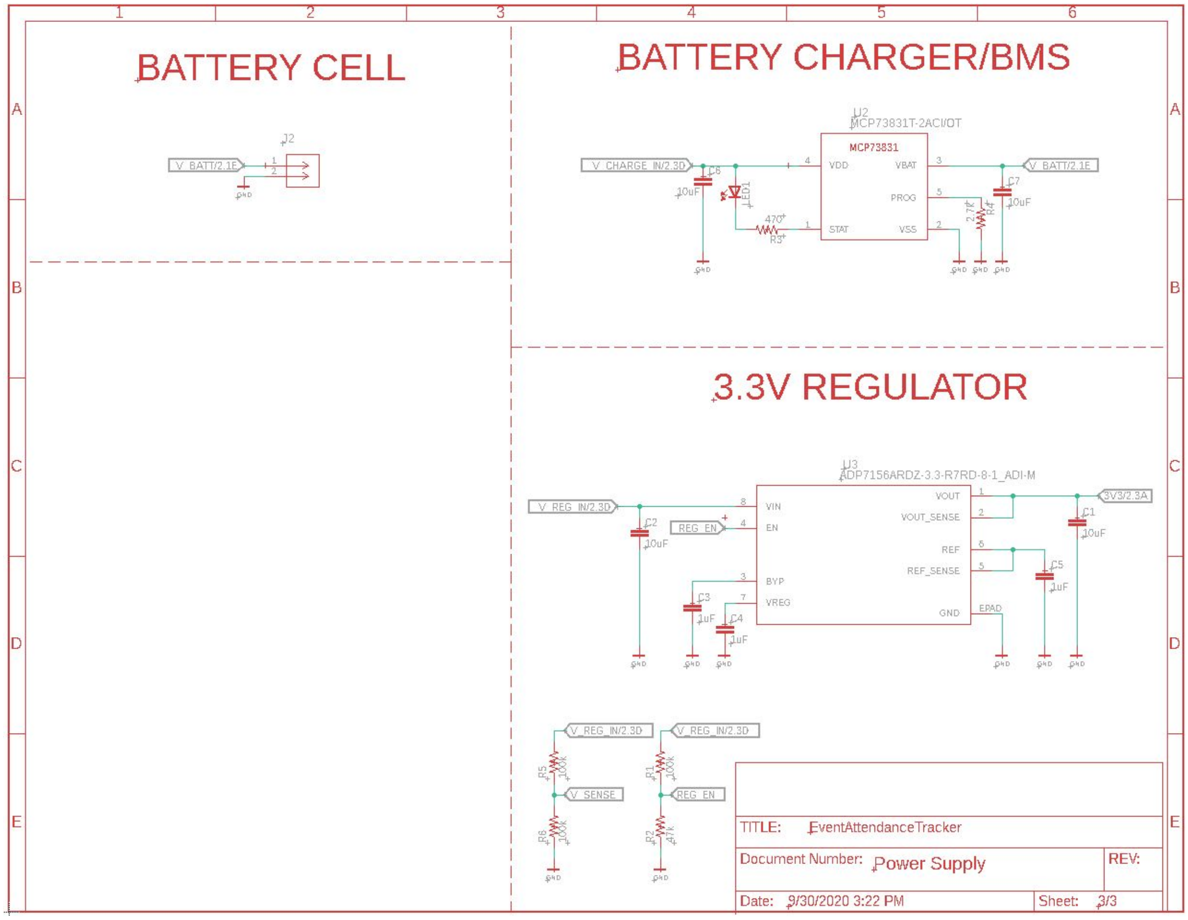

Figure 5: Power Supply Schematic

2.4. I/O

The I/O subsystem will provide the interfaces for the user to control the device and monitor the status of the device-smartphone app system. There are three main aspects of the I/O subsystem: the OLED display, DPDT power switch, and Mini B Female USB port.

2.4.1. Display

The display must fulfill two features, showing the total number of attendees for the devices' booth as well as display the estimated battery capacity remaining. It will be controlled by the microcontroller over an I2C bus in order to configure the display and show the desired information. To operate, the display will directly draw power from the regulated 3.3V output of the power supply subsystem.

We chose an 0.96" OLED display (non-RGB) to perform this block's task, as it will provide sufficient space for displaying the information, OLED displays have ample contrast for a variety of lighting environments, and have much better viewing angles compared to other display technologies like LCDs. The tradeoff is that OLED displays can draw more power when all pixels are brightest [8], but have the added benefit of saving energy when a large portion of the screen is black (as there is no backlight to constantly power). For simply conveying basic information about the attendees and battery capacity, the non-RGB display will be sufficient and is also less expensive (adding to the value of the device compared to other commercial options that provides similar information to the booth presenters [4]).

2.4.2. Power Switch

The power switch will control when the power supply subsystem outputs the regulated 3.3V supply and when the battery charger/BMS receives power to charge the battery cell while ensuring that the power supply subsystem is provided the correct conditions to safely charge the battery cell. To do this, it will be connected to the USB port as well as the battery cell and route the power to either of the two power inputs of the power supply subsystem and the power supply enable input depending on the switch position. Finally, the power switch will ensure the device uses the USB power when available, and only connects the battery cell to the power supply voltage input when no other power is available, but the power switch is turned on. This was decided to improve the lifetime of the battery cell by only drawing power from it when necessary.

To implement this block, we decided to use a double pole double throw (DPDT) switch, which will allow enough control over what connections are made in each switch position. Specifically, we need to ensure the battery is not charged at the same time as the device is active, to ensure the device does not draw excessive current. Further the switch chosen must be able to support the peak current of 750mA of DC current (and break the connection when switched).

Other than the switch itself, this block contains circuitry to source power from the USB port when available, and only draw current from the internal battery when the USB port is not powered. This was implemented by a P-channel MOSFET to switch the current from the battery on and off, and uses a schottky diode for its low forward voltage drop to prevent the battery from providing current to the USB port which could otherwise damage the power source the USB port is plugged into.

2.4.3. USB Port

The USB port will provide a way to charge the device's internal battery, power the device, as well as allow for serial communication between a host computer and the device. This USB port 5V line will be connected through the power switch block within the I/O subsystem before routing it to the various components of the device. The control unit is connected over the USB differential data pair to the USB port, enabling the device and a host computer to communicate. This communication will allow the device to export the list of anonymous IDs of booth attendees, as well as statistics about how many people attended the booth and for how long.

We chose a USB Mini B Female port to interact with the device for a few reasons. First of which is that the connector is robust, compatible cables are fairly common, and it is capable of providing the sufficient constant current. Additionally, it is also easily hand soldered (in a production environment this could be switched to a USB Micro B or USB C port, though it would not affect the performance drastically).

I/O	
Requirements	**Verifications**
The display's I2C bus must be functioning properly and be able to turn on and off every pixel of the display.	Connect an Arduino (or similar hardware), capable of operating as a mast of an I2C bus, to the display and verify the device is seen as a slave at the correct 7-bit address. Then individually turn each pixel on and off in sequence, verifying that each lights up as expected.
The power switch must not allow more than 0mA ± 0.1mA to flow out to the battery charger/BMS when the device is turned on (which is the "active" mode). This is set to draw no more current than the self-discharge current of the lithium-polymer cell [15].	Place a DMM in series between the switch and battery charger/BMS, and record the maximum leakage current after reaching steady state (or after 15 seconds during active operation).

When USB port power is available, the power switch must ensure no more than 0mA ± 0.1mA is allowed to flow in/out from the power supply battery voltage output. This is set to draw no more current than the self-discharge current of the lithium-polymer cell [15].	Place a DMM in series between the battery cell's output from the power supply subsystem and the input to the power switch. Record the maximum current after reaching steady state (or after 15 seconds).
USB Port must be able to provide sufficient power to charge the battery cell (current up to 0.5C [12]) or provide the constant current the rest of the device consumes when all subsystems are operating as required (at least 200mA continuous). The design must not cause the temperature of any component to exceed 85C [7][16].	Connect the USB port to a standard USB power supply, and measure the current passing through the USB port using a DMM while drawing current to charge the battery at 400mA (0.5C of minimum battery of 800mAh). Verify the drawn current is correct, and confirm the port is capable of providing the requested current for 15 minutes, recording the maximum temperature of the board using a thermistor probe, verifying it does not exceed 85C.

Table 2: I/O Subsystem Requirements and Verifications

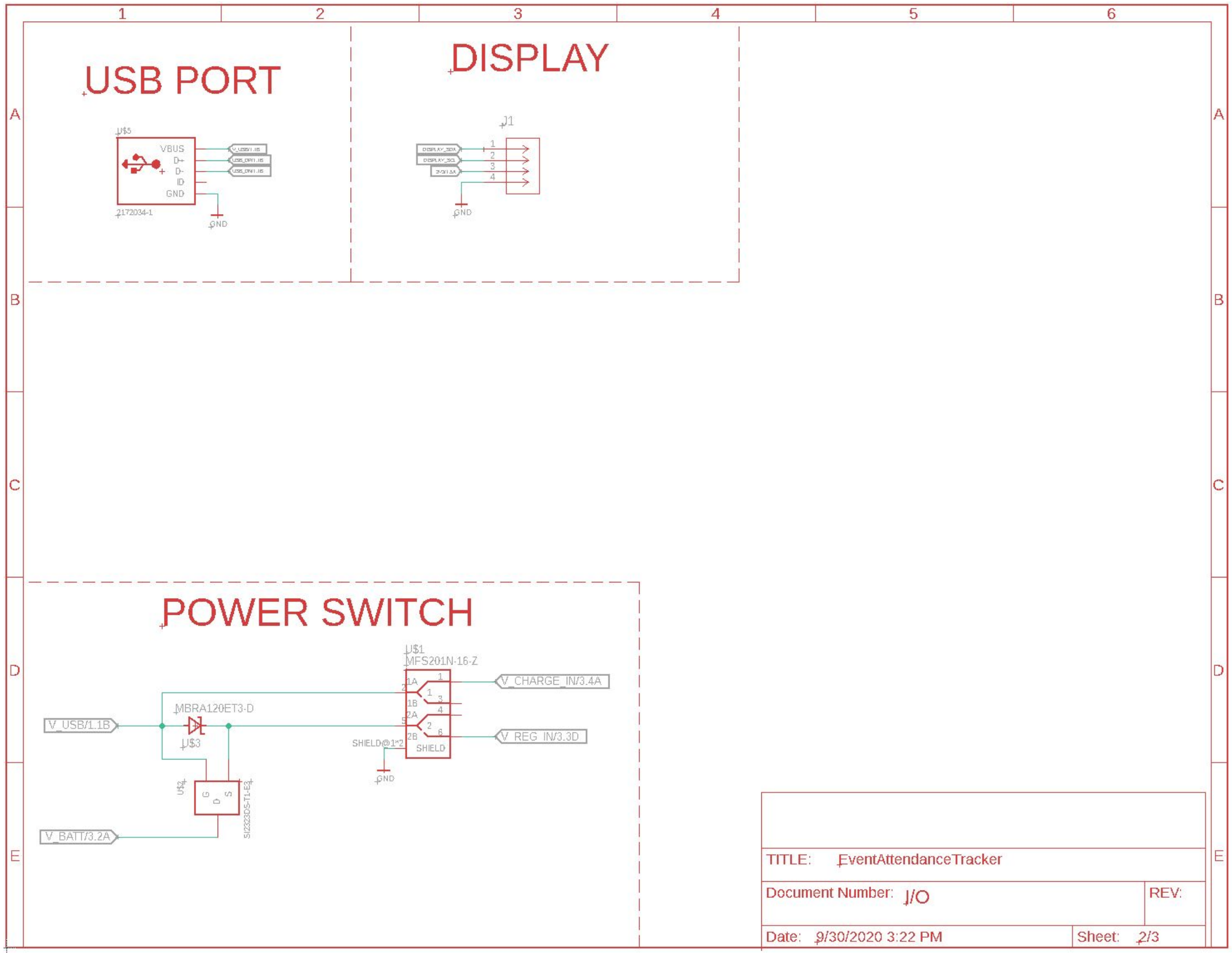

Figure 6: I/O Schematic

2.5. Control Unit

This subsystem is responsible for handling all of the wireless communication, as well as the UART to USB communication. We chose to use BLE for the wireless communication for a few reasons, the first of which being the low power consumption. Figure X [17] shows the power consumption and ranges for four RF standards, of which BLE, Wi-Fi, and Cellular are readily available to every consumer through their smartphones. From this we chose to use BLE for the low power consumption, as a high range is not a requirement of the device. In our case, the lower range helps to filter out booths that are further away to allow us to more easily and reliably determine which booth an attendee is attending.

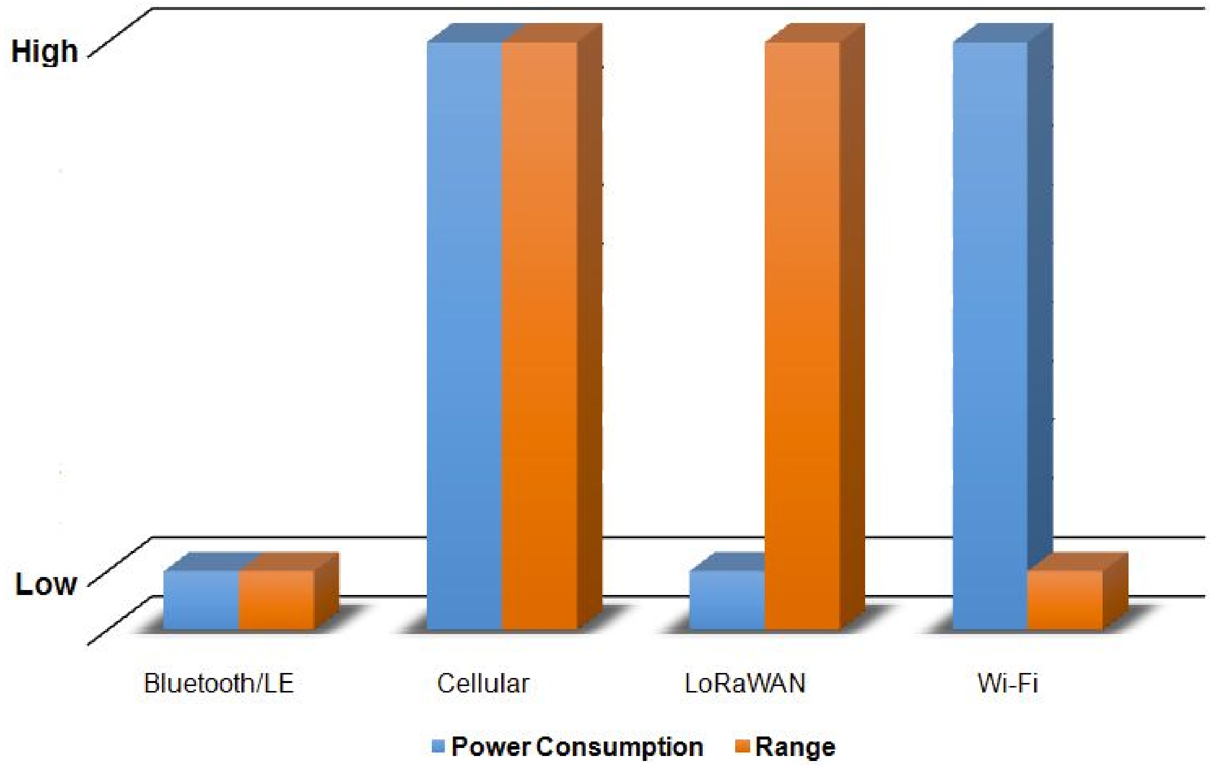

Figure 7: Power consumption & communication range across major RF standards

This subsystem is powered by the regulated 3.3V output and measures the battery voltage both output from the power supply subsystem. The subsystem interfaces with the I/O system over an I2C bus for managing the display, and over a USB differential pair to communicate with the USB port of the device.

2.5.1. ESP32 Microcontroller

Using BLE communication with the microcontroller in this subsystem, the smartphone app would be able to determine which booth it is attending and record how long the attendee was at each booth by communicating with the BLE device. This communication would also allow the device itself to record if an attendee was at the booth for a long enough period of time to be counted for attendance, which would then be displayed to the booth presenters. The other requirement the BLE communication fulfills is to quickly transfer data about the booth, ensuring several attendees can be "concurrently" interacting with the device without any smartphones "starved" of service time from the booth device.

The ESP32 was chosen for the integration of the BLE stack with a microcontroller, which enables fast integration and data collection, as well as having support for the multiple communication protocols the device will use internally (I2C and UART). It also has an integrated ADC which has acceptable accuracy for the task of measuring the battery voltage when the battery is in use.

This unit also communicates internally with the display over an I2C bus which it is the master of (bus frequency will be ~100KHz, standard-mode I2C frequency [18]). This allows the control unit to display the number of attendees, as well as the estimated battery capacity remaining.

This unit uses an internal ADC to measure, through a voltage divider, the battery cell voltage in the range (3.3V ~ 4.2V [12]) which is used in the estimation of the battery capacity remaining.

2.5.2. USB Serial Adapter

The UART to USB communication is necessary to allow the developers to easily upload the device's firmware, and for the device to be able to export to a host computer the following data: the list of attendee IDs, as well as statistics the device has collected. The control unit will be connected over a USB differential pair to the USB port which will further connect it to the host computer for communication. The USB serial adapter block within this subsystem will be responsible for initializing the USB communication, and then allow the host computer to communicate with the microcontroller seamlessly.

We chose to use a CP2104 USB to Serial Adapter, as it can support the UART communication we need, and the device has a driver built into Windows (and likely many UNIX OS's) so the device can operate seamlessly as a serial device for programming the device and for extracting data from the device after an event.

Control Unit	
Requirements	**Verification**
This subsystem must be able to be detected as a BLE device when closer than 5±0.1 meters [5].	Measure a distance of 5m from the device and a smartphone; then use a BLE scanning app to test if the device is recognized consistently across multiple scans. Repeat at a distance of 1m to confirm the device functions at short distances as well. Record pass/fail for each distance if it was recognized consistently.
When the USB port is plugged into a computer, the device must show up as a working (without errors according to the OS) serial COM/tty device.	Plug the device into at least 2 windows computers, and a UNIX based computer and verify it is listed as a serial device without additional software.
The I2C bus must be able to detect and communicate with any devices connected to the bus at the correct 7-bit address(es).	Connect an Arduino (or similar hardware) to the I2C bus, and set it to act as a slave device. Switch through all 7-bit slave addresses while the control unit microcontroller scans the I2C bus for devices, and verify the device shows up for each possible address.
The ADC of the control unit must correctly measure the voltage of the battery cell within ±0.035V of the actual voltage across the battery range (3.5V ~ 4.2V [12]) in order to provide a battery capacity estimate with 5% capacity steps with a 12-bit ADC [7].	Connect a variable bench power supply to the ADC and print out the ADC measured voltage, compare this reading against the power supply voltage read directly by an oscilloscope. Record the maximum difference between the ADC measured voltage and the value recorded by the oscilloscope.

The control unit must be able to reliably transfer up to 4kB of data within 1 second to ensure 1kB can be sent within 250mS to convey booth information and the booth ID.	Connect the control unit wirelessly to a Windows or UNIX computer as a wireless serial device. Generate 4kB of random bytes using Python (or similar), then program the control unit to transmit the data over the serial link, and record the time in the control unit it took to transfer the data (after verifying the 4kB were transferred reliably without errors on the Windows/UNIX computer).

Table 3: Control Unit Subsystem Requirements and Verifications

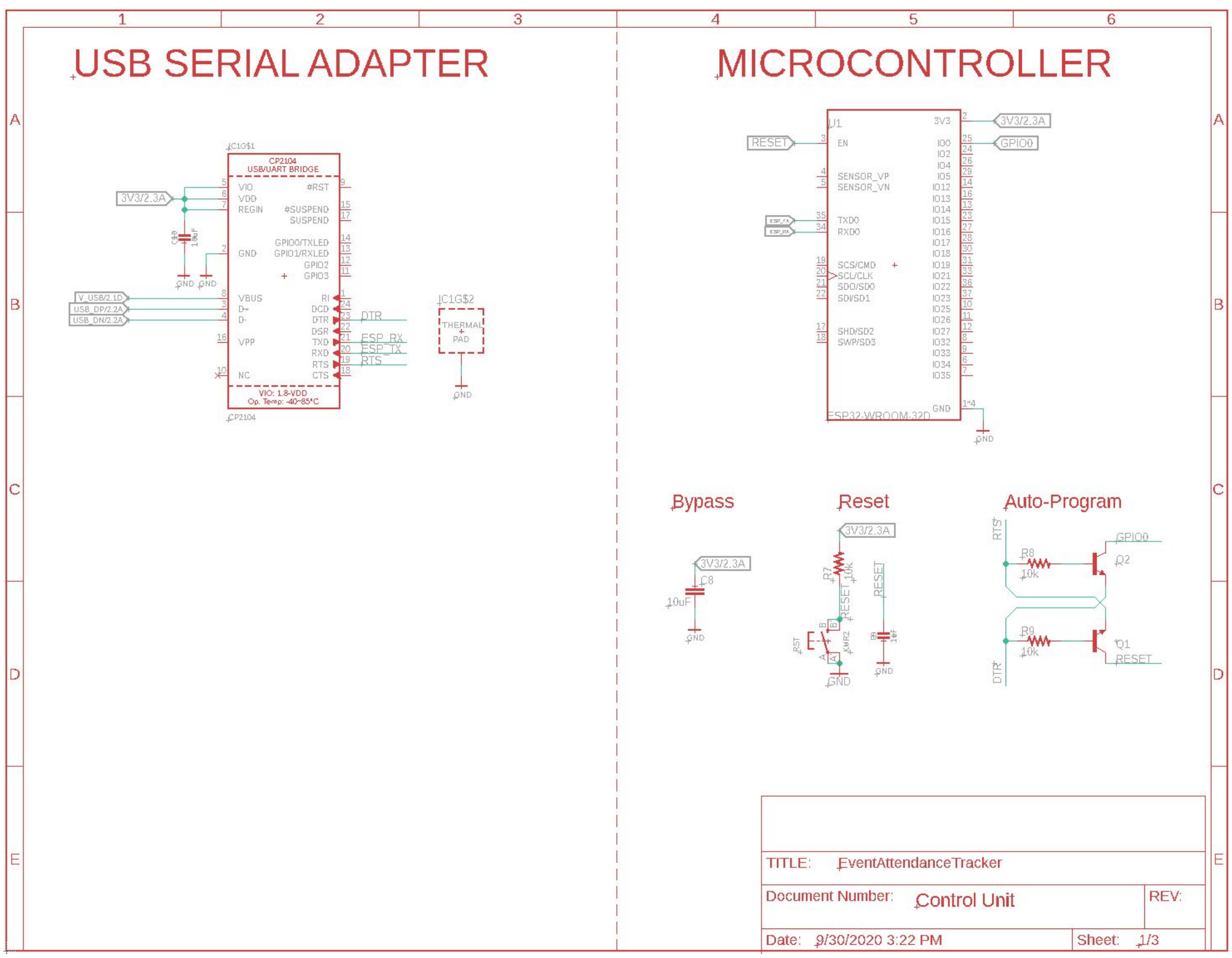

Figure 8: Control Unit Schematic

2.6. Smartphone App

The smartphone app allows the user to view which booths they have attended, share their contact information with the booth presenters if they wish, and provides basic information about the booth the attendee is currently at (including presenter contact information). These user-facing features are directly dependent on the "background" functionality of the app to communicate with the booth node devices and determine which specific booth the smartphone is attending, and for how long.

A smartphone app was chosen over dedicated hardware to reduce the cost of the system for events of all sizes, and to solve logistical problems associated with handing physical devices out to all attendees of an event (trying to get all the devices back at the end of the event).

The smartphone app interacts with the smartphone OS; this then would allow the app to communicate over BLE to determine which booth an attendee is attending, and communicate with a booth's device to exchange information about the booth.

Figure 9: Application Icon Mockups

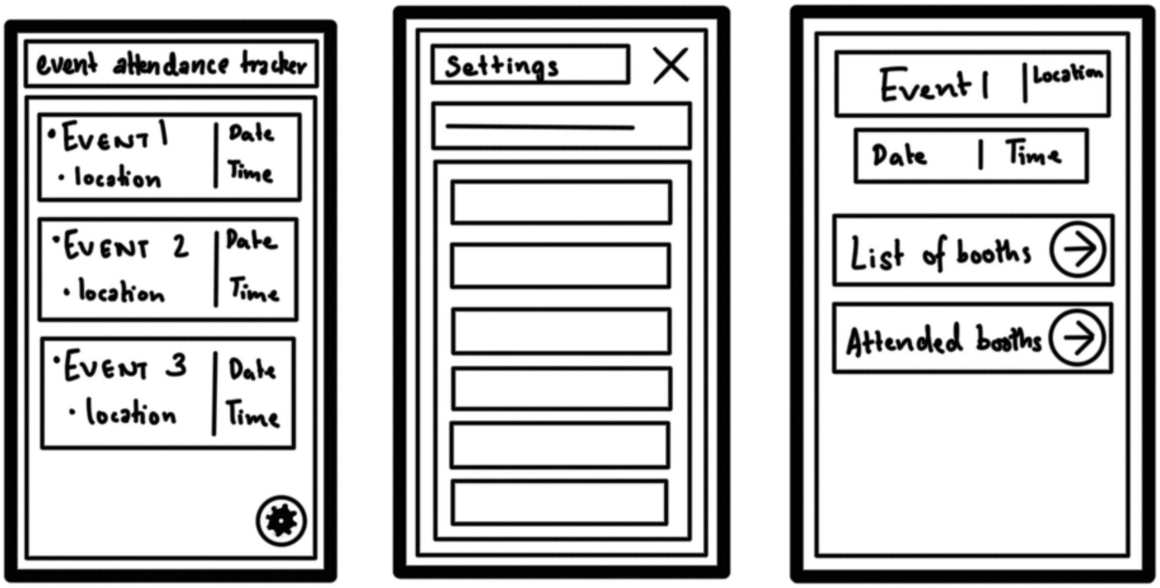

Figure 10: Smartphone App UI Mockups

2.6.1. Smartphone App Algorithm

The algorithm for the operation for the application is given in Figure 11. Upon opening the app, we would conduct a check for the earliest event that the user has listed. If the device timestamp is less than 10 minutes away from the event timestamp, then the smartphone will start searching for nearby booth devices. If not, then there will be no checks, so as to not waste resources. In the events that the smartphone checks for nearby booth devices, it will sort them based on the signal strength - in this case the RSSI values - and subsequently create a list to sort them based on the distance between the smartphone and the booths. The distance calculations will be done based on formula (11) present in section 2.7 (Tolerance Analysis). A check will be conducted to see if the estimated distance to the nearest booth is approximately less than 3m. If so, it would start a timer, and conduct another check for nearby devices. If the same booth is the closest, then it would check if the preset value for being marked as attended has been met. If not, it would continue scanning for booths, and making sure that it is still the closest. Once the time requirement has been reached, the booth is marked as attended, and the device ID for the smartphone is transmitted to the booth device. In case the device is not the closest at any point in time during the scans, then the app would re-sort the devices based on signal strength once more and begin the distance calculation and timer process once again.

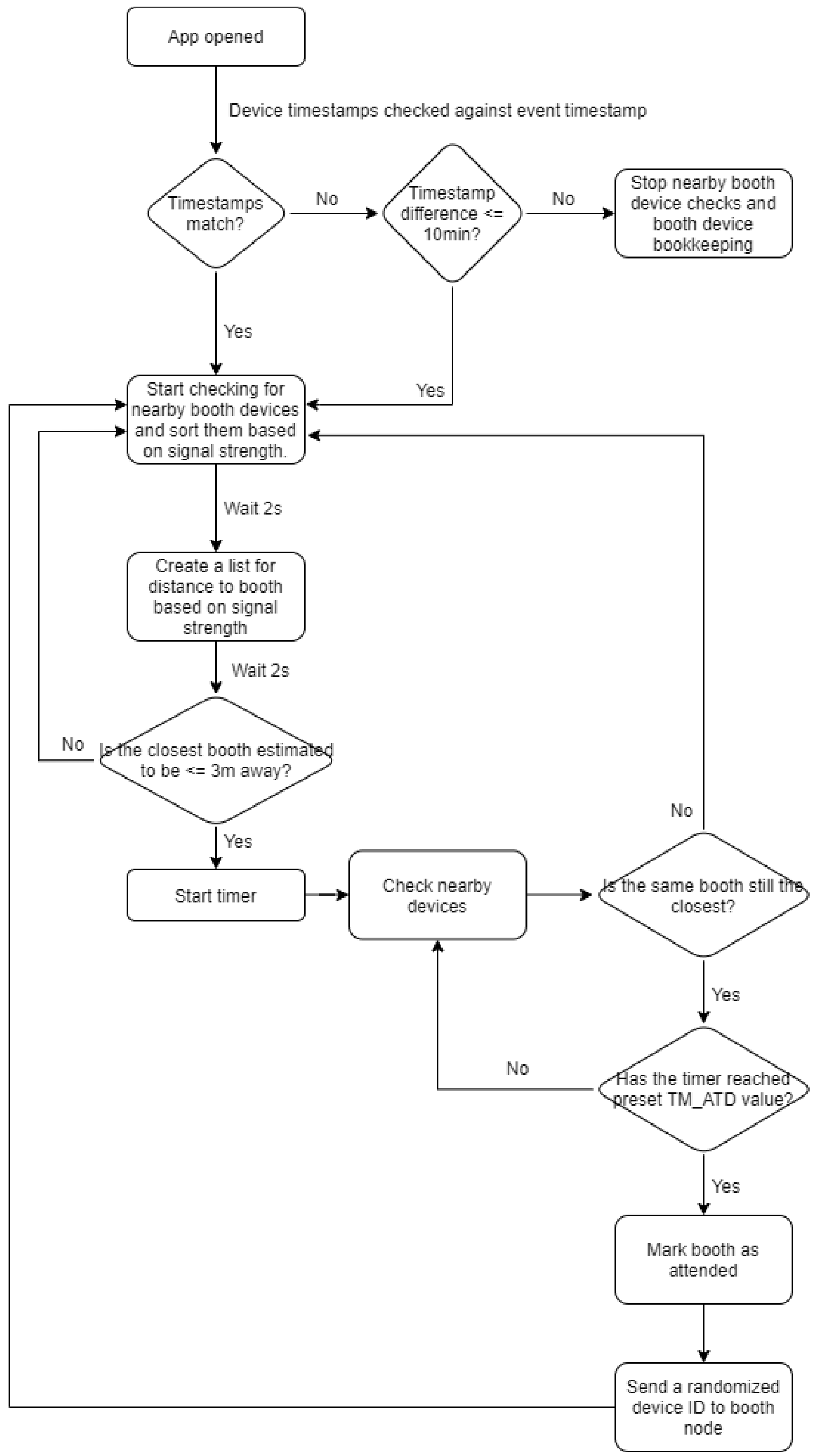

Figure 11: Smartphone App Flow Diagram

Smartphone App	
Requirements	**Verifications**
App must be able to detect all available BLE devices within at least 6 meters using RSSI values to estimate distance [19].	Approximate distance would be calculated through the formula given above, all devices which have values less than 6 would subsequently be placed in a list for “available devices”.
Smartphone app must be able to identify the BLE device with the strongest signal strength.	Device with the highest rssi value measured by the smartphone, using the formulas under the Tolerance Analysis section.
Monitor and record how long the smartphone app is within range of (according to strongest signal strength / calibrated threshold) a BLE device.	Keep a smartphone nearby a bluetooth device for a set amount of time, then move the smartphone away from the booth. Then check the data generated by the smartphone app to verify the time it estimated the smartphone was near the given bluetooth device.

Table 4: Smartphone App Subsystem Requirements and Verifications

2.7. Tolerance Analysis

The most important area of concern for tolerance analysis in our design is going to be the RF Interface between the booth device and the smartphone app. The main factor that we must consider are the calculations we make for approximate distance to the booth device based on signal strength

The determining factor for whether an individual is marked attended for a booth is based on the distance from the booth. As such we need a way to calculate the distance based on the received signal strength indicator (RSSI) values on the client devices - which is the smartphone. Some common methods of RSSI ranging are the path loss model of free space propagation and the block model of logarithmic normal. All formulas 5-12 are given in a journal by Shang et al. titled “A Location Estimation Algorithm Based on RSSI Vector Similarity Degree” [19].

The path loss model is as follows:

$$Loss = 32.44 + 10n * log(d) + 10n * log(f) \quad (5)$$

d = signal transmission distance [m]

f = wireless signal frequency [MHz]

n = path attenuation factor in the actual environment

However, in real world scenarios, the environment is not in a free space, therefore it needs to consider the shade and absorbance by obstacles, interference of scattered reflection, etc. This leads to a path loss model which follows the block model of logarithmic normal.

$$P_L(d) = P_L(d_0) + 10n * log(\frac{d}{d_0}) + X_\sigma \quad (6)$$

$P_L(d)$ = the path loss of the receiving signal when the measuring distance is $d(m)$, it indicates the absolute power value and has the units dBm

$P_L(d_0)$ = the path loss of the receiving signal when the reference distance is d_0

n = the path loss index in a specific environment. It indicates the speed of the path loss, which is increased along with increasing distance.

X_σ is in dB; it is a cover factor when the range of standard deviation σ is 4~10 and the mean value is 0; the larger the σ, the greater the uncertainty of the model.

Now, the RSSI values at clients is given by:

$$RSSI = P_t - P_L(d) \quad (7)$$

P_t = The signal transmission power.

$P_L(d)$ = The path loss at distance d.

Thus, the signal strength received from the reference nodes at distance d_0 is given by:

$$A = P_t - P_L(d) \quad (8)$$

If we take the reference distance d_0 = 1 m, when combined with equations (6) and (7) we can get:

$$RSSI = A - 10n * log(\frac{d}{d_0}) - X_\sigma \quad (9)$$

For $X_\sigma = 0$, it's mean value, we end up with:

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

From which we arrive at:

$$d = 10^{(\frac{A - RSSI}{10n})} \quad (11)$$

wherein d represents an approximation for the undetermined distance.

We can create a table for the values of n, the path attenuation, we use the formula:

$$n = \frac{(A-RSSI)}{(10*log(d))} \quad (12)$$

at predetermined distances with a large sample size. Subsequently taking the averages for them,

TABLE 1: The value of n.

	Distance								
	2 m	3 m	4 m	5 m	6 m	7 m	8 m	9 m	10 m
n	5.982	5.904	5.421	7.908	6.670	10.820	6.496	7.507	6.070

Figure 12: Values for n at various distances [19]

BLE has a minimum transfer speed of 125kbit/s which would allow us to transfer up to about 4kB of data in this time. Additionally, the maximum transfer speed would allow us to transmit up to 60kB of data [20]. Given the small amount of data we are transferring, this transfer speed (even the minimum) is adequate even considering events such as dropped packets or poor connectivity. This is enough to ensure the BLE communication is also fast enough to support our data transfer time requirement.

2.8. COVID-19 Contingency Plan

In the case that the university transitions to a fully online semester earlier than expected, the project will slightly change. As we will likely lose access to any sort of variable power supply and voltmeter, we will adjust our verification of the requirements for the power supply subsystem. We will test at discrete voltage operating points (namely, a USB voltage, a fully charged battery cell, and a nearly discharged battery cell) to verify the power supply subsystem can still operate and provide the required regulated power. Additionally, the requirement that the subsystem be able to deliver 750mA of regulated voltage would need to be verified through choice of the component (guaranteed on the datasheet) and supported through a valid implementation of the device. Other requirements of the subsystem can be validated through the use of a multimeter which we have access to personally. The I/O subsystem requirements can also be verified through the use of a multimeter, and programmatic testing of the display. The control unit subsystem requirements can all be verified as normal, except the ADC measurement requirement; this requirement will then be verified using multiple points over the discharge process of the battery cell, rather than sweeping the power supply voltage. The smartphone app subsystem can all be verified as normal in this contingency plan.

To assemble the hardware, we have the capability to assemble the PCB and all the through hole and surface mount components using a personal soldering iron and other related tools.

Beyond these changes, we do not expect the project to drastically change, though it will be more difficult to work through the debugging process we do not expect it to cause the failure of a high-level requirement.

3. Cost and Schedule

3.1. Cost Analysis

Labor:

Team Member	Hourly Wage	Weekly Hours	Number of weeks	Multiplier	Cost Per Member
Anand Sunderrajan	$38.46	15	12	2.5	$17,307
Eric Layne	$38.46	15	12	2.5	$17,307
Mason Edwards	$38.46	15	12	2.5	$17,307
				Total Labor Cost	$51,921

Table 5: Labor Cost Breakdown

The hourly wage is based on an offer a team member has, as we are all computer engineers with the same approximate hourly wage.

Parts:

Part Number	Description	Manufacturer	Quantity	Unit Cost [USD]	Extended Cost [USD]
ADP7156ARDZ-3.3-R7	3.3V Regulator	Analog Devices	1	$6.76	$6.76
B2B-PH-K-S(LF)(SN)	Battery Connector	JST Sales America	1	$0.17	$0.17
CL31A106KQHNNWE	Ceramic Capacitor (10µF)	Samsung Electro-Mechanics	6	$0.19	$1.14
CL31B105KAHNFNE	Ceramic Capacitors (1µF)	Samsung Electro-Mechanics	4	$0.14	$0.56
B4B-XH-A_(LF)(SN)	Display I2C Connector	JST Sales America	1	$0.21	$0.21
ESP32-WROOM-32D	ESP32 Microcontroller	Espressif Systems	1	$4.50	$4.50
5988230107F	LED	Dialight	1	$0.48	$0.48
MCP73831T-2ACI/OT	Li-Po Charger	Microchip Technology	1	$0.56	$0.56
SI2323DS-T1-E3	P-channel mosfet	Vishay Siliconix	1	$0.64	$0.64
MFS201N-16-Z	Power Switch	Nidec Copal	1	$1.15	$1.15
RMCF1206FT100K	RESISTOR (100KΩ)	Stackpole	3	$0.10	$0.30
RMCF1206FT10K0	RESISTOR (10KΩ)	Stackpole	3	$0.10	$0.30
RMCF1206FT2K70	RESISTOR (2.7KΩ)	Stackpole	1	$0.10	$0.10
RMCF1206FT470R	RESISTOR (470Ω)	Stackpole	1	$0.10	$0.10
RMCF1206FT47K0	RESISTOR (47KΩ)	Stackpole	1	$0.10	$0.10
MBRA120ET3G	Schottky Diode	ON	1	$0.40	$0.40
KMR231NG ULC LFS	SMT Tact Switches	C & K	1	$0.49	$0.49
DZT2222A-13	Transistor	Diodes Incorporated	2	$0.33	$0.66
2172034-1	USB Mini B Female Connector	TE Connectivity	1	$0.93	$0.93
CP2104-F03-GMR	USB to UART Bridge	Silicon Labs	1	$1.65	$1.65
	Printed Circuit Board	PCBWay	1	$5.00	$5.00

U602602	0.96" OLED Module	UCTRONICS	1	6.99	$6.99
PL803450	1500mAh 1S Lithium-Polymer Cell	YDL	1	12.49	$12.49
3D PLA-1KG1.75-BLK	3D Printer Filament	Hatchbox	1	22.99	$22.99
	M3 Screws	iexcell	1	10.99	$10.99
				Total Parts Cost	**$79.66**

Table 6: Parts Cost Breakdown

Grand Total:

$51,921 (Labor) + $238.98 (Parts for 3 boards) = $52159.98

3.2. Schedule

Week	Anand Sunderrajan	Eric Layne	Mason Edwards
October 5th	1. Work on more UI design rough sketches for the app. 2. Begin icon design in adobe suite. 3. Finish PCB schematic. 4. Finalize I/O, control unit, and power supply parts.	1. Finalized PCB Schematic. 2. Finalize I/O, control unit, and power supply components.	1. Finalized PCB schematic. 2. Finalize I/O, control unit, and power supply parts.
October 12th	1. Start designing app wireframe, and building the base for the app. 2. Finish PCB layout. 3. Finish ordering I/O parts.	1. Finish PCB layout. 2. Finish ordering control unit parts. 3. RSSI data	1. Finish PCB layout. 2. Finish ordering power supply parts.

		collection & analysis.	
October 19th	1. Design the software state diagrams. 2. Start assembling PCB. 3. Finish UI design for the app. 4. Start Display and I/O testing	1. Design the software state diagrams. 2. Start assembling PCB 3. Start implementing control unit design.	1. Start assembling PCB. 2. Begin implementing power supply design.
October 26th	1. Start RF/BLE software. 2. Implement information to be shown on OLED display. 3. Finish Display and I/O testing. 4. Work on implementing control unit design with Eric.	1. Start RF/BLE software. 2. Work on implementing power supply design with Mason.	1. Work on implementing the power supply with Eric's hardware.
November 2nd	1. Finish RF/BLE software. 2. Finish smartphone app.	1. Finish RF/BLE software.	1. Assist finalizing and debugging BLE software
November 9th	1. Combine, test, and verify RF/BLE and smartphone app. 2. Finish control unit	1. Finish control unit and power supply implementatio	1. Finish power supply implementation. Start testing and

	implementation. Start testing and verification.	n. Start testing and verification.	verification.
November 16th	1. Full system testing	1. Full system testing.	1. Full system testing.
November 23rd	1. Work out any bugs/issues and prepare for the mock demo.	1. Work out any bugs/issues and prepare for the mock demo.	1. Work out any bugs/issues and prepare for the mock demo.
November 30th	1. Demo, system testing, and start final paper/report.	1. Demo, system testing, and start final paper/report.	1. Demo, system testing, and start final paper/report.
December 7th	1. Finish Final Paper.	1. Finish Final Paper.	1. Finish Final Paper.

Table 7: Schedule

4. Ethics and Safety

Our device has a few potential safety concerns that must be addressed during the development process. This device will incorporate a lithium-polymer cell battery; this type of battery chemistry can be prone to explosions or fire when not kept in a safe voltage/current draw range or if exposed to high temperatures [12]. We must ensure that the battery control circuitry can maintain the operation of the device and keep the battery cell within safe operating ranges for both voltage and current draw [12]. This, in addition to warnings about not exposing to extreme temperatures, will help to reduce the chance of the device posing a risk of personal injury or property damage. The best way to approach this challenge would be to use conventional and reliable components and implement them to manufacturer specifications. This means we can leverage the development and testing the manufacturer went through in the design process to ensure the device operates as expected and will safely manage the battery.

This device will incorporate RF communication via Bluetooth Low Energy and any venture into RF transmission requires adhering to FCC guidelines.

> "The FCC regulates radio frequency (RF) devices contained in electronic-electrical products that are capable of emitting radio frequency energy by radiation, conduction, or other means. These products have the potential to cause interference to radio services operating in the radio frequency range of 9 kHz to 3000 GHz." [21]

Specifically, our device is what is designated as an "Intentional Radiator" [21]. For this application we will be using an RF IC incorporated into an ESP32 SOM with an intentionally limited communication power. As such there will be little to no risk of introducing adverse amounts of RF interference, even with several of these devices operating in close proximity. By using an FCC certified device [7], the ESP32, as well as responsibly utilizing the RF communication (not constantly broadcast at max transmission power) through BLE, we can insure the device does not interfere with the operation of other wireless devices nearby beyond what the standard for BLE allows.

IEEE's 7.8 Code of Ethics, Section I Policy 1 [22] "To hold paramount the safety, health, and welfare of the public…" is relevant when considering the use of a display in our device, as any flashing lights could lead to photosensitive epileptic seizures. As such the display would only be used to show attendee count and battery status, with no additional animations or flashing lights which could lead to a seizure. Every effort will be made to negate the possibility for the display to cause an epileptic seizure.

Sections 1.3, 1.6, and 1.7 of the ACM code of conduct [23] dictate that we be honest, be trustworthy, and to respect privacy. Our system can be designed in a way such that it will not have to remotely store sensitive user data; however, we still have a duty to not hoard, mine, sell, or distribute any data that we are entrusted with which is temporarily stored locally in the app. This could include names, email addresses, majors, or any other information that users wish to share. The feature that our design uses to meet these responsibilities is that all BLE communication uses a randomly generated user ID which cannot be directly correlated to any specific user. This user ID can only be correlated when that specific user consents to have their information shared with their chosen booths they visited at a particular event. Ideally, we should act as a pure middleman between the attendee and the booth by not storing any of the data, and rather simply passing it along once the user consents to sharing their information.

Further, it is our responsibility to not abuse the trust that users place in the smartphone app. We must not abuse the processing power of the device we are given access to, nor attempt to extract any other data from their personal device. In the same light, we must ensure our app does not abuse its ability to locally track the user's smartphone. To do this, we will ensure the app does not connect to or localize with nearby smartphones, and only communicate with the booth node devices for the purposes of localization.

Lastly, given the current global situation involving COVID-19, all members of the group would be following CDC recommended safety guidelines [24] to prevent the spread of COVID-19, and receive testing twice a week, per the student guidelines provided by the University of Illinois. Furthermore, we will conduct nearly all our work virtually unless in-person contact is absolutely necessary.

References

[1] “Attendance Audit Summary CES 2019,” Consumer Technology Association, Las Vegas, Nevada

[2] D. Takahashi, “CES 2019 will have 4,500 exhibitors, 2.75 million square feet of space, and 180,000 attendees,” *VentureBeat*, 01-Nov-2018. [Online]. Available: https://venturebeat.com/2018/11/01/ces-2019-will-have-4500-exhibitors-2-75-million-square-feet-of-space-and-180000-attendees/. [Accessed: 18-Sep-2020].

[3] M. Faire, “Maker Faire Bay Area Celebrates A Dozen Years As the Maker Movement Continues to Engage & Inspire,” *PR Newswire: news distribution, targeting and monitoring*, 26-Jun-2018. [Online]. Available:

[4] “Event Attendance Tracking Software & Mobile App,” *ScanTrakk*, 30-Nov-2017. [Online]. Available: https://scantrakk.com/. [Accessed: 02-Oct-2020].

[5] “Health Professions Career Fair,” *Bradley University*, 2019. [Online]. Available: https://www.bradley.edu/offices/student/scc/employers/fairs/NursingPTFair.dot. [Accessed: 02-Oct-2020].

[6] “HOW MANY RECRUITERS DOES IT TAKE TO WORK A CAREER FAIR?,” *How Many Recruiters Does It Take to Work a Career Fair?*, 12-Apr-2019. [Online]. Available: https://www.naceweb.org/talent-acquisition/best-practices/how-many-recruiters-does-it-take-to-work-a-career-fair/. [Accessed: 02-Oct-2020].

[7] Espressif Systems, “ESP32-WROOM-32D & ESP32-WROOM-32U” datasheet, 2019
Career Fair Plus. [Online] Available: https://www.careerfairplus.com/ [Accessed: Sept 16, 2020]

[8] L. Bank, How much current do OLED displays use?, 30-Jan-2020. [Online]. Available: https://bitbanksoftware.blogspot.com/2019/06/how-much-current-do-oled-displays-use.html. [Accessed: 18-Sep-2020].

[9] Silicon Labs, “USBXpress™ Family CP2102N” datasheet, Rev. 1.3

[10] “ADP7156 Datasheet.” Analog Devices.

[11] Last Minute Engineers, “Insight Into ESP32 Sleep Modes & Their Power Consumption,” *Last Minute Engineers*, 13-Dec-2019. [Online]. Available: https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/. [Accessed: 18-Sep-2020].

[12] J. Black and J. Hughes, “Lithium Battery Safety.” University of Washington, Seattle, Apr-2018.
[13] “MCP73831/2 Datasheet.” Microchip.

[14] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, and Philips, “Universal Serial Bus Standard”, Rev. 2.0, April 27, 2000

[15] A. Moore, “Lithium Polymer (Lipo) Battery Guide,” *REVOTICS*, 07-Jan-2020. [Online]. Available: https://revotics.com/articles/lithium_polymer_lipo_battery_guide?v=a284e24d5f46. [Accessed: 02-Oct-2020].

[16] “PH Connector Datasheet.” JST.

[17] J. de C. Silva, A. M. Alberti, J. Rodrigues, and P. Solic, *Power Consumption vs Range for Bluetooth/LE, Cellular, LoRaWan, and Wi-Fi technologies*. 2017.

[18] NXP Semiconductors, “I^2C-bus specification and user manual”, Rev. 6 - April 2014

[19] W. S. Fengjun Shang, “A Location Estimation Algorithm Based on RSSI Vector Similarity Degree - Fengjun Shang, Wen Su, Qian Wang, Hongxia Gao, Qiang Fu, 2014,” *SAGE Journals*, 01-Aug-2012. [Online]. Available: https://journals.sagepub.com/doi/full/10.1155/2014/371350. [Accessed: 02-Oct-2020].

[20] M. Afaneh, “Bluetooth 5 & BLE: Achieving maximum throughput and speed,” *Novel Bits*, 16-Jun-2020. [Online]. Available: https://www.novelbits.io/bluetooth-5-speed-maximum-throughput/. [Accessed: 02-Oct-2020].

[21] “Equipment Authorization – RF Device,” *Federal Communications Commission*, 20-Mar-2018. [Online]. Available: https://www.fcc.gov/oet/ea/rfdevice. [Accessed: 18-Sep-2020].

[22] IEEE Code of Ethics. [Online] Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: Sep 16, 2020]

[23] “The Code affirms an obligation of computing professionals to use their skills for the benefit of society.,” *Code of Ethics*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 18-Sep-2020].

[24] University Guideline for COVID-19 for Students. [Online] Available: https://covid19.illinois.edu/guides/students/ [Accessed: Sept 16, 2020]