Event Attendance Tracker

Team 13: Anand Sunderrajan (anands3), Eric Layne (ealayne2), Mason Edwards (masonae2)
ECE 445 Design Document - Fall 2020
TA: Dhruv Mathur (dmathur2)

1. Introduction

1.1. Problem and Solution Overview

Every year, there are hundreds of trade shows, career fairs, expos, etc. held globally wherein there are hundreds of presenters (i.e. exhibition booths) and thousands of attendees. Attendees see many booths, try to talk to presenters who see potentially thousands of attendees, and hopefully remember information about each booth. 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 each booth they stopped at.

In the United States, events such as Maker Faires and CES are prime examples. 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. All of these events have a commonality, one that has led to a quite noticeable issue. Attendees visit a multitude of booths, and it is inevitable that an attendee would fail at recalling every detail regarding the booths they visited, and the contact information for the respective presenters. Furthermore, the presence of a crowd in each booth could lead the presenters to miss out on talking to every attendee that may have been interested in their company, project, or idea. Additionally, brochures and other disposable information mediums produce a notable amount of trash. We propose a standalone, battery powered device which will continuously monitor for booth attendants via Bluetooth Low Energy. Upon detecting an attendee, it will sync with the companion smartphone app and 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 could include functionality to notify the booth presenter with information provided by the user, 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.

Currently 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 present there. 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. 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 for the number of attendees rather than a specific quantitative total for individuals interested in specified booths.

1.2. Visual Aid

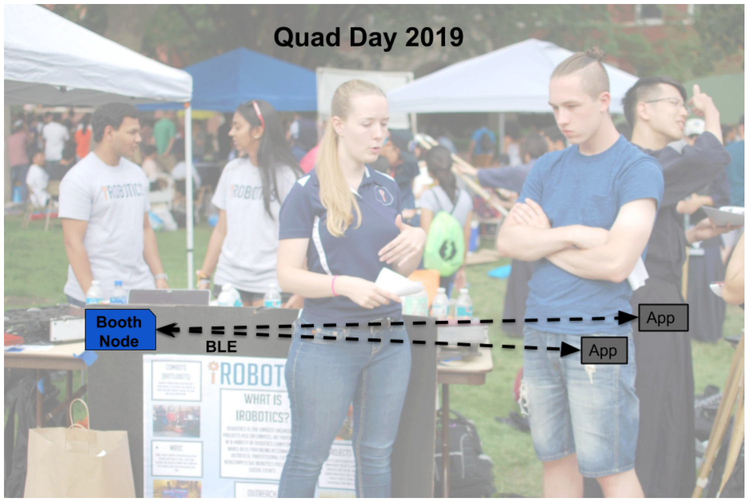

Figure **X**: Visual Aid

Figure **X** shows the operation of the device; the device will be self contained and be placed at each booth, and smartphone applications of attendees at each booth will communicate with the device to determine how long the attendee is at the booth, record information about the booth including contact information, and inform the booth device of when a user has been at the booth long enough to be counted as an attendee to give additional information in realtime to the booth presenters.

1.3. High-Level Requirements

- The device's battery will need to be able to provide sufficient power for normal operation for a period of at least 4 hours per charge.
- The interface between the ESP32 and the Smartphone itself must be able to log attendees within 3 meters [CITATION #] 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 [CITATION #] 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, thereby allowing fast, efficient, and concurrent booth polling.

2. Design

2.1. Block Diagram

This system was designed to be able 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, and have the a relatively low power requirement which gives us the ability to choose a common capacity which allows us to meet the requirement for the entire device to operate on battery power for at least 4 hours per charge. The control unit subsystem is designed around a microcontroller which is capable of BLE communication to satisfy the requirement to detect which booth a smartphone application is at, 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.

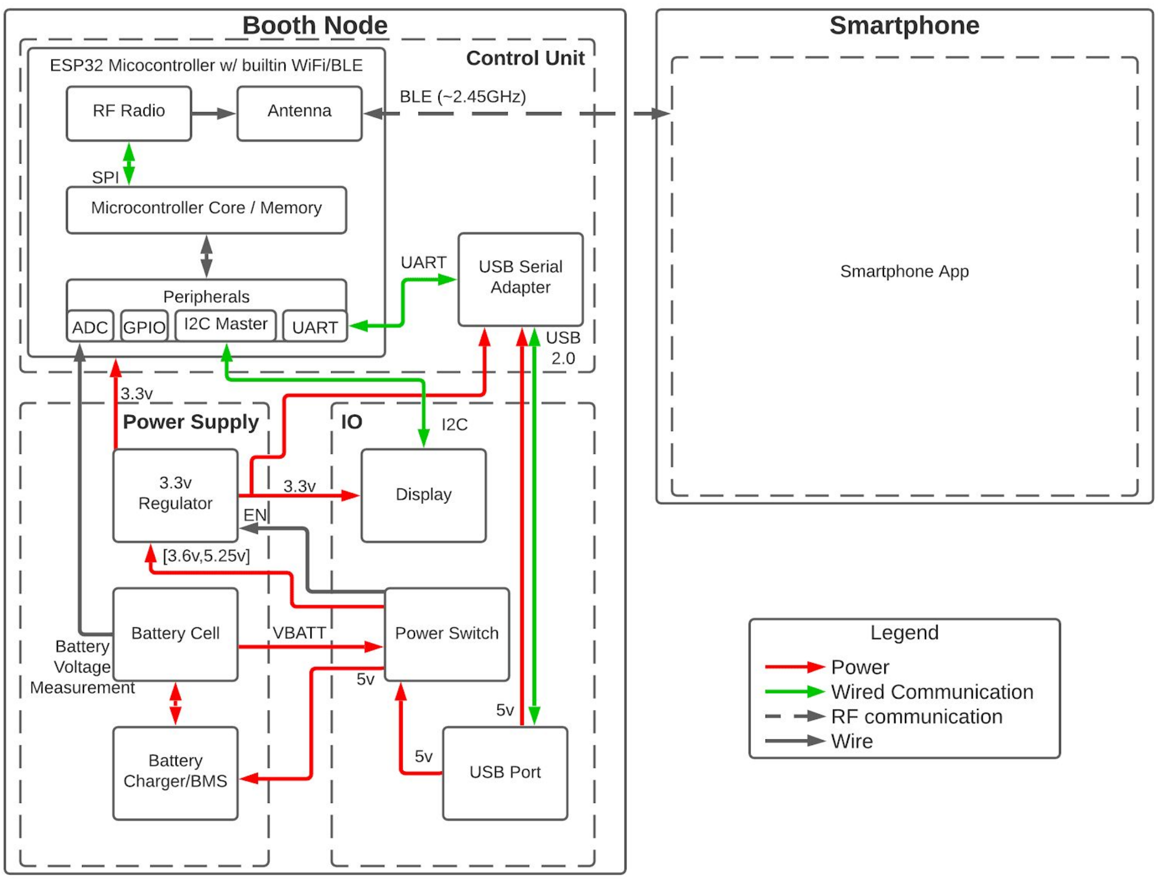

Figure X: Block diagram for System

2.2. Physical Design

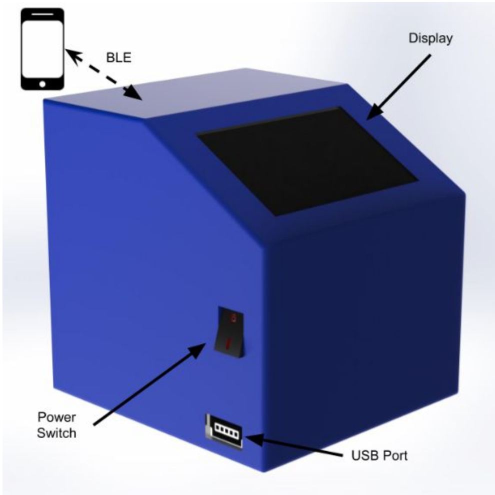

Figure X: Physical Design Render

<Add dimensions/mechanical drawings' image(s)>

2.3. Power Supply

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

2.3.1. 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 IO 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 (either during testing [CITATION] or from prior experience with the device).

$$Capacity[Ah] = Current[A] * Runtime\ [hours]$$

$$Capacity[Ah] = 0.2A * 4h = 0.8Ah$$

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).

2.3.2. Battery Charger/BMS

This block must safely charge the battery cell and monitor it to ensure it is not overcharged to an unsafe level [5]. 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) [5]. This connects to the power switch within the IO subsystem when the device is not active (in charge mode) and is always connected to the battery cell for monitoring and charging.

2.3.3. 3.3V Regulator

The device could be expected to draw up to ~200mA when powered (microcontroller: 80mA [6], display: 108mA [7], USB serial adapter: 13.7mA [8]) 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 IO 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 being drawn when the battery voltage is too low, the enable pin will be connected through a voltage divider to the input voltage to the regulator. This divider calculations can be seen below with equations and device characteristics from the regulator datasheet [CITATION HERE:ADP7156].

$$R_{up} = R_{down} * (V_{en} - 1.22v) / 1.22v$$

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 end result).

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 electronic load pulling 750mA, and at the same time measure the output voltage using a bench volt meter.
The subsystem must be able to operate normally (fulfilling all other requirements)	Repeat the test to confirm the subsystem can output a regulated voltage at 750mA, but

when the power in voltage is anywhere in the range (3.6v ~ 5.25v)[5][9].	instead sweep the input voltage from 3.6v up to 5.25v using the same bench power supply.
The subsystem, provided a steady USB voltage (Hub voltage: 5v ± 0.25v [9]), 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 [5] 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 a separate volt meter. Both the current and voltage requirements must be met, while the battery is charging, and after it is "fully charge"

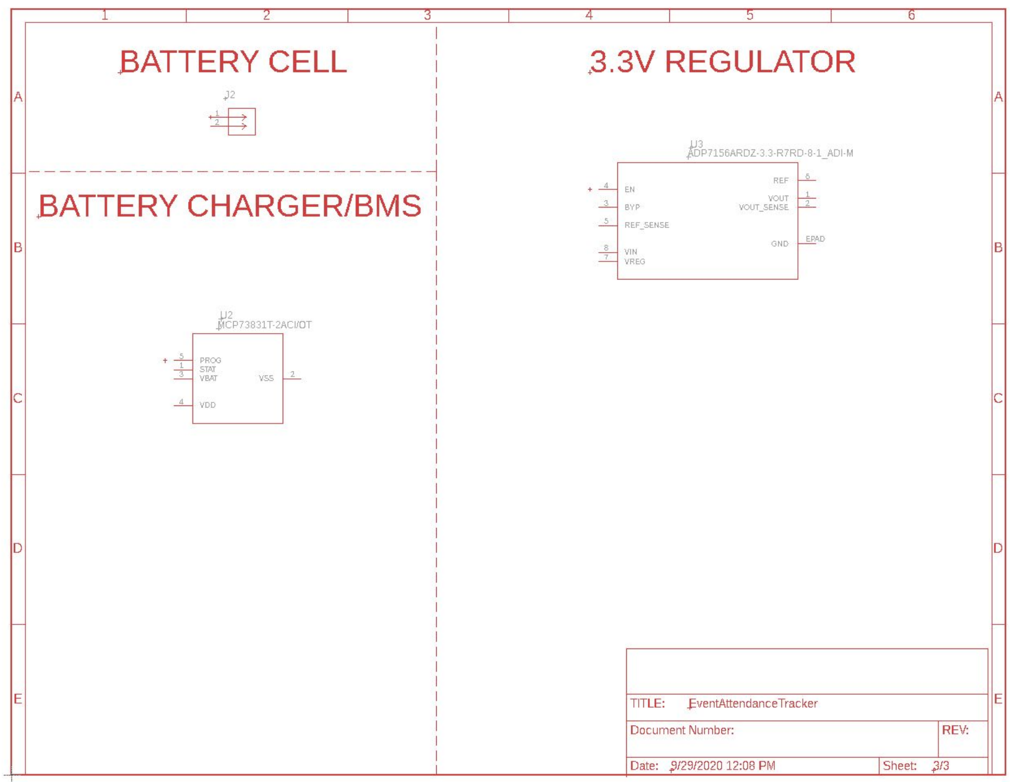

Figure X: Power Supply Schematic

2.4. I/O

The IO subsystem will provide the interfaces for the user to control the device and monitor the status of the device-smartphone application system. There are three main aspects of the IO subsystem: the display, power switch, and USB port. Each of these must operate within some constraints to ensure the device has the main functionality needed, as well as is safe and reliable.

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 remaining. Without this, there would be no feedback to the booth presenters if the device is functional, or if the device may need to be plugged in to charge soon. It will be controlled by the microcontroller over an I2C bus in order to configure the display and show the required information (battery remaining/attendee count). To operate, the display will directly draw power from the regulated 3.3v output of the power supply subsystem.

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 functionality is required to ensure the safe operation and charging of the battery to meet our battery lifetime high level requirement.

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 IO subsystem before being routed to the various components of the device. It is crucial the port is able to supply sufficient power to either charge the battery cell or power the rest of the device so it can operate as expected and all fulfill the high-level requirements. The control unit is connected over the USB differential data pair to the USB port, enabling the device and a host computer to communicate (the USB 5v may also be connected to the control unit to allow for detection when the device is plugged in). This communication will allow the device to export the list of anonymous IDs of those who attended the booth, as well as statistics about how many people attended the booth and for how long.

Requirements	**Verifications**
The display's I2C bus must be functioning properly and be able to control every 'pixel' of the display.	
The power switch must not allow more than 0mA ± 5mA to flow out to the battery charger/BMS when the device is turned on (which is the 'active' mode).	
When USB port power is available, the power switch must ensure no more than 0mA ± 10mA is allowed to flow in/out from the power supply battery voltage output.	
USB Port must be able to provide sufficient power to charge the battery cell (current up to 0.5C) or provide the constant current the rest of the device consumes when all subsystems are operating as required (at least 200mA continuous).	

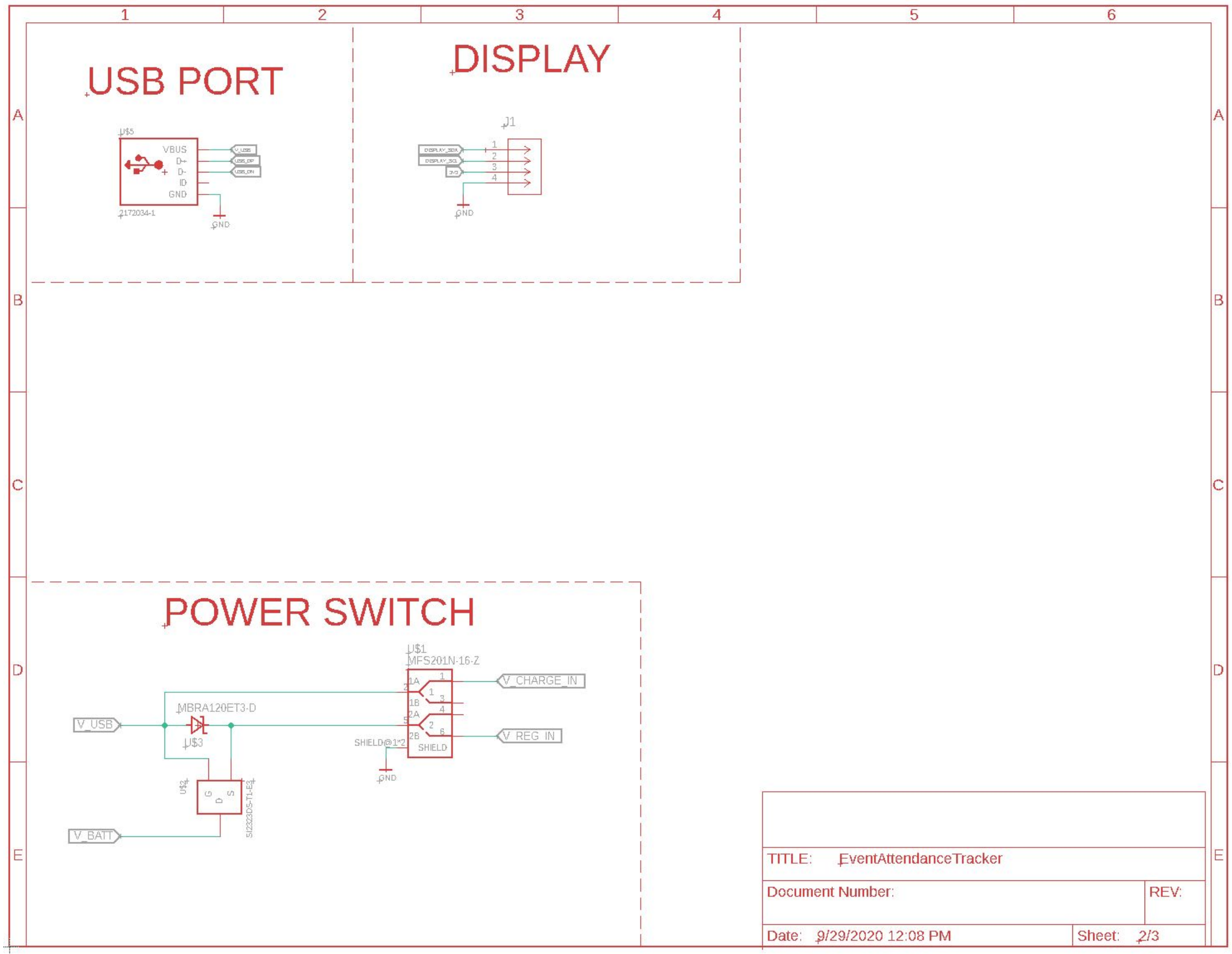

Figure X: I/O Schematic

2.5. Control Unit

This subsystem is responsible for handling all of the BLE communication, as well as the UART to USB communication. The BLE communication enables the device to record attendees, as well as provides a way for the smartphone application to determine the booth it is at, which allows the system to fulfill two of the high-level requirements.

2.5.1. ESP-32 Microcontroller

Using BLE communication with the microcontroller in this subsystem, the smartphone app would be able to determine which specific booth it is at 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 detect if an attendee were 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 connection can fulfill is to ensure user data transactions can take place within 250ms to ensure many attendees can be concurrently interacting with the device without any smartphones being 'starved' of service time.

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 [10], but is dependent on the display's capabilities). This allows the control unit to (assuming the IO subsystem is functioning correctly) display the number of attendees, as well as the estimated battery remaining to the user of the device.

Finally, this unit uses an internal ADC to measure, through a voltage divider, the battery cell voltage in the range (3.3v ~ 4.2v) which is used in the estimation of the battery remaining, as well as for disabling the device if the battery voltage drops too low for safe operation.

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.

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 IO system over an I2C bus for managing the display, and over a USB differential pair (and USB voltage sense line) to communicate with the USB port of the device.

Requirements	Verification
This subsystem must be able to be detected as a BLE device when within 5 meters.	
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.	
The I2C bus must be able to detect and communicate with any devices connected to the bus at the correct address(es).	
The ADC of the control unit must correctly measure the voltage of the battery cell within ±0.01v of the actual voltage across the battery range (3.5v ~ 4.2v).	

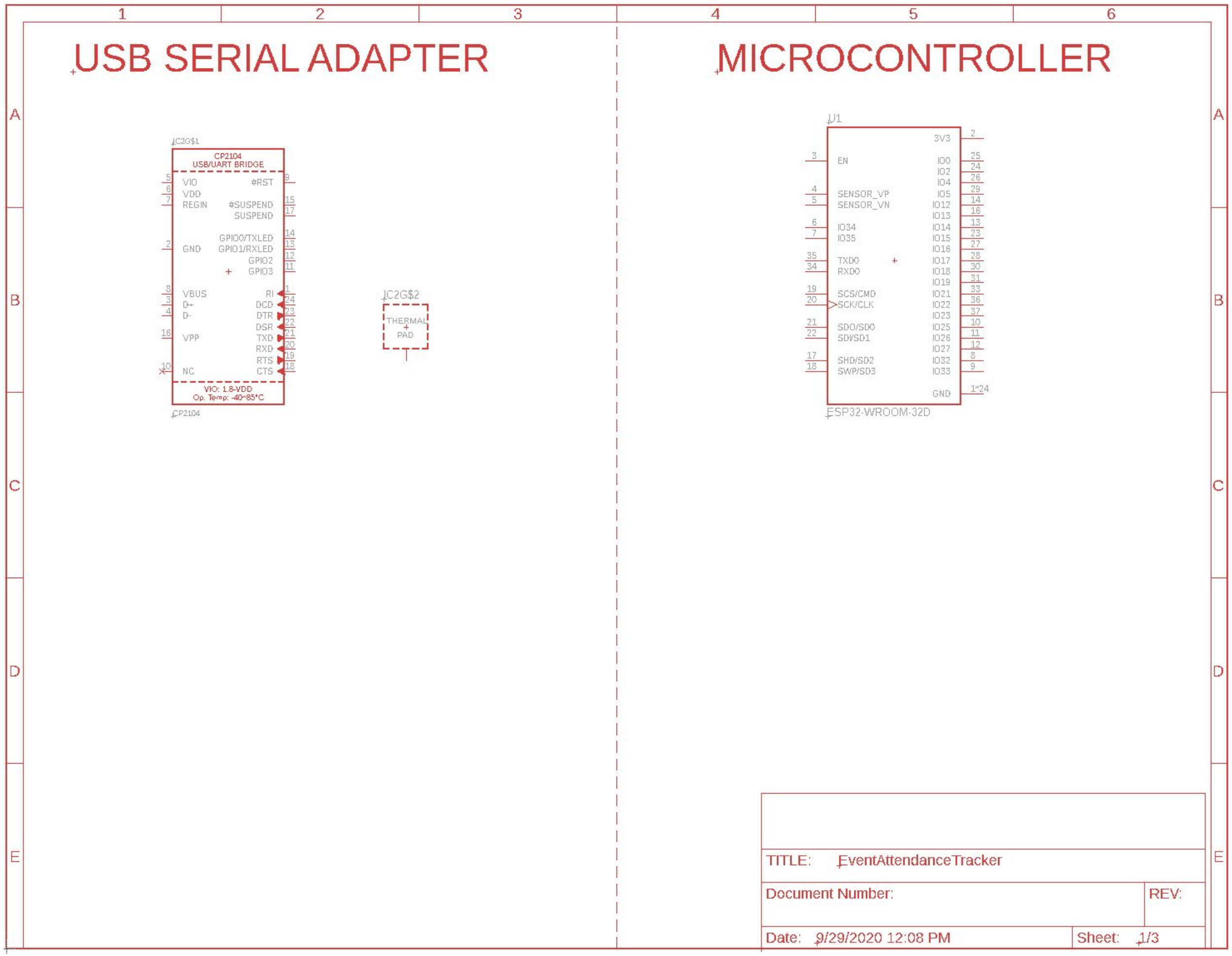

Figure X: Control Unit Schematic

2.6. Smartphone App

The smartphone app allows the user to view which booths they have attended, allow them to share their contact information with the booth presenters if they wish, as well as provide basic information about the booth the attendee is currently at (including 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 at. This directly meets the a high level requirement for the full system to be able to correctly identify which booth the attendee is at, and take action to record how long they attended each booth while informing the booth node device once any given smartphone app has been at the booth long enough to count as attended. Yet another high-level requirement is met based on the functionality of the app, as the app would be able to communicate quickly enough to allow user data transactions in under 250ms.

The smartphone app would interact with the smartphone OS, and indirectly communicate with the smartphone radio through abstraction. This then would allow the app to indirectly communicate over BLE with the booth node device to communicate and exchange information.

Smartphone Verification	
Requirements	**Verifications**
Application must be able to detect all available BLE devices within at least 6 meters	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'.
Subsystem must be able to identify the BLE device with the strongest signal strength	Device with the highest rssi value measured by the smartphone.
Monitor and record how long the smartphone app is 'at' (according to strongest signal strength / calibrated threshold) a BLE device / booth	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 application to verify the time it estimated the smartphone was near the given bluetooth device.

2.7. Tolerance Analysis

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 IO 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

Parts:

Part Number	Description	Manufacturer	Quantity	Unit Cost	Total Cost
					$
					$
					$
				Total Parts Cost	$

Grand Total:

$ (Labor) + $ (Parts) = $

3.3. Schedule

Week	Anand Sunderrajan	Eric Layne	Mason Edwards
October 5th	1. Start UI design rough sketches for the app. 2. Begin icon design in adobe suite. 3. Finish PCB design. 4. Finalize IO, control unit, and power supply parts.	1. Finish PCB design. 2. Finalize IO, control unit, and power supply parts.	1. Finish PCB design. 2. Finalize IO, control unit, and power supply parts.
October 12th	1. Start designing app wireframe, and building the base for the app. 2. Finish ordering IO parts.	1. Finish ordering control unit parts.	1. Finish ordering power supply parts.
October 19th	1. Design the software state diagrams. 2. Finish UI design for the app. 3. Start Display and IO testing	1. Design the software state diagrams. 2. Start implementing control unit design	1. Start implementing power supply design.
October 26th	1. Start RF/BLE software. 2. Finish Display and IO testing. 3. Work on implementing control unit design with Eric.	1. Start RF/BLE software. 2. Work on implementing power supply design with Mason.	1.
November 2nd	1. Finish RF/BLE software. 2. Finish smartphone app.	1. Finish RF/BLE software.	1.
November 9th	1. Combine, test,	1. Finish control	1. Finish power

	and verify RF/BLE and smartphone app. 2. Finish control unit implementation. Start testing and verification.	unit and power supply implementation. Start testing and verification.	supply implementation. Start testing and 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 30	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.

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 [5]. 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 [5]. 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." [12]

Specifically, our device is what is designated as an "Intentional Radiator" [12]. For this application we will be using an RF IC incorporated into a 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 [6], 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 [13] "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 [14] dictates 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 application does not abuse its ability to locally track the user's smartphone. To do this, we will ensure the application 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 [15] 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: https://www.prnewswire.com/news-releases/maker-faire-bay-area-celebrates-a-dozen-years-as-the-maker-movement-continues-to-engage--inspire-300471098.html. [Accessed: 18-Sep-2020].

[4] *Career Fair Plus*. [Online] Available: https://www.careerfairplus.com/ [Accessed: Sept 16, 2020]

[5] J. Black and J. Hughes, "Lithium Battery Safety." University of Washington, Seattle, Apr-2018.

[6] Espressif Systems, "ESP32-WROOM-32D & ESP32-WROOM-32U" datasheet, 2019

[7] 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].

[8] Silicon Labs, "USBXpress™ Family CP2102N" datasheet, Rev. 1.3

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

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

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

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

[14] "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].

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