SWISH TRAINER

By Andrew Jeon, Pujith Bezawada and Michelle Cheong

Design Document for Senior Design/ECE 445 Spring 2021

TA: Alex Sirakides

03 March 2021

Team 30

1 Introduction

1.1 Problem and Solution Overview:

Due to the advent of the COVID-19 pandemic, many people wanting to shoot some hoops find themselves at the court alone. There's already variations of rebound machines, self-trainers, and passing machines. But these are either not portable and too expensive for your average player or they are too simple to effectively help in all the ways that players need. What we are proposing is a self-trainer, built for the average hooper. Someone who doesn't have access to various trainers, machines, and regimens but still wants to put in work, anywhere.

Our problem has always existed, in that every day hoopers have never really had anything like this that helps track shooting statistics. The solutions out there [1] fail to provide a truly satisfying solution to this problem. Specifically, The Shot Tracker App reportedly has many sensor issues [1] and the Wilson X Connected Ball reportedly has no replaceable battery [1], which means ballers simply have to give up the shot tracking as soon as the ball runs out of battery. Every other variation of a solution to this problem currently in the market either involves sensors [2] or heavy-duty equipment such as tripods [1] that are not practical for every day hoopers. Additionally the ultrasonic sensors utilized in a lot of these designs are reportedly inaccurate depending on what material the ball is made of, and also limited to a range of 10 meters [3].However, our problem has been exasperated recently by the advent of the COVID pandemic. Gyms, athletic clubs, and even outdoor courts have been shut down all across the country [4]. Now, hoopers are expected NOT to ball with each other and instead either stay home or shoot around by themselves. With this in mind, it is now even harder to get a productive training session by yourself. This is where our practical solution with minimal equipment really shines.

We are proposing a combination of a remote and an app to solve this issue. First, a portable remote that can be attached to the edge of a player's shorts, for convenience, collects the data of which shots are made and where they are being made from. These predetermined spots around the court should be set on the app, and the remote would have a yes and no button that the shooter has to click to account for the success rate of the shot. The remote will also have a shift button to let the app know when a person is switching spots around the court. The app itself will be present to view the results and input the spots the shooter wants to shoot from. The app will take all this data of the shots taken from all these varying spots and create a report of percentages from different spots around the court and analyze what spots the person would have to work and more. It would report percentages for each day/session and save the varying percentages under each date. The person does not have to have their phone on them as our bluetooth distance of <100 ft should enable successful information relay at most practical distances.

1.2 Visual Aid

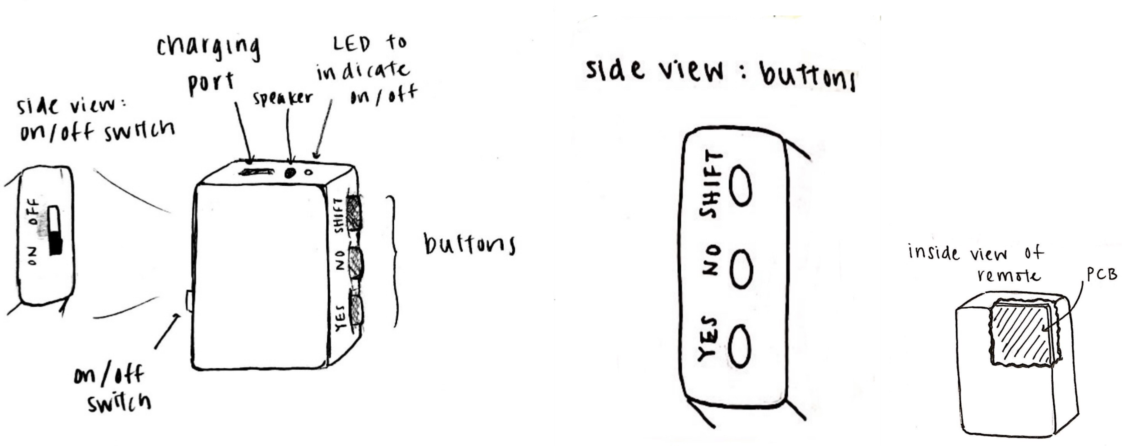


Figure 1: SWISH Trainer remote

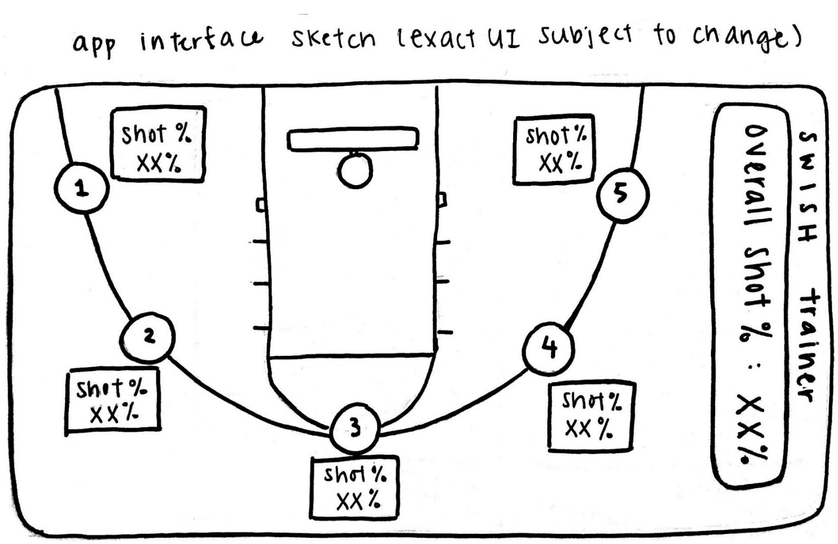


Figure 2: SWISH Trainer App Interface

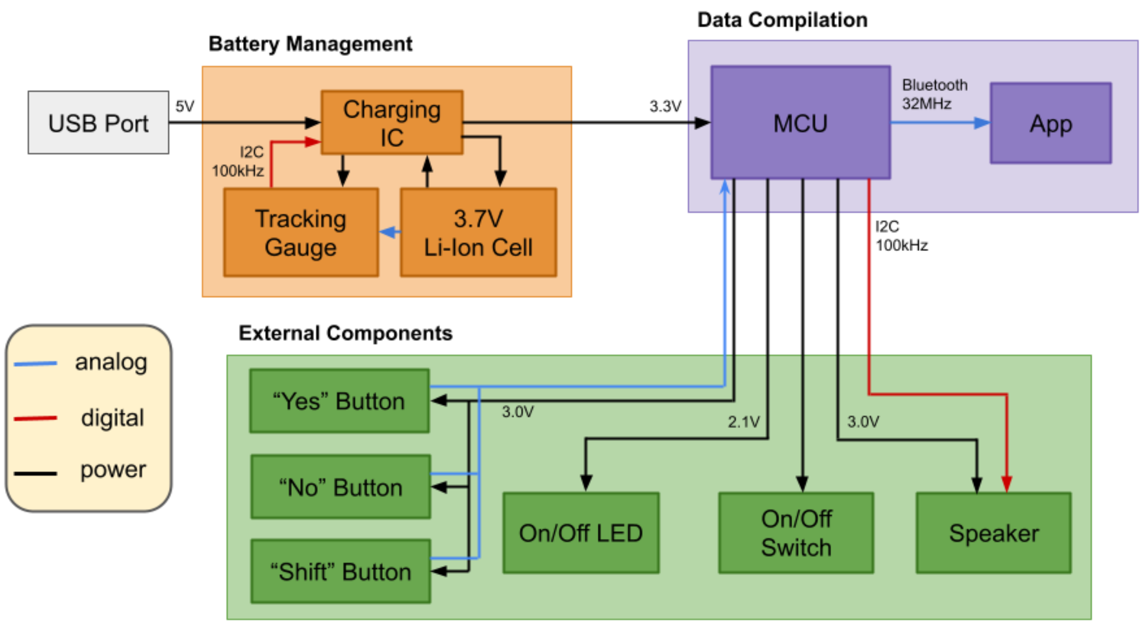
1.3 High-Level Requirements

- When any of the buttons/switches are pressed, the system must clearly notify the user via speaker or light.
- The remote has to relay the correct inputs into the app connected via bluetooth to the phone. We should be able to see this as the values update in real time, and the time between button input and reporting on the app should be <5 seconds.

- Our app must accurately display shot percentages (hundredth) from at least 5 different spots throughout a basketball court. We can clearly see this as the interface we plan to create on this app is a half court that shows the percentages of the shots at varying points.

2 Design

2.1 Block Diagram



In this block diagram, the external components are directly connected to the microcontroller. Signals coming from the buttons and the on/off switch are registered in the microcontroller, the microprocessor sends a signal to the external components to reflect the correct outcome (i.e. if the switch is in the "on" position, then the on/off LED will light up; if a button is pressed, the speaker will play a beep). The bluetooth block is also directly connected to the microcontroller, so signals sent to the microcontroller from the buttons can be sent to the bluetooth block and communicated to the app. The app can then perform the necessary calculations to correctly display the shot percentage.

2.2 Battery Management

The Battery Management regulates the charging of the lithium-ion cell and powers the entire system (microcontroller and external components) by providing a consistent 3.3V. This subsystem contains four main components: the battery charger IC, the gas gauge IC, the lithium-ion cell, and the USB charging port. The power budget is 150mAh at 3.7V. The microcontroller consumes the majority of the power (3.3V), followed by the battery charger IC and the gas gauge IC. This Battery Management System (BMS) will be used for charging/discharging.

2.2.1 Lithium-Ion Battery Charger with LDO

The lithium-ion battery is charged through a charging IC, the STNS01. The STNS01 takes an input supply voltage to charge the battery and provide power to the LDO regulator within the chip, using a CC/CV algorithm (i.e. constant-current to a specific voltage and then constant-voltage to a floating voltage value of 4.2V). This device also contains a charger enable input that will stop the charging process when conditions are outside of the normal operating range (outside of 0 - 45°C). When there is no input supply voltage, the IC will be powered by the battery.

Requirement	Verification
1. Charges battery up to floating voltage of 4.2 V.	1. Discharge to 3.7V cell voltage. 2. Charge battery from STNS01 from input of 5V. 3. After charging is complete (logic low on charge enable pin pf STNS01), measure voltage of battery to check it is between 4.16-4.23V.
2. Battery temperature remains between 0°C and 45°C while charging.	

2.2.2 Tracking Gauge

The tracking gauge IC STC3155 implements a gas gauge for battery monitoring, utilizing current sensing and other measurements of the battery voltage to estimate the state-of-charge of the battery. There is also an internal temperature sensor for temperature compensation. This device is connected to the lithium-ion cell to effectively monitor charge level and is also connected to the charger enable input of the battery charging IC.

Requirement	Verification
1. Low battery alarm is triggered when battery charge is below programmed value 3.3V.	1. Discharge battery to 3.3V. 2. Check that the alarm pin of STC3155 is at logic low.

2.2.3 Lithium Ion Cell

The lithium-ion cell needs to keep the remote powered for up to ~30 hours of continuous use on the basketball court and thus must have a capacity of 150mAH. Monitoring of the battery (charge left, temperature) is done by the tracking gauge and charging ICs.

Requirement	Verification
1. Stores 150 mAH of charge.	1. Fully charge the battery to floating voltage of 4.2V. 2. Discharge the battery at ~5mA for 30 hours 3. Measure the battery voltage and ensure that it stays above 3.0V.

2.2.4 USB Charging Port

The USB charging port feeds 5V to the battery charger IC.

Requirement	Verification
1. Voltage is 5V at node between USB charging port and battery charger IC.	1. Measure voltage of the node between the charging port and battery charger IC, make sure it remains within +/-0.5% of 5V while charging.

2.3 Microcontroller

The microcontroller reads I/O information from the Yes, No, Shift, On, Off buttons and will also power the speaker and the LED in the external components. The microcontroller will then store this I/O information in its data memory where it will then apply this data to communicate through a bluetooth device to our app.

Requirement	Verification
1. Flash memory requirement of < 5Mb before data is transmitted to the app	1. Ensure that each time data is inputted via buttons, it is reported on the app

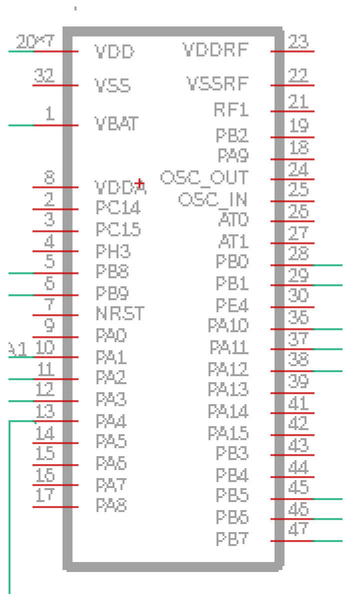


Figure 3: ST Microcontroller

2.4 Data Compilation

The data compilation part of this project entails a bluetooth device that connects the remote to the other part which is an app. The app does the brute of the calculations and analysis of this project.

2.4.1 Bluetooth

The bluetooth device we are going to use connects the remote wirelessly to the app making the relaying of information from the varying buttons on the remote easy and efficient. This is going to be a part of the MCU. (<100 ft)

Requirement	Verification
1. The inputs from the remote have to be correctly transmitted to the app.	1. The percentages are correctly updating at each of these spots on the interface of the app. This is easily verifiable through independent calculation.

2.4.2 App

The app itself takes in all the varying inputs of the remote and receives this data via bluetooth. The app then portrays all the shots taken from the varying spots on the court. The interface of this app consists of a half basketball court and the five spots the person is shooting from and shows the percentages from each of these spots.

Requirement	Verification
1. The interface must display the percentage of every shot at the varying spots	1. The percentages are correctly calculated at each of these spots on the interface of the app. This can be verifiable through independent calculation.

2.5 External Components

The external components on this project consist of three buttons, a switch, a LED, and a speaker. All these components are relevant in taking in input and for the user to recognize that the input was taken successfully.

2.5.1 “Yes”, “No”, “Shift” buttons

These three buttons are what the user must press. The “yes” button should be pressed if the shot goes in. The “no” button should be pressed if the shot is missed. The “shift” button should be pressed if the shooter is moving to the next spot in the shooting rotation. These three buttons encompass the external inputs that are involved in our project.

Requirement	Verification
1. These buttons must be able to register correctly, so we can count our shots. These buttons should be debounced so they only read one input.	1. The input is read correctly, specifically it shows up on our app. The speaker emits one beep for yes, two beeps for no, and three beeps for shift.

2.5.2 “On/Off” Switch

The “on/off” switch’s purpose is to turn the remote on or off respectively. This allows for power to flow throughout the board and saves energy when the remote is off.

Requirement	Verification
1. The switch must turn the circuit on and off.	1. The led lights up when the switch is turned on.

2.5.3 LED & Speaker

The LED and speaker are present to let the user know that the buttons and switch are working correctly. The LED will light up when the remote is on and will be off respectively if the remote is off. The speaker will make a beep if the “yes” button is pressed, two beeps if the “no” button is pressed, and three beeps if the “shift” button is pressed.

Requirement	Verification
1. The LED and speaker must signal to the user that the remote is functioning correctly.	A. The led lights up when the switch is turned on. The speakers emit a noise when each button is pressed.

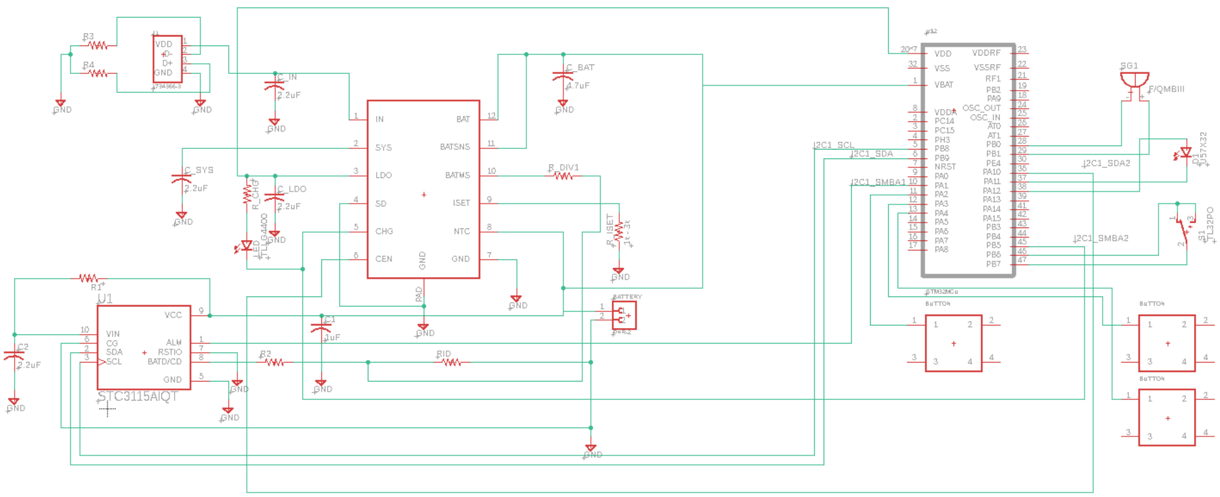


Figure 4: Remote Schematic

2.6 Risk Analysis/Tolerance

One of the main usability components of our project is the fact that a shooter/player can view their shooting stats at will whenever they feel like it, whether this be after an entire session or after a single shot. As a result, we not only need to ensure that the data that is inputted by the remote properly, but that it is inputted and displayed on the app on time.

Upon multiple test trials, we have found that the average time it takes to take out a phone from said person's pocket and unlock it and open up an app is anywhere from 5~10 seconds depending on multiple factors including password length, phone build, and connectivity latency. To ensure a timely reporting time of data to our app layout, we will ensure that our reporting time (T_R) < 5 seconds.

On-board time (t_b) + remote-app transition time (t_{ra}) + app computation time (t_a) < 5 [sec]

The on-board time should be minimal as these will be wired connections and as such the data transfer time should be near instantaneous.

The remote-app transition time can be calculated by accounting for how many bits of data we want to transfer at a time. The on-air data rate is 1Mbps so in order to meet our requirement at the minimum we should ensure that we transfer data that is < 4Mb at a time so we don't exceed our tolerance of 5 seconds.

The app computational time will depend heavily on our actual code and implementation, as such we cannot specify this at this time. This also depends on the strength of the user's connection to WiFi. With strong WiFi and the Bluetooth v5.0 integrated into our MCU, the app's computational time in terms of updating the percentages should be within a second of the input getting registered by the remote; once again, this time does not exceed our tolerance.

3 Cost & Schedule

3.1 Cost Analysis

The overall cost of the project is split into two categories, fixed (personnel pay) and variable (parts and manufacturing). Our parts and manufacturing prototype costs are estimated to be $35.08 each. Bulk costs are provided for scaled up production.

Type	**Description**	**Fixed Cost**	
Fixed	$40/hr 10 hr/day 5 days/week 16 weeks 3 people	$96,000	
	Part	**Cost (Prototype)**	**Cost (Bulk)**
Variable	Battery Charging IC	$1.88	$1.69
	Tracking Gauge IC	$3.04	$2.73
	3.7 Li-Ion Battery Cell, 150mAH	$5.95	$5.95
	USB Type-A Charging Port	$0.46	$0.44
	Microcontroller	$5.00	$4.49
	12mm Tactile Buttons (3)	$1.80	$1.80
	SPDT Slide Switch	$0.75	$0.75
	Speaker	$1.95	$1.95
	3mm Yellow Green LED	$0.25	$0.15

	PCB	$4.00	$4.00
	Assorted passive components (e.g. resistors, capacitors, etc)	$10.00	$10.00
VARIABLE SUBTOTAL		$35.08	$33.95

3.2 Schedule

Week	Andrew	Michelle	Pujith
3/3/21: Design Document	Introduction/Tolerance Analysis, RV, Schematics/PCB, Ethics & Safety	Block Diagrams, Cost, RV, Schematics/PCB	Ethics & Safety, High Level Req. (Software)
3/9/21:Parts	Circuit Design, Check Schematics and connections (need resistors capacitors on external components?)	Order Parts, Check Schematics	Initial App Dev.
3/15/21:PCB 1	Stress-Test First Step nodes	Complete Power Tests	Initiate First Step Programming
3/22/21: PCB 2	Fix bugs in routing steps for first step nodes	Collect data for power vs data from nodes	Complete final design for data transmission
3/29/21: PCB 3	Stress-Test Second Step nodes	Convert routing steps to Second Step	Work and Test the data transmission
4/12/21: Mock	Prototype Experimentation	Prototype Experimentation	Fix any bugs in data transmission
4/19/21: Final	Prep Final Demo and Start on Final Report	Prep Final Demo and Start on Final Report	Prep Final Demo and Start on Final Report

4 Ethics & Safety

There are a few safety concerns with our project. Primarily, the use of a lithium ion battery does provide us with some challenges. Damage to lithium ion batteries can occur when they are dropped, crushed, or punctured. Additionally, damage can occur when temperatures are too high (above 130°F), or if the batteries are charged in temperatures below 32°F [5].

If a lithium ion cell is damaged, then the possible heat release from this damage can result in thermal runaway which is when this excess heat damages other cells which leads to a chain reaction of heat release. Additionally, during thermal runaway the excess byproducts released from this process may ignite or cause other harmful side effects [5].

In order to ensure that lithium ion cell runaway or other damage does not occur, we will monitor the temperature with a thermistor and immediately isolate the battery from the rest of the system if the temperature reaches above 115°F or below 32°F. Additionally, we will make sure our PMIC does not charge the lithium battery over 4.21 V. These precautions are in strict adherence to the IEEE Code of Ethics #1: “to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment” [6].

In order to prevent water damage by the form of short circuits, we will have to adhere to IP65 waterproofing standards. The case will be able to withstand water jets from various directions without resulting in damage to the circuit within [7].

In terms of privacy laws with respect to the app, we can secure the data to ensure it is ethical and private. This is in compliance with the IEEE Code of Ethics #1: “to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment” [6] However, it should be mentioned that this is not much of a necessity to us as the information and data we have is not higher-level information that could potentially cause harm to the user of this data if it is leaked. If we did want to secure the app despite this, we would have to write encrypted code and abide by the privacy laws set in place by the app store at the current time. We want this app to be able to live by IEEE Code of Ethics #3: “to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist” [6]. We want the app to abide by all these laws to prevent any potential harm to the user, but as I stated earlier, the only potential harm that can come from this information is percentages being shown by the app being potentially leaked. We also feel like it is important to bring up IEEE Code of Ethics # 6: “to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations[6].” The overall idea of this project is the emphasis on improving a practice or personal training for a basketball player, and we plan to abide by all the IEEE Code regulations to the best of our ability.

5 References

[1] dbkobr, “Best Basketball Shot Tracker [2021 Review] Shot Counter App Trackers,” *Dear Basketball*, 29-Sep-2020. [Online]. Available: https://dearbasketball.com/best-basketball-shot-tracker/. [Accessed: 02-Mar-2021].

[2] “Automatic, Real-Time Basketball Stats and Analytics,” *ShotTracker*. [Online]. Available: https://shottracker.com/. [Accessed: 02-Mar-2021].

[3] K. Gross, “Ultrasonic Sensors: Advantages and Limitations,” *MaxBotix Inc.*, 28-Oct-2020. [Online]. Available: https://www.maxbotix.com/articles/advantages-limitations-ultrasonic-sensors.htm/. [Accessed: 23-Mar-2021].

[4] J. Zillgitt, “Cities take drastic measures, like removing rims, to keep people from playing basketball on public courts,” *USA Today*, 01-Apr-2020. [Online]. Available: https://www.usatoday.com/story/sports/basketball/2020/03/31/basketball-rims-removed-nets-tied-prevent-games-public-courts-coronavirus/5095142002/. [Accessed: 02-Mar-2021].

[5] “UNITED STATES DEPARTMENT OF LABOR,” *Safety and Health Information Bulletins | Preventing Fire and/or Explosion Injury from Small and Wearable Lithium Battery Powered Devices | Occupational Safety and Health Administration*. [Online]. Available: https://www.osha.gov/dts/shib/shib011819.html#:~:text=flashlights, and defibrillators.-,Hazards,fire and/or explosion hazard. [Accessed: 16-Feb-2021].

[6] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 16-Feb-2021].

[7] Haas, B. (2021, February 19). Ip ratings explained: The mysteries of ip65, ip66, and ip67. Retrieved March 03, 2021, from https://www.budind.com/blog/2014/02/the-mysteries-of-ip65-ip66-and-ip67-rated-enclosures-explained/

[8] “Wearable sensor unit reference design for fast time to market,” 2016. [Online]. Available: https://www.st.com/resource/en/data_brief/steval-wesu1.pdf. [Accessed: 20-Feb-2021].

[9] “Li-Ion linear battery charger with LDO,” 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/stns01.pdf. [Accessed: 28-Feb-2021].

[10] “Gas gauge IC with alarm output for handheld applications ,” 2018. [Online]. Available: https://www.st.com/resource/en/datasheet/stc3115.pdf. [Accessed: 01-Mar-2021].

[11] “Wearables,” *Wearables Solutions*. [Online]. Available: https://www.cypress.com/solutions/wearables-solutions. [Accessed: 16-Feb-2021].

[12] “Battery Management Reference Design for Smartwatch & Wearable Applications.” Oct. 2015.<https://www.ti.com/lit/ug/tidub09/tidub09.pdf?ts=1613346750822&ref_url=https://www.ti.com/sitesearch/docs/universalsearch.tsp?searchTerm=block diagram for smart watch> [Accessed 16 February 2021].

[13] Lutkevich, B., 2021. *What is a Microcontroller and How Does it Work?*. [online] IoT Agenda. Available at: <https://internetofthingsagenda.techtarget.com/definition/microcontroller> [Accessed 16 February 2021].