SWISH TRAINER

Final Report

By
Andrew Jeon
Michelle Cheong
Pujith Bezawada

TA: Alex Sirakides
April 3, 2021
Team 30

Contents

1. Introduction

1.1 Idea

The inspiration from our project came from an everyday occurrence that we ran into. As people who enjoy basketball, we realized that a lot of players shoot by themselves and need something to track their statistics. We initially sought to make a self rebounder, but we eventually pivoted into a basketball shooting self-trainer in order to address this need.. Especially with the advent of COVID, many more players are forced to play or shoot around themselves. Gyms, athletic clubs, and even outdoor courts have been shut down all across the country [1]. Now, hoopers are expected NOT to ball with each other and instead either stay home or shoot around by themselves. Due to these circumstances, we felt that a self-shooting basketball trainer was sorely needed.

The currently available solutions out there [2] fail to provide a truly satisfying solution to this problem. The Shot Tracker app [2] and the Wilson X Connected Ball are the two premiere solutions that currently exist. The Shot Tracker App reportedly has many sensor issues [2] and the Wilson X Connected Ball reportedly has no replaceable battery [2], which means ballers simply have to purchase a new ball every time the battery runs out or the sensors fail. Every other variation of a solution to this problem currently in the market either involves sensors [3] or heavy-duty equipment such as tripods [2] 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 [4].

So, while the need for a solution has been exasperated recently by the advent of the COVID pandemic, the inadequacy of current products is accentuated. With this in mind, it is now even harder to get a productive training session by yourself. This is where our practical, effective, and reliable solution really shines.

1.2 Objective

Our main objective was to produce a product that is simple, effective, and reliable. We needed the SWISH Trainer's design to be simple as it would be used on court, and users most likely would not want a complex User Interface. We also needed our product to effectively relay stats so that it would actually help users quantify their shooting. Finally, we needed our product to be robust as ultimately it is an athletic branded device and would most likely be exposed to water, elements, or external trauma.

1.3 Sketch

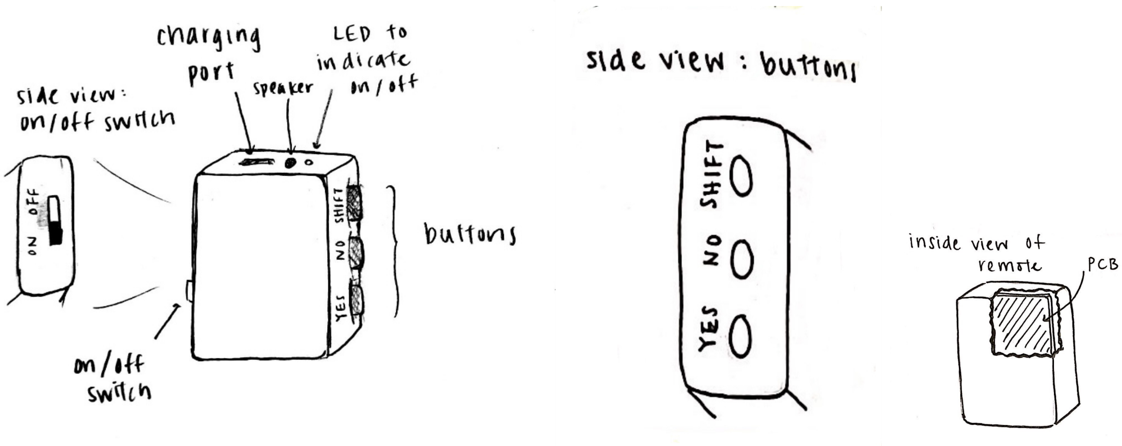

Figure 1: SWISH Trainer remote

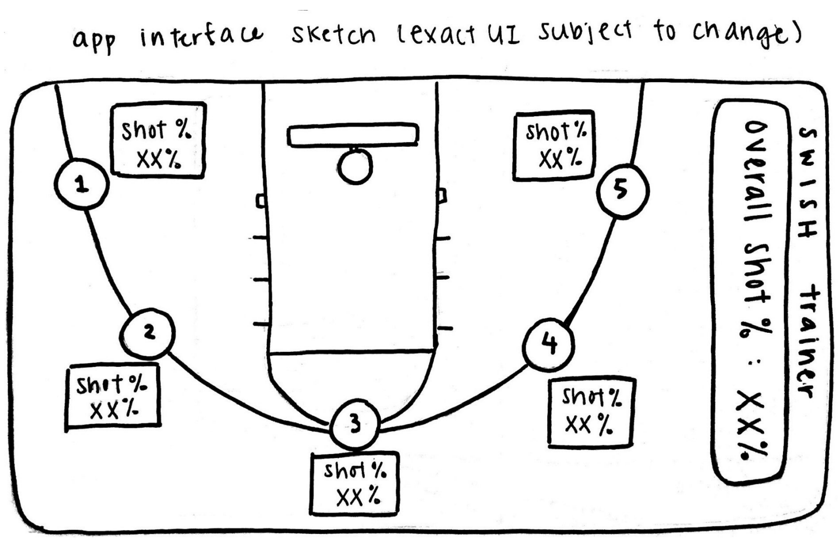

Figure 2: SWISH Trainer App Interface

2. Design

Figure 3 shows our final block diagram for the design. There are three main subsystems: battery management, data compilation, and external components.

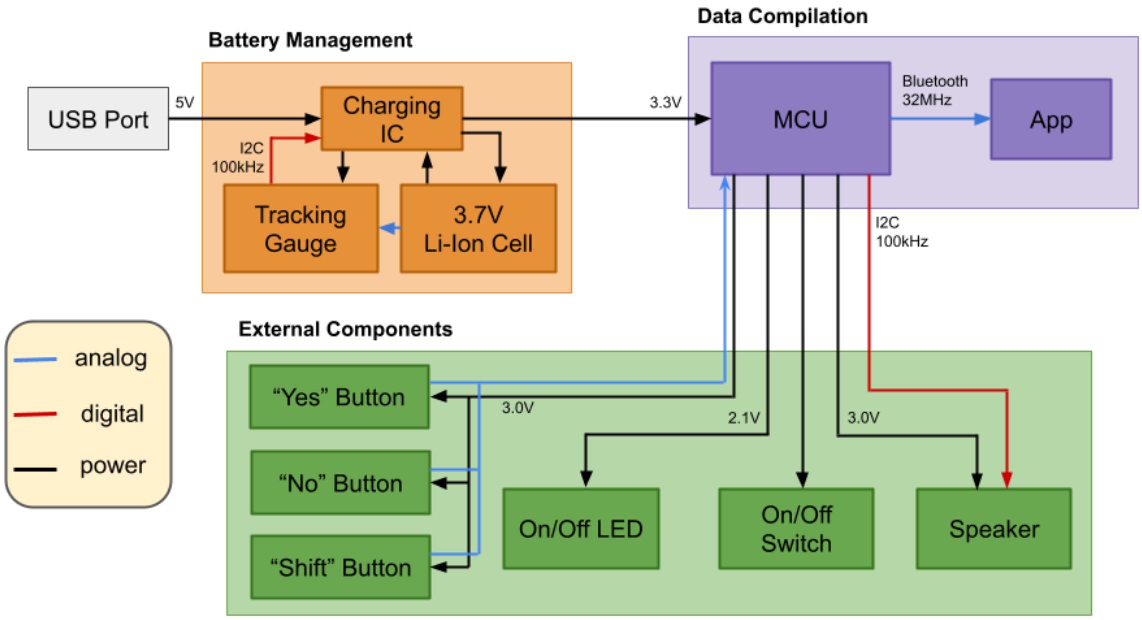

Figure 3: Full block diagram.

2.1 Design Procedure: Hardware

We first began the hardware design process by looking at several reference designs for wearable solutions by different companies to understand the basic structure and blocks needed for functionality. Based on those designs, we determined that we would have three main subsystems in which to group our hardware: battery management to support a rechargeable battery, data compilation to collect data and send it to the app, and external components that would collect data or inform the user of the status of the remote (i.e. on or off, charging, button pressed, etc). Throughout the design process, we made it a priority to select IC or parts that were readily available and could feasibly be soldered by hand in the lab.

2.1.1 Battery Management

Similar to the overall design of the block diagram, the design process for the battery management system also began with in-depth research on battery management solutions for wearable devices. The majority of the reference designs were much more complex than what we were looking for in our design, and in the end, we decided two ICs would be enough to implement charging: a battery charging IC and a tracking gauge IC.

Due to the small nature of wearable devices, a large majority of the reference designs we studied featured ICs that were incompatible with soldering by hand. Further research later provided a reference design by ST Microelectronics in which a battery charging IC and a tracking gauge IC were used together and came in packages with exposed pads instead of a ball grid array [5]. While we could have chosen the ICs with packages that we are able to be soldered by hand from separate designs, we prioritized selecting a set ICs from a preexisting design because it would provide more documentation for implementation that would ultimately make it easier to configure in the setting of our own design.

2.1.2 MCU and External Components

We needed an MCU with enough PINs as well as a Bluetooth module attached as our entire design relied on the MCU being able to connect our remote system to our app. We initially looked at several cypress MCUs but these weren't a good fit as many of them were 128 PINs and some were lacking Bluetooth capability. We eventually decided on the STM32 series, specifically the STM32WB30CEU5A. We felt that this was the best fit as this MCU had 48 PINs which is ideal for our project as well as a Bluetooth Low Energy (BLE) module that would enable connection to our app. Additionally, the supply voltage range of 2V~3.6V fits with our Power Management System (PMS). Additionally, we felt that the on-air data rate of 1Mbps would be sufficient for any data transmission we would need to do. We actually conducted 10 trials with multiple people in order to determine that the average time it takes to get a phone out of a pocket and open up an app was roughly 5 seconds. With this and our 1Mbps on-air data rate in mind, we determined a tolerance of 5 Mbps transmitted at a time between our remote and app to ensure that our information is relayed within a reasonable time so the user wouldn't have to wait too long.

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

The on-board time, as well as the app computation time, can be assumed to be near 0 [sec] so the only time that really matters is the remote-app transition time which is how we determined our tolerance of < 5 [sec].

2.2 Design Procedure: Software

We developed an IOS app to integrate with the remote that transmits data to the app via BLE integration from the MCU. We chose to create an IOS app over an Android app because to potentially mass scale the app, converting an IOS app to an Android app is much easier than converting an Android app to an IOS app. We chose to use the platform Xcode to develop our app using the programming language Swift to actually code to the functionality of the app. The Xcode platform allows for the user-friendly building of the user interface and allows for connections to Python connector files to integrate varying Bluetooth devices.

2.3 Design Details

See Appendix A for the full schematic, PCB, and app interface.

2.3.1 Battery Management

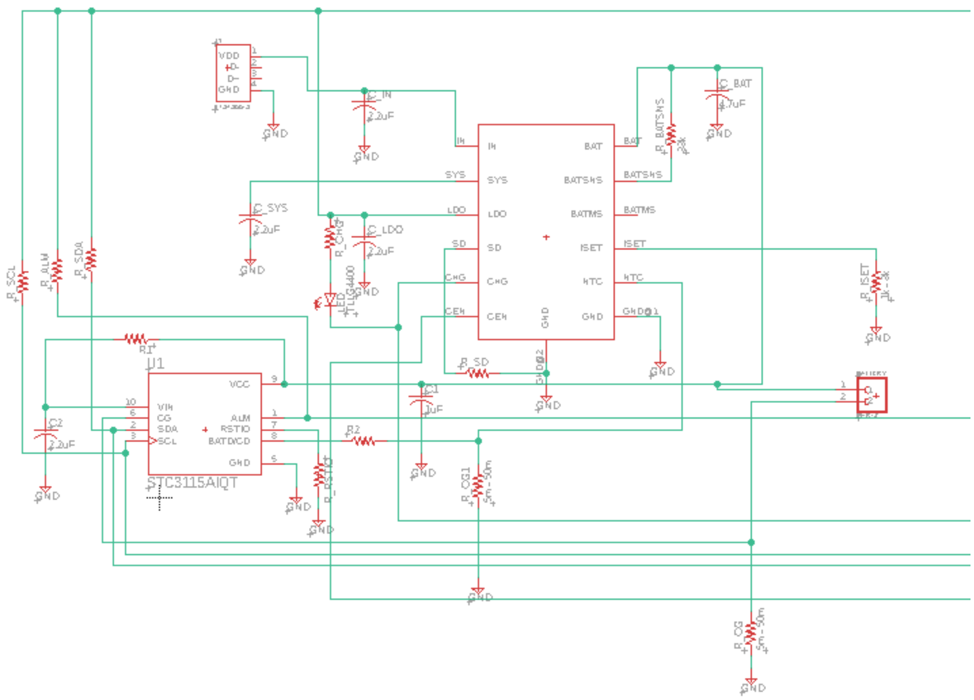

Figure 4: Battery Management Subsystem Schematic

The battery management system is used for charging and discharging. There are four main components within this subsystem (see Figure 4): the lithium-ion battery charging IC with LDO, the tracking gauge IC, the 3.7 V lithium-ion battery cell, and the USB port acting as the interface between the system and the charging source.

Lithium-Ion Battery Charger with LDO. The lithium-ion battery charger with LDO, STNS01 (see Figure 5), charges the lithium-ion battery by taking in a 5V supply voltage and outputting 3.3V through the LDO pin that powers the microcontroller. While connected to the 5V supply, the output at the pin is bypassed by the LDO and supplies 3.1V. If no input supply voltage is connected, the IC is powered by the battery [6].

The battery is charged using a constant-current constant-voltage (CC/CV) algorithm, where the battery is charged using a constant current (determined by the resistor connected to the ISET pin) until the battery reaches the 4.2V floating voltage. Once at the floating voltage, the charger switches to constant voltage to regulate battery voltage while decreasing the charging current [6]. When the battery cell is in conditions outside of the temperature range 0°C - 45°C, the charge enables the pin (CEN) to halt the charging process [6].

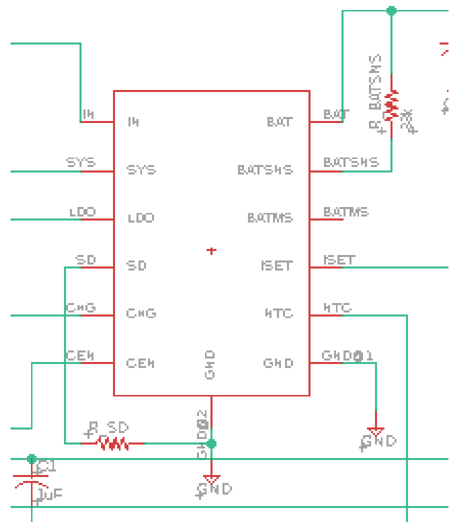

Figure 5: STNS01 in battery management subsystem schematic.

Tracking Gauge IC. The tracking gauge IC, STC3115 (see Figure 6), is a gas gauge that performs current sensing, temperature monitoring, and battery voltage monitoring to estimate the present charge state of the battery cell. There is also a low battery alarm (ALM) that outputs a signal to the microcontroller to notify the system of a low battery status [7].

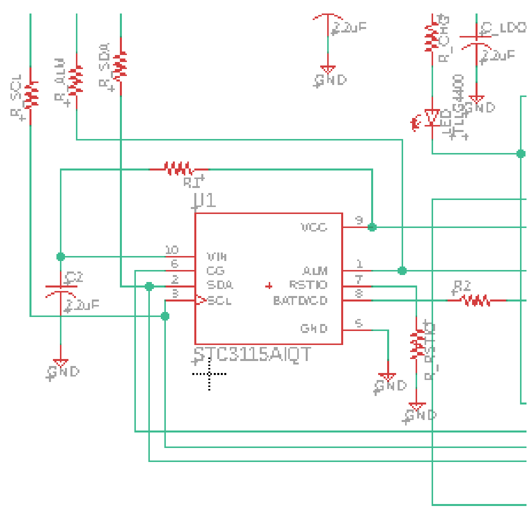

Figure 6: STC3115 in battery management subsystem schematic.

Lithium-Ion Battery. The 3.7 V lithium battery cell has a 150mAH capacity and can fully charge to a floating voltage of 4.2V. The battery charger and tracking gauge ICs are connected to the battery cell to closely monitor the voltage, current, and temperature conditions.

USB Port. The USB charging port is used to supply 5V to the battery charger IC. The data pins are not used because the port is purely for charging.

Connection to MCU. The output of one of the pins on the battery charging IC supplies 3.3V to the microcontroller. Additionally, the charging (CHG) and the charge enable (CEN) pins are connected to GPIO pins on the microcontroller to send signals.

The tracking gauge IC has the alarm (ALM) pin connected to the microcontroller to send an alarm signal in case of a low battery condition. The I^2C serial data (SDA) and I^2C serial clock (SCL) pins are also connected to the microcontroller for digital data transfer.

2.3.2 Hardware: MCU/External Components

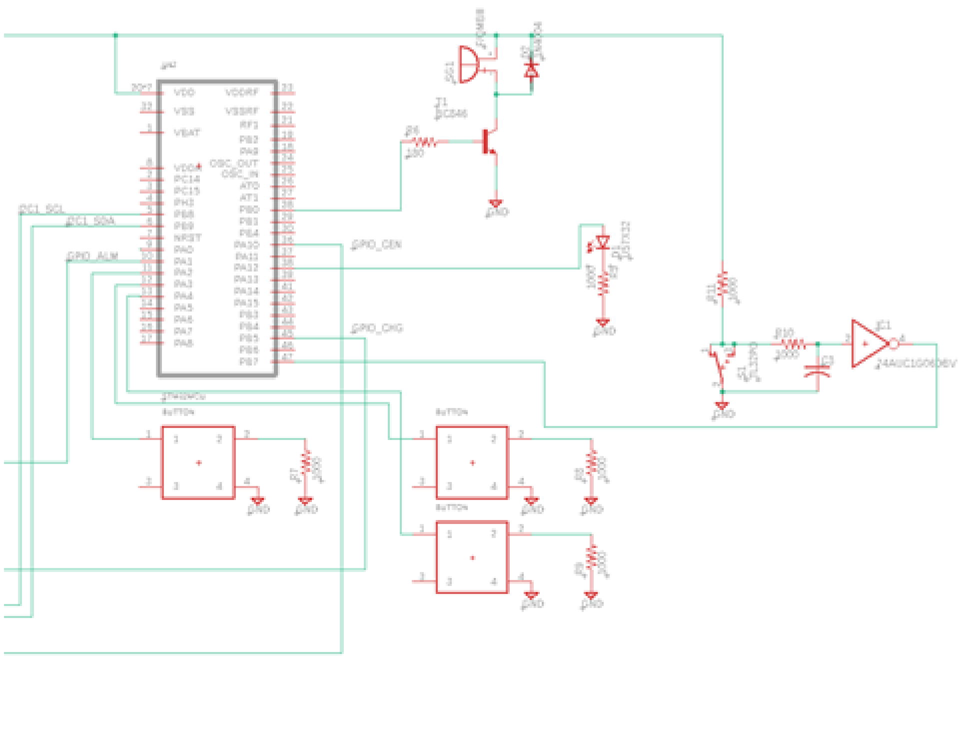

Figure 7: STC3115 in battery management subsystem schematic.

The MCU-External Components system has the MCU as the central control module and provides input/output connection to 4 smaller subsystems. These systems include the speaker amplification circuit, LED, Switch debouncing circuit, and the 3 buttons.

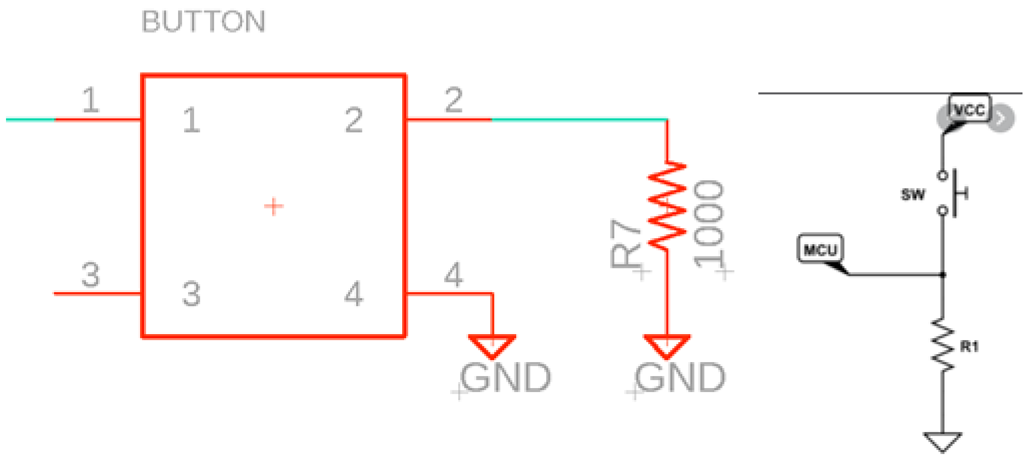

Figure 8: TL3300CF160Q Tactile Button Circuit

The Button circuit shown in figure 8 is a standard tactile button with a resistor attached to one of the pins while one pin is grounded and the other relays input information. The schematic on the right of Figure 3 displays how this circuit works, the button is pressed down which creates a short circuit, and when the button is not pressed this creates an open so no current/signal passes through.

The buttons each require a pull-down resistor to ensure that the pin gets a stable voltage by enabling the pin to be grounded in the absence of an input voltage.

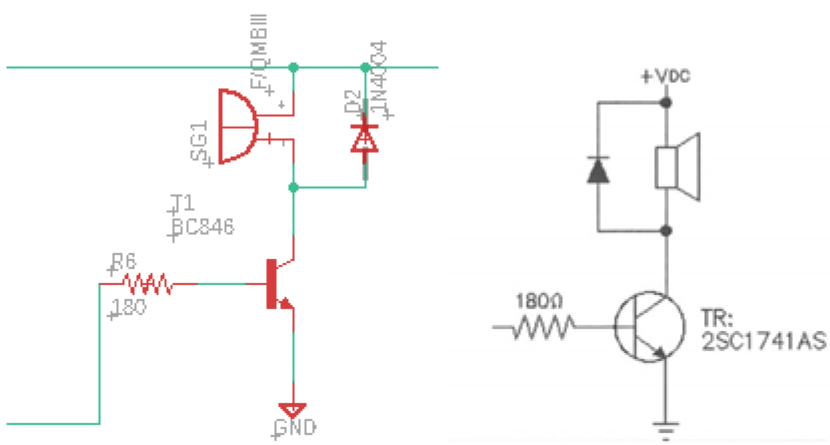

Figure 9: CEM-1203 Buzzer Circuit & Buzzer schematic from the datasheet

The buzzer circuit was stripped from the buzzer's datasheet. The datasheet specified an obsolete BJT, 2SC1741AS so we had to find a newer one that had equivalent characteristics. The parameters that we absolutely had to have were NPN, surface mounted, a V_{CE} of 50V (Collector to Emitter Voltage), a V_{CB} of 50V (Collector to Base Voltage), a V_{EB} of 5V (Emitter to Base Voltage), and a collector current I_C of 0.5A. This speaker amplification essentially increases the signal input and drops impedance in order to produce sound properly. The transistor that met all these conditions was the ROHM 2SD1484KT146Q transistor.

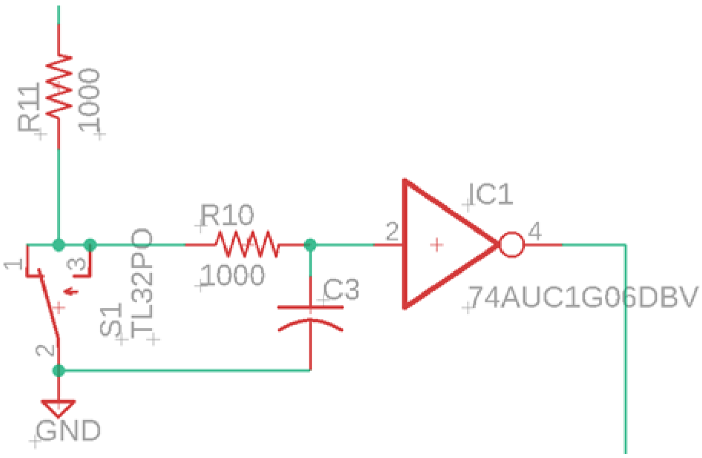

Figure 10: COM SPDT Slide Switch Debouncing Circuit

When it comes to switches and buttons, a brief mechanical contact within the nanosecond range can last in multiple signal pulses. With this in mind, we designed the following standard switch debouncing circuit in order to ensure that no double reading of inputs occurs. Additionally, we include an AUC1 Inverter IC in our circuit for signal inversion.

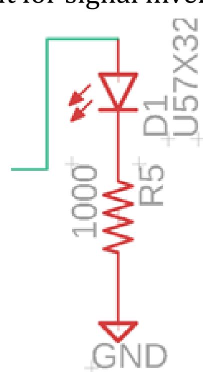

Figure 11: COM SPDT Slide Switch Debouncing Circuit

We ensured that our L314 Green LED was connected to a resistor to ensure that it does not burn out.

2.3.3 Software

Figure 19 depicts the UI interface of the app we developed for this project. It’s a standard, basic UI interface that allows for the user to press on the varying spots around the court and shows the percentage based on the spot that was selected. The app is written in Swift and essentially works by keeping track of which buttons are pressed on the remote. The app basically holds an empty fraction at the beginning of each spot. This edge case is programmed to showcase “none” on the overall shot percentage. When the yes button is pressed to signal a made shot, the counter on both the denominator and the numerator increment to create the new percentage that is then respectively displayed in overall shot percentage. If the no button is pressed which signals a missed shot, only the denominator is incremented and a newly calculated percentage shows up in the overall shot section. If the third shift button is pressed, the app pushes the calculation into the next shot set on the app. The prior spot's percentage is still saved and can be viewed by the user by pressing the prior spot such as 1 in the above image. When the shift button is pressed, the process restarts with the buttons until the shift button is pressed again to the third location. When the shift button is pressed and there are no more spots remaining to go to, the app is pushed into a state of finish which signifies that no more user input needs to read. The shift button then allows for the process to be reset from shot 1, and this icon will prompt a "are you sure" pop-up that confirms the decision of the user that they want to reset the process.

3. Verification

3.1 Functional Testing: Breadboard

While waiting for the PCB to arrive at the lab, we completed some breadboard testing for the battery charging IC and the external components.

3.1.1 Battery Management System

We verified the functionality of the battery charger IC portion of the battery management subsystem circuit in the schematic by building it out on a breadboard. The tracking gauge IC was not included in these initial breadboard tests because we were unable to acquire and surface mount to a through-hole adapter that fit the IC's dimensions.

The starting voltage of the battery was first measured to be 3.85V before it was connected to a modified design that excluded the gas gauge IC. Immediately, the problems with our design became apparent: the exposed pads on the ICs led to issues with soldering and the destruction or melting of those ICs, the LED was toggling on and off signifying the IC was in the incorrect state, and thus peripheral resistors were set up incorrectly. Eventually, after employing multiple soldering methods and trying different combinations for peripheral resistors based on an existing schematic, we were able to confirm that the IC was charging the battery. The battery was then allowed to charge for a few minutes (and the battery voltage was observed to be increasing on a voltmeter connected across the battery), disconnected from the circuit, and observed the voltage to stabilize at the end battery voltage 3.96V.

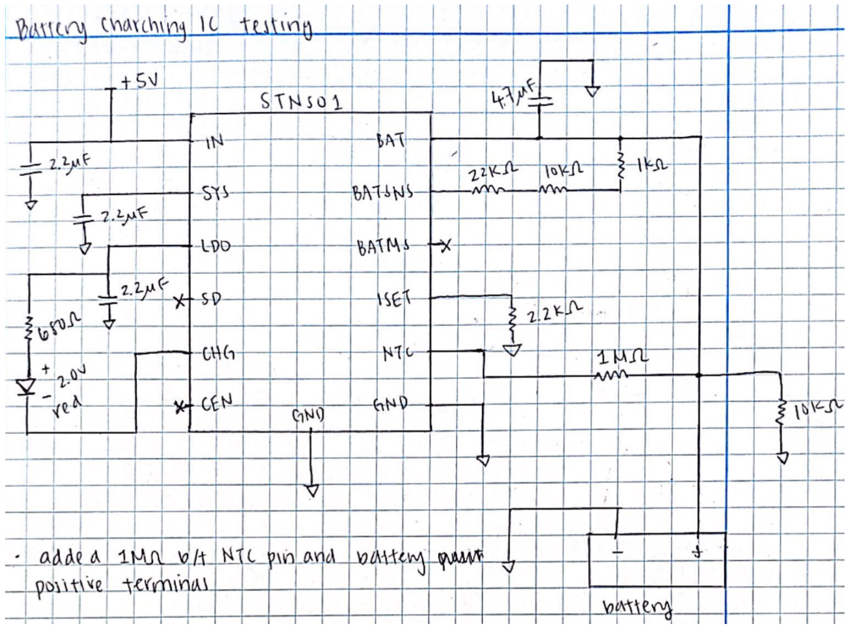

Figure 12: Circuit with successful charging of the battery by STNS01.

The STNS01 has the option of selecting a fast charge current ranging from 15mA to 200mA depending on the value of the resistor connected to the ISET pin of the battery charging IC. We performed tests in the lab to evaluate the differences in charging time for two values on the lower end of the range (1kΩ - 13kΩ). The charging current is determined by an equation provided in the datasheet [6]:

$$I_{FAST} = \frac{V_{ISET}}{R_{ISET}} \times K \tag{3.1.1}$$

Where V_{ISET} = 1V and K = 200. R_{ISET} was initially set to 1kΩ (the greatest charging current, 200mA) while testing for the functionality of the IC.

	Resistor: 1kΩ			**Resistor: 2.2kΩ**
Time (min:sec)	**Battery Voltage**		**Time (min:sec)**	**Battery Voltage**
0:00	3.928		0:00	3.928
1:00	4.020		1:00	3.933
2:00	4.042		2:00	3.940
3:00	4.077		3:00	3.942
4:00	4.152		4:00	3.948
.	.		5:00	3.952

⋮	⋮	9:00	4.013
9:32	4.201	...	...
		10:00	4.017

Table 1: Voltage of battery while charging for different $R_{ISET.}$

After obtaining the data shown in Table 1, we decided to calibrate the IC using the fastest charging for optimal user experience. While a slower charging time will prolong the quality of the battery over time, this device is intended for planned and unplanned sessions and it is likely that people will not want to continue using the remote if it takes a long time to charge.

3.1.2 External Components

LED, Switch, and Push Button. The functionality of both the LED that will be used to indicate the remote is on and the switch that turns the system on was tested using a small circuit developed on a breadboard separate from our overall design.

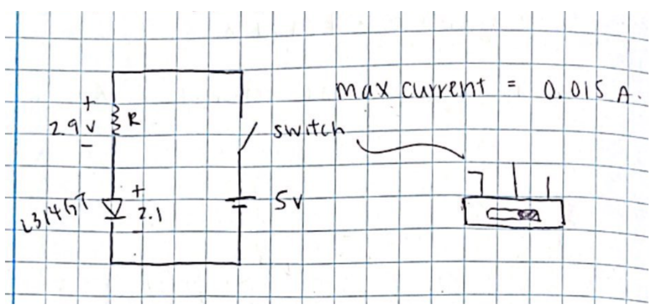

Figure 13: Circuit used to LED, switch, and push button.

The value of the resistance needed based on the maximum current was calculated and rounded up to ensure that the LED would not be damaged. After flipping the switch, the LED lit up, confirming that both the LED and the switch were functioning correctly.

To test the push button used to record successful or unsuccessful shots, the switch was swapped with the push button. The LED lit up when the button was pressed, thus confirming that the push button was also functioning correctly.

Speaker. The speaker needed a small circuit to set it up correctly, but we, unfortunately, did not have through-hole versions of the transistor and diode needed to complete the circuit. We tried replacing the components with a transistor and diode previously used in other labs, but the speaker did not produce any sound when connected to the completed circuit.

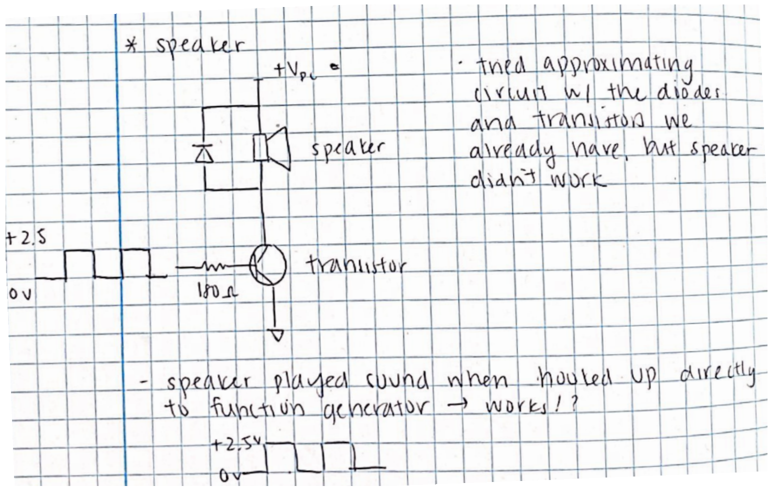

Figure 14: Notes from testing the speaker.

The speaker was then directly connected to the function generator and produced a sound when the function was "high" at +2.5V and no sound when the function was "low" at 0V.

3.2 Functional Testing: Simulations

3.2.1 Debouncing Circuit

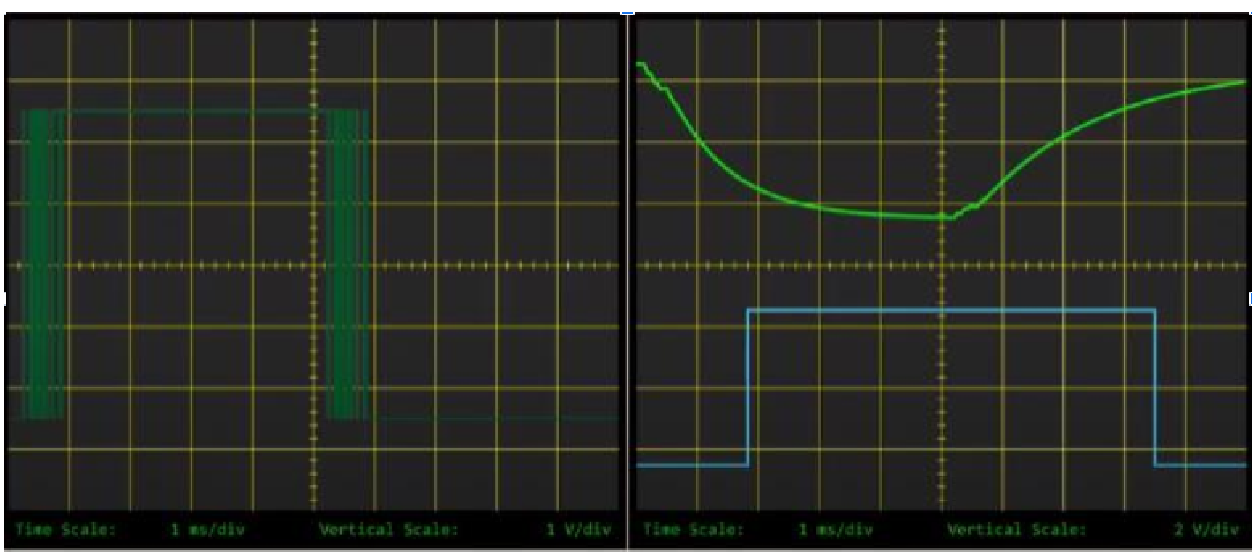

Figure 15: Debouncing Waveform - before and after

As mentioned before, with devices that rely on mechanical contacts such as buttons and switches, multiple input signals can be faultily read. Seeing as a mechanical contact can last for multiple nanoseconds, which is all that is needed to read multiple pulses, it is important to debounce.
On the left half of figure 15, you can see the result of a lack of debouncing. There are multiple pulse signals on the falling edge and the rising edge of the mechanical contact. This is basically the

“bounce” of the signal. To prevent this, we added our debouncing circuit and took readings afterward. As you can see on the right half, the pulses are now gone on the rising and falling edges and the signal is somewhat smooth.

4. Cost

4.1 Prototype Cost

Throughout the semester, we kept detailed records of the items purchased through the ECE department to remain within the $100 budget. We purchased extra components for testing and reserve in case anything went wrong in the lab. Figure 16 details the purchase orders submitted to the department. Table 2 includes the fixed labor cost for the semester, yielding a total cost of $96,086.31.

PROTOTYPING COST						
Subsystem	**Device**	**Manufacturer**	**Package**	**Description**	**Purchase Quantity**	**Cost**
Battery Management	STNS01	ST	DFN12	battery charger IC	4	$ 11.90
Battery Management	STC3115	ST	DFN10	tracking gauge IC	4	$ 11.16
Battery Management	1317	Adafruit		3.7V lithium-ion battery, 150mAh	2	$ 6.80
Battery Management	CU01SAV0S00	Cvilux USA		USB charging port	3	$ 1.26
Control Module	STM32WB30CEU5A	ST		microcontroller	3	$ 13.50
External Components	L314GT			3mm yellow green LED	3	$ 0.75
External Components	TL3300CF160Q	Lamb Industries		12mm tactile buttons	6	$ 3.60
External Components	slide switch			SPDT slide switch	2	$ 1.50
External Components	CEM-1203(42)	Cui Inc		speaker	2	$ 3.90
Control Module	2SD1484KT146Q	ROHM		TRANS NPN 50V 0.5A SOT-346	2	$ 0.72
Battery Management	IPC0070	Chip Quik		DFN-12 TO DIP-16 SMT ADAPTER	1	$ 5.67
Battery Management	IPC0088	Chip Quik		DFN-10 TO DIP-10 SMT ADAPTER	1	$ 4.77
Control Module	204-0014-01	Schmartboard, Inc			1	$ 5.40
External Components	RR0816P-102-D	Susumu	x0603	1kohm resistor	10	$ 0.77
External Components	1N4004-TP	Micro Commercial Co		diode, 1V	3	$ 0.36
External Components	CML0805Y5V104ZT50V	Stackpole Electronics	x0805	0.1 uF capacitor	3	$ 0.27
Physical Case				Waterproof Case	1	$ 9.98
					TOTAL	**$ 82.31**

Table 2: Prototype cost.

Type	**Description**	**Cost**
Fixed	$40/hr 10 hr/day 5 days/week 16 weeks 3 people	$96,000.00
Variable (prototype)	PCB	$4.00
Variable (prototype)	Components for PCB, see Figure 16	$82.31
TOTAL		**$96,086.31**

Table 3: Total prototype cost.

4.2 Mass Production Cost

If the production of the remote were to be scaled up to a minimum of 10,000 units, the cost of the remote goes down significantly. Table 4 shows the bulk order unit costs from Digikey and the actual quantity of components (ICs, resistors, capacitors, etc) necessary for the design.

MASS PRODUCTION COST							
Subsystem	**Device**	**Manufacturer**	**Package**	**Description**	**Quantity in Design**	**Unit Bulk Cost**	**Subtotal**
Battery Management	STNS01	ST	DFN12	battery charger IC	1	$ 1.05	$ 1.05
Battery Management	STC3115	ST	DFN10	tracking gauge IC	1	$ 1.73	$ 1.73
Battery Management	1317	Adafruit		3.7V lithium-ion battery, 150mAh	1	$ 5.95	$ 5.95
Battery Management	CU01SAV0S00	Cvilux USA		USB charging port	1	$ 0.23	$ 0.23
Control Module	STM32WB30CEU5A	ST		microcontroller	1	$ 2.75	$ 2.75
External Components	L314GT			3mm yellow green LED	1	$ 0.06	$ 0.06
External Components	TL3300CF160Q	Lamb Industries		12mm tactile buttons	3	$ 0.60	$ 1.80
External Components	slide switch			SPDT slide switch	1	$ 0.75	$ 0.75
External Components	CEM-1203(42)	Cui Inc		speaker	1	$ 1.95	$ 1.95
Control Module	2SD1484KT146Q	ROHM		TRANS NPN 50V 0.5A SOT-346	1	$ 0.07	$ 0.07
External Components	RR0816P-102-D	Susumu	x0603	1kohm resistor	6	$ 0.01	$ 0.07
External Components	1N4004-TP	Micro Commercial Co		diode, 1V	1	$ 0.02	$ 0.02
External Components	CML0805Y5V104ZT50V	Stackpole Electronics	x0805	0.1 uF capacitor	1	$ 0.02	$ 0.02
Physical Case				Waterproof Case	1	$ 9.98	$ 9.98
Battery Management	L513ED	American Opto Plus LED		red LED	1	$ 0.05	$ 0.05
Battery Management	resistors, capacitors			other passive components we didn't purchase but are needed for the design	25	$ 0.03	$ 0.75
						TOTAL	**$ 27.24**

Table 4: Mass production cost using prototype bill of materials.

With access to industrial equipment, the cost of one remote could be further reduced as many of the higher-cost items (i.e. the ICs and physical case) can be used in smaller packages (such as the ball grid array). Additionally, we would not be restricted to the specific ICs and would have the option to purchase from manufacturers at lower prices. Overall, we are confident that the total cost of the remote can be reduced to ~$13 per remote. See Appendix B for a mass production real-world estimate.

4. Conclusions

4.1 Timeline

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	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

Table 5: Timeline

4.2 Ethics and Safety

There are a couple of safety concerns with our project. Our primary safety concern is the use of a lithium-ion battery. 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 [8].

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. During thermal runaway, the excess byproducts released from this process may ignite or cause other harmful side effects [8].

In order to ensure that lithium-ion cell runaway or other damage does not occur, we made 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” [9].

In order to prevent water damage in the form of short circuits, we adhered to IP65 waterproofing standards. The case will be able to withstand water jets from various directions without resulting in damage to the circuit within [10].

In terms of privacy laws with respect to the app, we secured the data to ensure that 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” [9]. If the app ever was mass released on the actual apple store, we would have to write encrypted code and abide by the privacy laws set in place by the app store at the current time. Additionally, the app would have to adhere to the 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" [9]. Throughout the development of the app and the project as a whole we kept in mind 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[9]." The overall idea of this project is the emphasis on improving 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. Citations

[1] 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].

[2] 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].

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

[4] 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].

[5] "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].

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

[7] "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].

[8] "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].

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

[10] 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/

[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].

[14] 2021. *Multiprotocol wireless 32-bit MCU Arm®-based Cortex®-M4 with FPU, Bluetooth® 5 or 802.15.4 radio solution*. [pdf] Available at: <https://www.st.com/resource/en/datasheet/stm32wb30ce.pdf> [Accessed 27 February 2021].

[15] 2010. *TL3300CF160Q*. [pdf] Available at: <https://cdn.sparkfun.com/datasheets/Components/Switches/N301102.pdf> [Accessed 2 March 2021].

[16] 2009. *SPDT Slide Switch*. [pdf] Available at: <http://cdn.sparkfun.com/datasheets/Components/Switches/1193302209860_SS12D01-A.gif> [Accessed 5 April 2021].

[17] 2006. *Magnetic Buzzer CEM-1203*. [pdf] Available at: <http://cdn.sparkfun.com/datasheets/Components/General/cem-1203-42-.pdf> [Accessed 5 April 2021].

[18] 2014. *American Opto Plus LED Corp. L314GT 3mm Green Lamp LED*. [pdf] Available at: <https://www.aopled.com/AOP_PDFs/L314GT.pdf> [Accessed 5 April 2021].

[19] *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].

[20] “STEVAL-WESU1_Schematic,” ST Microelectronics. [Online]. Available: https://www.st.com/content/ccc/resource/technical/layouts_and_diagrams/schematic_pack/group0/f7/9c/ba/9b/15/8a/4f/3f/STEVAL-WESU1%20schematic/files/steval-wesu1_schematic.pdf/jcr:content/translations/en.steval-wesu1_schematic.pdf. [Accessed: 2-Apr-2021]

Appendix A

Full Schematic

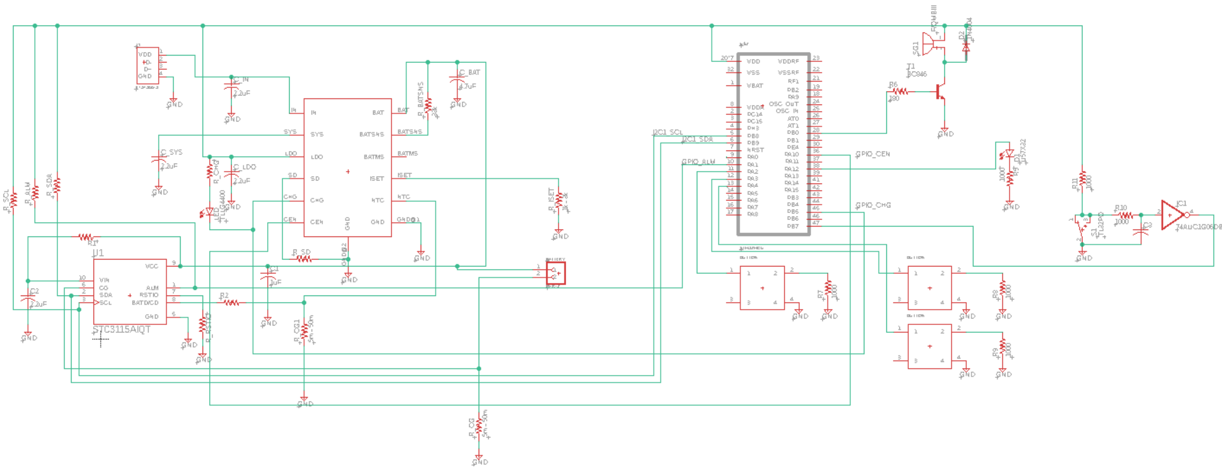

Figure 17: Full schematic

Full PCB

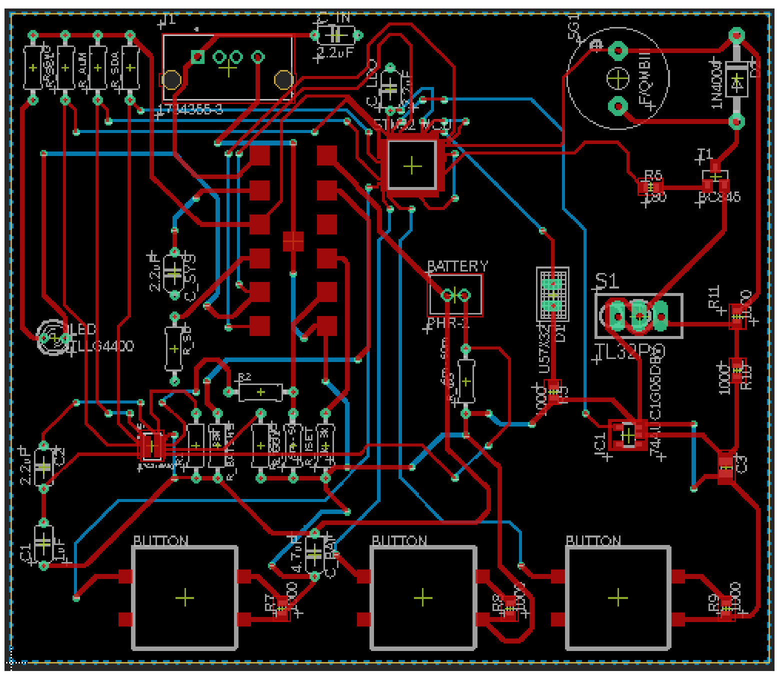

Figure 18: Full PCB

Full App Interface

Figure 19: App Interface

Appendix B

Mass Production Real World Estimate

The average user would spend at most $30 for the remote, with the app assumed to be free. If we were to take all costs into account with a 40% profit margin, we can estimate the maximum BOM, manufacturing, and other labor costs of producing 10,000 units if we were to charge $30 per unit. Table 4 shows the details of the financial breakdown.

GROSS REVENUE	
$30/unit for 10,000 units	$300,000
COST	
Bill of Materials (majority, estimate 75% of Cost Total)	$135,000
Manufacturing and Other Labor (estimate 25% of Cost Total)	$45,000
Cost Total	$180,000
PROFIT	

Profit Total (estimate 40% of Total)	$120,000

Table 6: Financial Breakdown for industrial production of 10,000 units.

Based on the estimates in Table 5, the cost of one remote must be reduced to $13.50 in order to generate a 40% profit margin for a remote that retails $30 each.