Voice Activated Scorekeeper

ECE 445: Design Document

Team 31: Jason Hwang, Christopher Kalebich, Allan Zou

TA: Enliang Li

Fall 2019

Table of Contents

1. Introduction

1.1. Problem & Solution Overview

Pickup games are very casual and quickly organized games with either strangers or a group of friends. Due to its casual nature, there is no referee or scorekeeping system; rather, the players self-officiate and keep score themselves. Keeping track of the score can be especially difficult for games such as volleyball, where the games are played to twenty-five points. In addition, long rallies where players are focused on winning the current point result in players forgetting the score. As a result, players often ask what the score is after most points.

Our goal is to eliminate this problem by creating a visual scorekeeper that will allow players to know the score at a glance. In order to facilitate an easy user interface, the scorekeeper will be voice activated, where specific keywords are recognized in order to increment the score for each team. This allows for players to quickly update the score without affecting the flow of the game.

1.2. Background

A quick search through both the Google Play store and Apple App store reveals a multitude of scoreboard apps with many even tailored to volleyball specifically. These apps are often either very simple: keeping track of the score and time [1], or very complex: keeping track of timeouts, substitutions, rotations, and more, in addition to the score and time [2]. For a casual game of volleyball, the latter is much too complex. Instead, players are more likely to use the simpler application. While these simpler applications are almost perfect for casual games, they do not provide a quick and effective means of updating the score. Because they require touch-input, a separate person, or a player must manage the device; neither option being ideal as there may not be more than 12 players.

1.3. Visual Aid

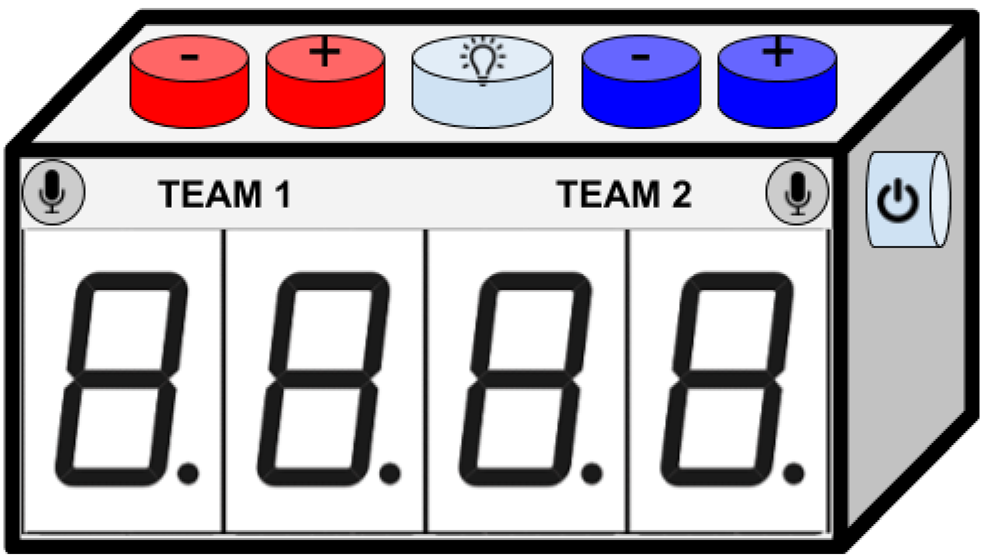

Figure 1: Physical Representation of the Device

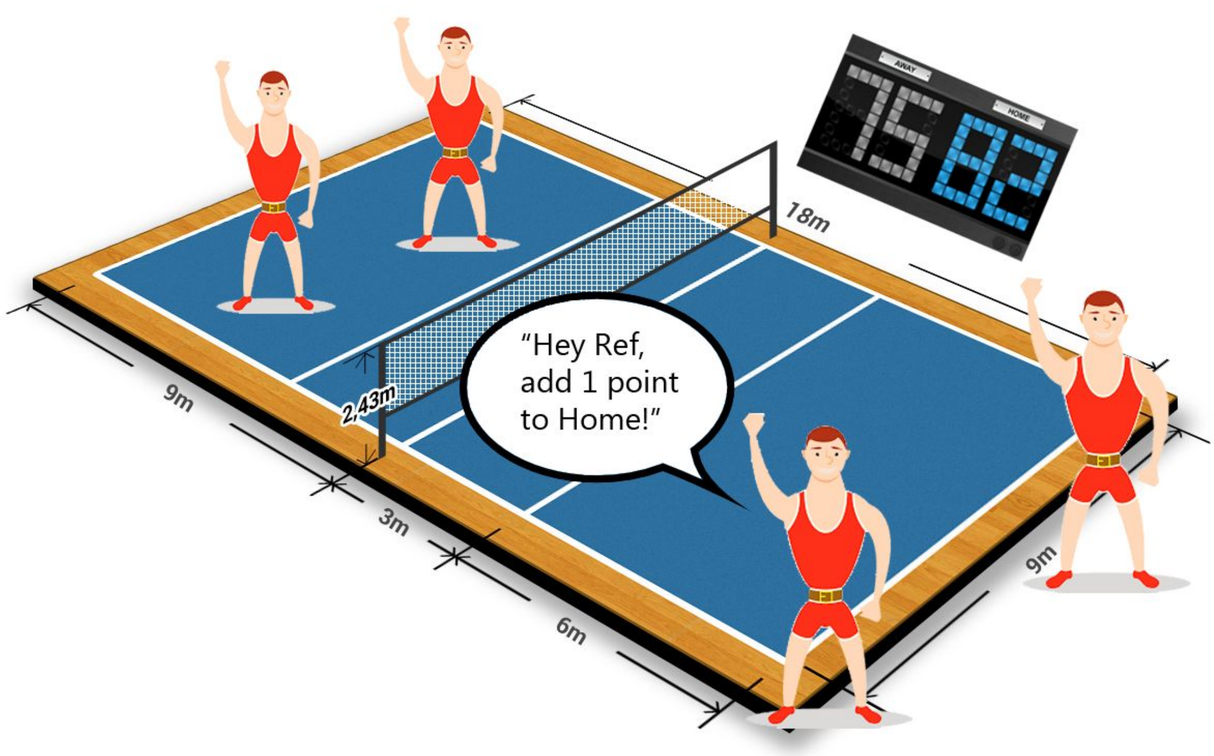

Figure 2: Volleyball Scorekeeper Main Usage

1.4. High-level requirements list

- Score correctly changes through keywords spoken by a user with at least 85% accuracy.
- Microphone should pick up audio commands ≤ 10 ft from the user with a background noise of 75 ± 5% dB.
- Battery should be rechargeable, lasting ≥ 4 hours.

2. Design

The block diagram shown in Figure 3 contains three subsystems: power, control, and user interface. The power subsystem allows the battery to be recharged safely and in turn power the rest of the system for at least 4 hrs when supplying 3.3V. The seven-segment displays in the UI output subsystem must be large enough to be seen from the service line 30ft away. Since they will be larger than standard seven segment displays, the system utilizes a boost converter to increase the voltage from the voltage regulator in order to properly power the displays. Finally, the UI input subsystem contains a microphone which allows the system to capture the keywords spoken by a user and pass it to the microcontroller for processing and determining what action to perform. The microphone should still be able to receive user commands in an environment where the background noise is 75 db ± 5%. We arrived at this value by measuring the noise level of a gym while players were playing volleyball and finding the average noise level over a three-minute time frame. The graphs of the three time frames used are seen in Figure 3.

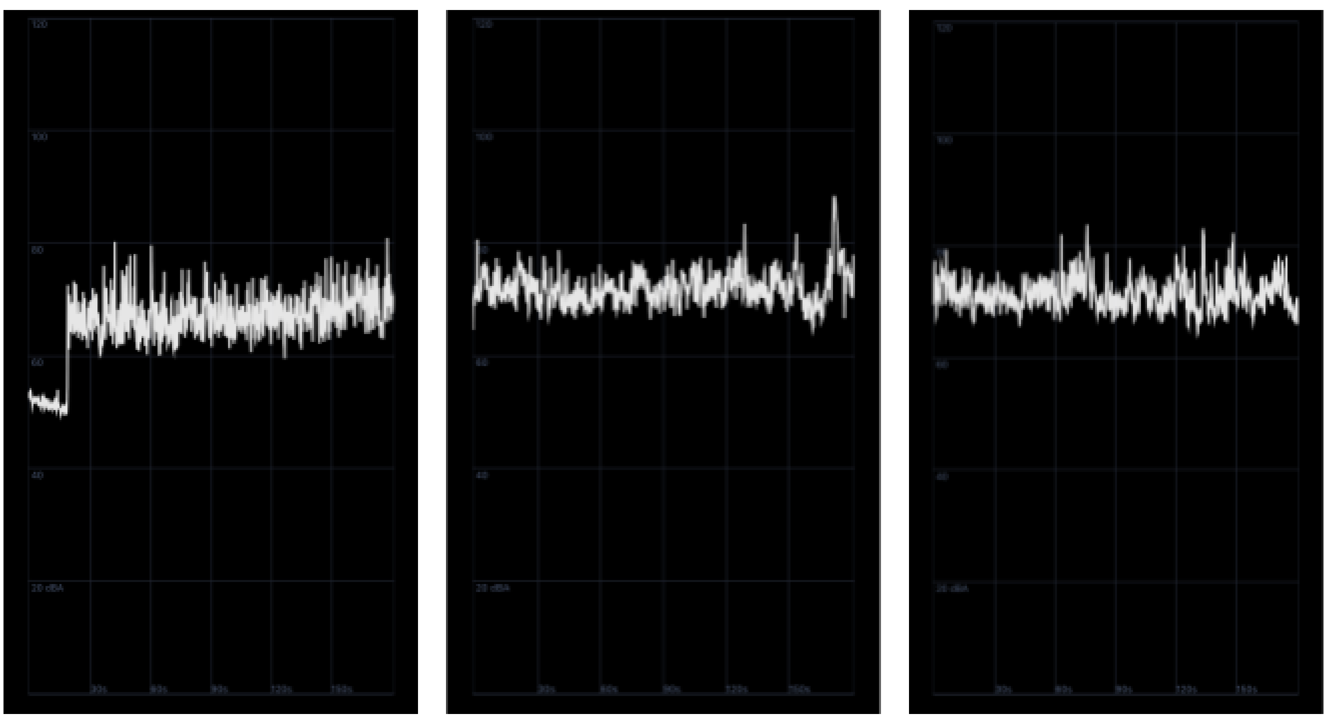

Figure 3. Three noise level graphs of 3-minute time frames (dB)

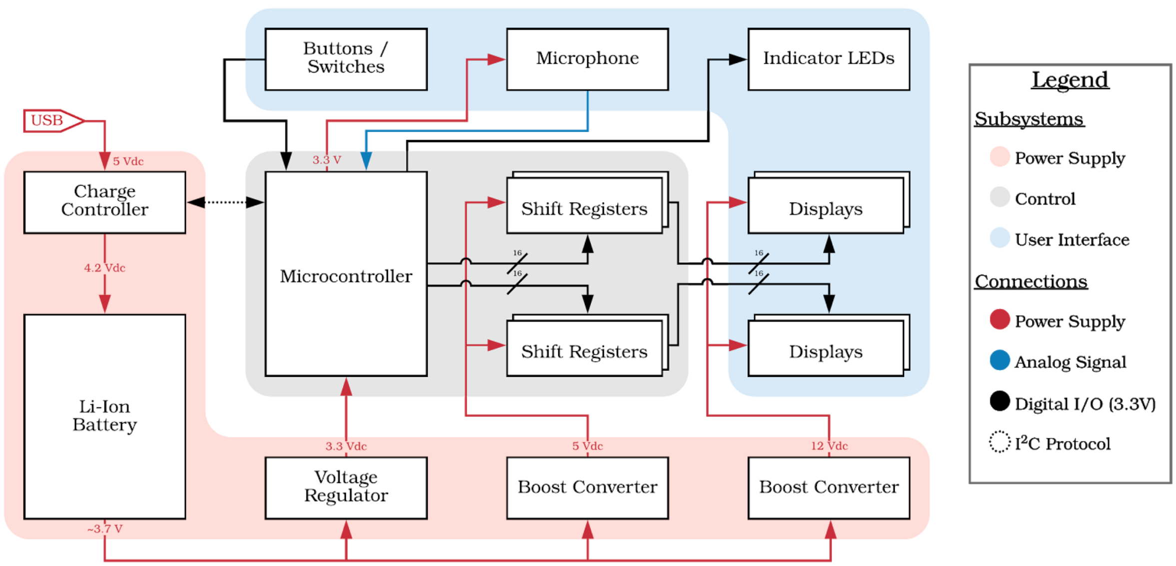

Figure 4. Block Diagram

2.1. Control Subsystem

The control system consists of the following components: microcontroller, seven-segment display driver. As a whole, this subsystem is responsible for processing the functionality of our device correctly. This subsystem will interact will all other subsystems and receive/deliver data from the other subsystems. After the software system is correctly implemented, it will process and execute based on the software provided. This subsystem will satisfy the high-level requirement: Score must be correctly changed through keywords spoken by a user.

2.1.1. Microcontroller

The microcontroller, a STM32F401CCU7, will handle a variety of tasks. First, it will receive input from either the buttons, or the microphone. If input is received from the microphone, the MCU will transform the analog signal to a digital signal. It then feeds the newly transformed signal into the Keyword Spotting (KWS) model stored in its 256kb flash memory. Depending on the keyword detected by the model, the MCU can turn on the indicator light to show users it is listening or update the game score. If the input is received from the buttons, the MCU can immediately update the score based on which button it received a signal from. In order to visually update the score, the MCU will communicate with the seven-segment display driver via SPI. Finally, the MCU will also receive input from the battery management IC to show users the charge status of the battery.

This microcontroller was chosen for a few reasons. First, it is an ARM Cortex M-4 based MCU. The Cortex M-4 contains a DSP extension to ARM's Thumb instruction set with additional instructions which accelerate DSP algorithms. In addition to faster DSP algorithms, by using an MCU with a Cortex M-4, is able to reduce manufacturing cost, development cost, and system complexity by removing the need for a separate DSP unit. This enables our system to rapidly modify the signal from the microphone so that it can be used as input for the KWS model. Secondly, this MCU has 256kb of flash memory which is necessary to store our KWS model. Deployment of a KWS application onto a Cortex M-7 STM32F746G-DISCO development board used ~70kb of memory[3], so we estimate that our application will use around the same amount of memory, while allowing us leeway to increase the size of the model.

The microcontroller will be programmed using a NUCLEO-F401RE development board.

Requirements	**Verification**
1. Flash memory is ≥ 100kb	1. a. Connect MCU to PC with an ST-Link connector b. Compile and load a test project with 100kb of data into flash memory using Arm Mbed IDE c. Connect MCU to PC with a USB-to-Serial Converter d. Using a terminal (ie. PuTTY), connect to the board on the associated COM port e. Transmit data loaded in flash memory from the MCU using USART f. Verify that data received matches data loaded
2. Must communicate over I2C	2. a. Connect MCU to a chip which can generate data and communicate over I2C(ie. RTC, Temperature sensor, etc.) b. Connect MCU to PC with a USB-to-Serial Converter c. Using a terminal (ie. PuTTY), connect to the board on the associated COM port d. Transmit data received over I2C from sensor to terminal using USART e. Verify data received is data that can be generated by the chip

2.1.2. 8-Bit Shift Registers

The shift registers receive 8 bits of data from the microcontroller and store the data which will be output to the seven segment displays.

Requirements	Verification
1. Must have Serial in Parallel Out	1. a. Feed 4 bits of data into the register through serial in b. Read data from parallel out on 4th clock cycle c. Verify data read is same as data entered
2. Each drain pin must be capable of sinking at least 140mA	2. a. Connect device to 5V power b. Insert serial data using microcontroller dev board c. Drive an output drain pin low with a 40Ω, 5V load attached d. Measure drain current with multimeter

2.2. User Interface Subsystem

The user interface subsystem consists of the following components: two microphones, seven-segment displays, indicator lights, buttons/switches. As a whole, this subsystem is responsible for interacting with our users. The microphone will allow users to speak to the device, the seven-segment displays will show the score, the indicator lights will display when speech recognition is activated, and the buttons/switches will allow the user to manually change the score if required.

2.2.1. Microphone

The microphone will pick up the keywords spoken by the user. The system will use an Adafruit MAX9814 microphone amplifier with an attached microphone. This chip comes with automatic gain control which enables our system to pick up audio from a range of distances without having to manually adjust the gain.

Requirements	Verification
1. Must pick up audio from ≤ 10 ft away	1. a. Drive chip at 3mA b. Connect chip output to an oscilloscope c. Measure 10 ft from microphone d. Play music/speak and verify that the signal is appearing on the oscilloscope

2.2.2. Seven-Segment Displays

The seven-segment display will indicate the current scores of two teams.

Requirements	Verification
1. Each segment must illuminate when applied with 12V and drive at a maximum of 140mA	1. a. Supply display with 12V and drive current of 140mA b. Ensure segments are illuminated

2.2.3. Indicator LEDs

The LEDs should light up when the microcontroller detects the keyword in order to give the user visual feedback that the device is listening.

Requirements	Verification
1. Must be visible from 10 ft away with drive current of 20mA	1. a. Measure 10ft from LED b. Deliver 20mA to the LED c. Ensure it is clear that LED is on

2.2.4. Buttons/Switches

Our device will have an increment & decrement button for both of the team scores to manually change the score value. The clear buttons will be used to reset the score to 0.

Requirements	Verification
1. Buttons click when pressed	1. a. Press button and ensure a click is felt

2.3. Power Subsystem

When connected, power will flow from the power supply to our prospective loads, however in order to design a proper supply, we must first select and examine all of the loads in the system. Using this top-down approach, we can then sum the components' maximum currents and design the power supply. As a result, the required voltage levels and maximum current draw are the two main specifications to be considered.

Our project will run on a potpourri of voltages, as our microcontroller runs on 3.3V; the shift registers, LEDs, and the microphone amp run on 5V; and our seven-segment displays run on 12V. In addition, because we know that we want to use a lithium-ion battery which has an output voltage of around 3.7V, our power supply will need to use a variety of regulators and boost converters.

2.3.1. Charge Controller

The charge controller will supply the battery with a constant 4.2 V supply. More importantly, the controller will monitor the input charge current and stop charging once the current drops below 3% of the battery's rated current.

Requirements	Verification
1. Must provide the battery with a constant charging voltage of 4.2 V.	1. a. Properly wire the chip with attached battery b. Measure the charging voltage across the battery with an oscilloscope
2. Must stop charging the battery once the charging current drops below 3% of the battery's rated current.	2. a. While charging, measure the battery input current with a multimeter

2.3.2. Battery

A 3.7 V 6600mAh lithium-ion battery.

Requirements	Verification
1. Must provide a nominal output voltage of 3.7 V ± 1%	1. a. Connect a simple load to the battery to drive 1A b. Measure the voltage across the load using an oscilloscope
2. Must provide a continuous discharge current of at least 1.5A (.23C) for four hours	2. a. Connect a load of 2.5 Ohms b. Measure the load current with a multimeter every ten minutes

2.3.3. 3.3 V Voltage Regulator

The voltage regulator will output 3.3VDC while receiving variable voltage input from the battery.

Requirements	Verification
1. Must supply 3.3V ± 1% with a varied input voltage range of 3V to 5V	1. a. Connect output to an oscilloscope b. Input a range of DC voltages from 3V to 5V c. Monitor the output voltage with an oscilloscope
2. Ensure the chip can drive up to 2A	2. a. Vary the load across the output b. Monitor the load current using a multimeter

2.3.4. 5V Boost Converter

The voltage regulator will output 5VDC while receiving variable voltage input from the battery.

Requirements	Verification
1. Must supply 5V ± 5% with a varied input voltage range of 3V to 5V	1. a. Connect output to an oscilloscope b. Input a range of DC voltages from 3V to 5V c. Monitor the output with an oscilloscope
2. Ensure the module can drive up to 3A	2. a. Vary the load across the output b. Monitor the load current using a multimeter

2.3.4. 12V Boost Converter

The voltage regulator will output 12VDC while receiving variable voltage input from the battery.

Requirements	Verification
1. Must supply 12V ± 5% with a varied input voltage range of 3V to 5V	1. a. Connect output to an oscilloscope b. Input a range of DC voltages

	from 3V to 5V c. Monitor the output with an oscilloscope
2. Ensure the module can drive up to 3A	2. a. Vary the load across the output b. Monitor the load current using a multimeter

2.4. Software

This system is a seperate part of the control subsystem. Once the software has been correctly implemented and tested, it will be uploaded to the microcontroller and determine how the microcontroller functions. As a whole, it will be responsible for speech recognition and changing the score based on the keywords, as well as updating the score using the buttons/switches. This subsystem will satisfy the high-level requirement: Score must be correctly changed through keywords spoken by a user.

2.4.1. Speech Recognition State Control

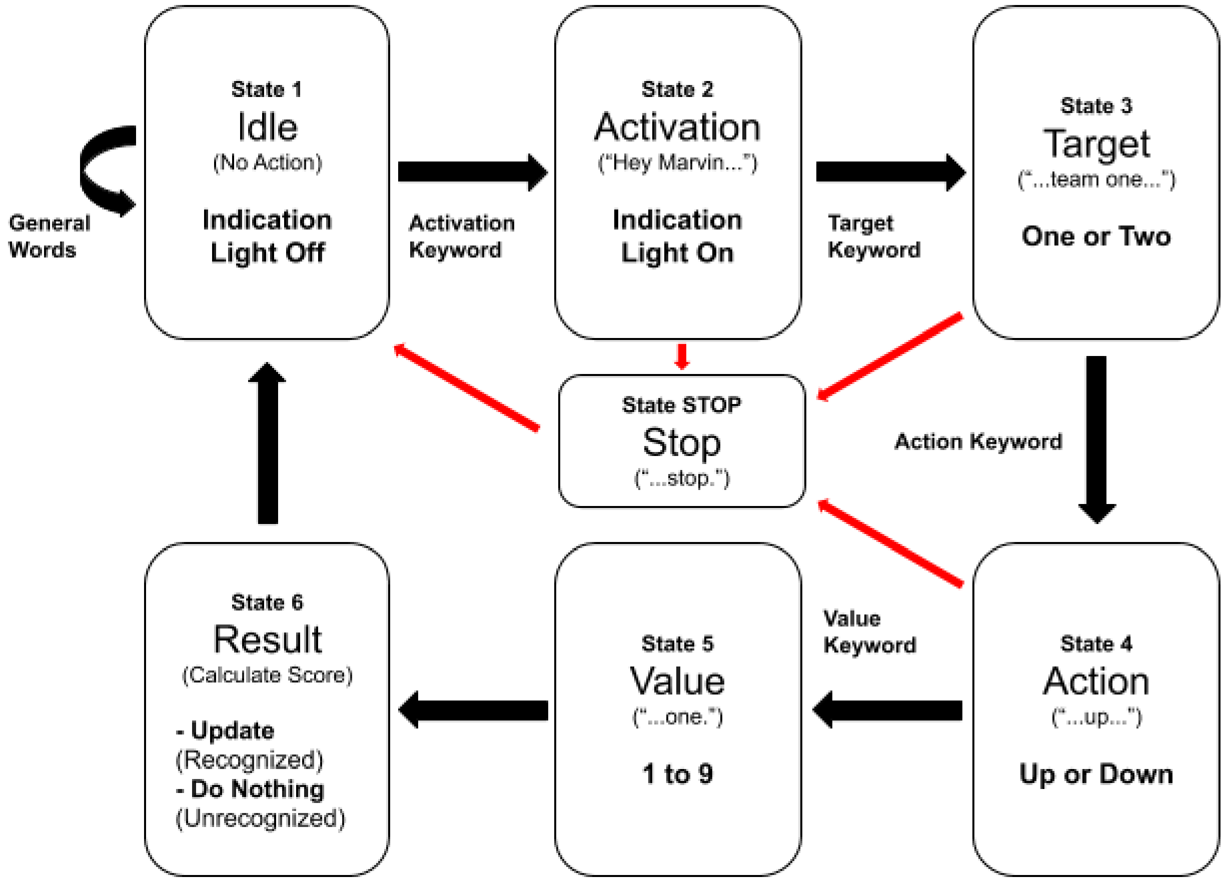

Figure 5. State Diagram

Our device will execute speech recognition based on Figure 5. This state control has 6 states: idle, activation, target, action, value, and result. As a summary, the device will typically be in its idle state. When the activation keyword is recognized (ie. "Hey Marvin!"), the device will actively listen to the next keywords consisting of the target (ie. team one or team two), the action (ie. 'up' to add, 'down' to remove), and the value (ie. amount of score to be changed, 1 to 9). If all keywords are recognized, it will update the score. If the keywords are not recognized correctly, it will do nothing. The "stop" keyword will be used whenever the device is unable to recognize any additional words and stuck in an unknown state, resetting the device back to the idle state.

2.4.2. Keyword Spotting Model

The keyword spotting model we will develop will be influenced by the research performed by the "Hello Edge: Keyword Spotting on Microcontrollers" [3] research paper. As a summary, the research team analyzed the different ways effective speech recognition could be implemented on microcontrollers and devices with lower power and lower computing resources. Their speech recognition model was implemented using tensorflow and online speech libraries. We plan to do the same using our selected keywords. Selected keywords were chosen based on if they were available in the audio dataset [6].

The final keywords selected are the following:
Activation Keyword: "Hey **Marvin**..."
Target Keyword: "Team **One**/**Two**…"
Action Keyword: "**Up**/**Down**…"
Value Keyword: "[**One**...**Nine**]."
Stop Keyword: "**Stop**"

The software will receive as input a one second speech signal containing one word and should correctly identify the word.

Requirements	Verification
1. Must identify inputs with at least 85% accuracy.	1. a. Input at least 100 audio files of each keyword into the trained model b. Report model accuracy and ensure it is at least 90%
2. Size of model in memory must be no greater than 200KB.	2. a. Convert tensorflow model to a C byte array b. Check the length of the array and verify it is less than 200,000

2.5. Tolerance Analysis

A key tolerance to ensuring our product is successful is ensuring that the battery is able to power the system for a minimum of four hours. In order to determine the battery life of our product, we must determine an approximate maximum current draw for our system. As seen in (1) this can simply be calculated by adding all of the individual components' maximum currents.

$$Max\ current\ draw\ of\ system\ =\ \sum Max\ current\ draw\ of\ all\ components \tag{1}$$

As seen in Figure 6, our estimated maximum current draw is 1585mA. Now that we have the max current draw of the system, we can determine how many mAh the battery needs. This can be solved by simply multiplying our max current draw by the number of hours the system must be powered. Seen in (2) and (3), the battery should be rated at a minimum C value of 6,340 or 6340mAh.

$$Minimum\ battery\ capacity\ (mAh)\ =\ Max\ current\ draw\ of\ system\ \times \#\ of\ hours \tag{2}$$

$$= 1585mA\ \times 4\ hours\ =\ 6340\ mAh \tag{3}$$

As a result, we chose to use a 6600mAh battery. This battery has a peak discharge current of 3000mA, however, for safety and battery health, a battery discharge rate of 0.2 - 0.3C (1320 - 1980mA) is recommended. With our estimated maximum current draw of 1585mA, our system will discharge the battery at 0.24C or less as seen in (5).

$$Estimated\ maximum\ discharge\ rate\ =\ \frac{maximum\ current\ draw}{battery\ capacity} \tag{4}$$

$$= \frac{1585mA}{6600mAh} = 0.24h^{-1} \rightarrow 0.24C \tag{5}$$

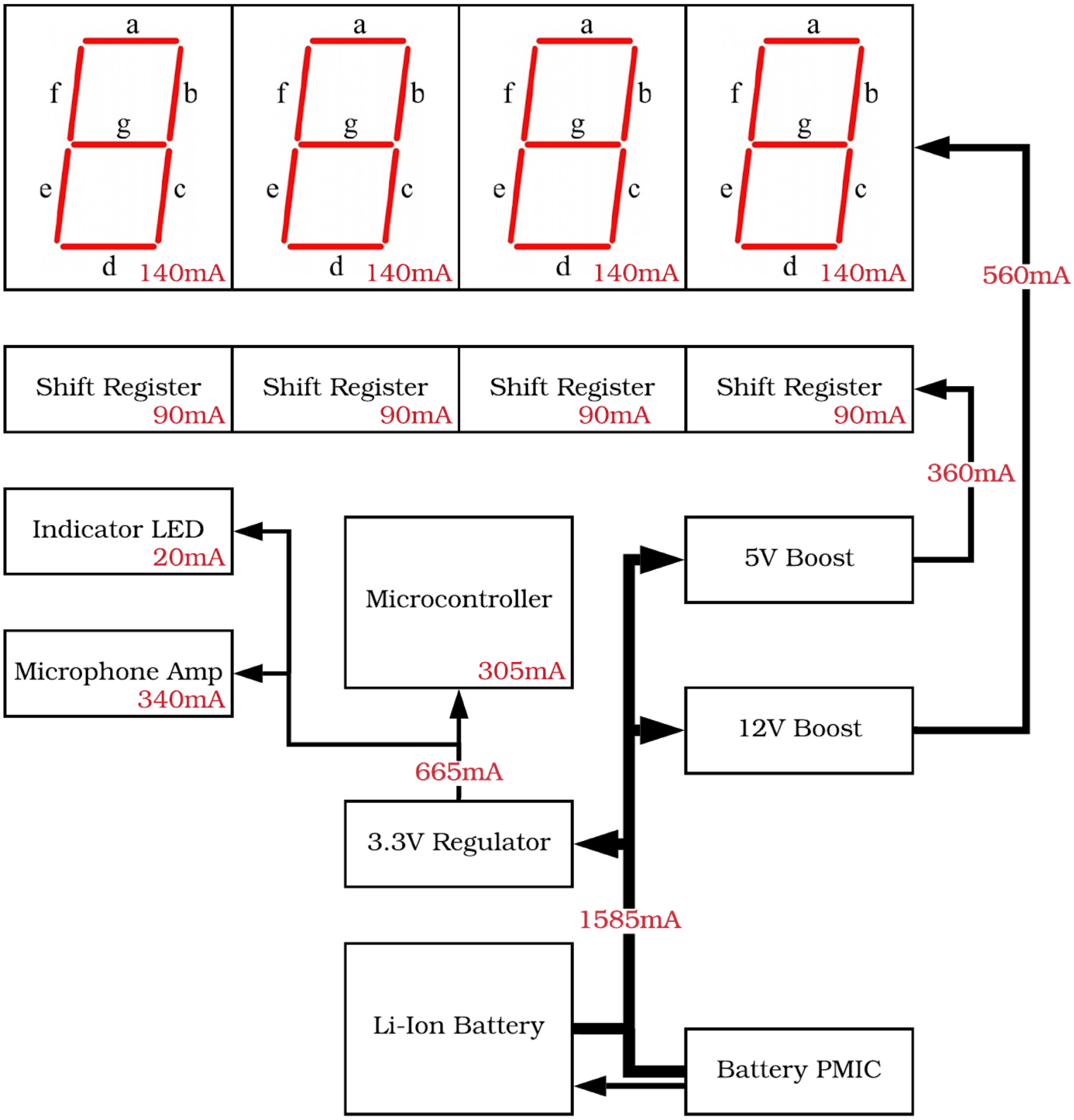

Figure 6. Maximum Current Draw Diagram*

* Approximate maximum current values were sourced from the component datasheets found in Appendix A

3. Cost & Schedule

3.1. Cost Analysis

Power Subsystem							
Part Name		Make	Part No.	Cost ($)		Quantity	Total ($)
				Single	Bulk*		
1	Battery PMIC	Texas Instruments	BQ25896	3.87	2.35	1	3.87
2	Battery	Pkcell	ICR18650	29.50	-	1	29.50
3	Voltage Regulator	Texas Instruments	TPS75233	10.81	6.13	1	10.81
4	5V Boost	Texas Instruments	TPS61089RNRR	2.94	2.94	1	2.94
5	12V Boost	Texas Instruments	TPS61089RNRR	2.94	2.94	1	2.94
Control Subsystem							
Part Name		**Make**	**Part No.**	**Cost ($)**		**Quantity**	**Total ($)**
				Single	**Bulk***		
1	Microcontroller	STMicroelectronics	STM32F401RCT6	6.27	3.32	1	6.27
2	Shift Registers	Texas Instruments	TPIC6C596DRG4	1.13	0.51	4	4.52
User Interface Subsystem							
Part Name		**Make**	**Part No.**	**Cost ($)**		**Quantity**	**Total ($)**
				Single	**Bulk***		
1	Microphone	CUI Inc	CMA-4544PF-W	0.83	0.38	2	1.66
2	Amplifier	Maxim Integrated	MAX9814	1.51	0.85	2	3.02
3	Seven-Segment Displays	SparkFun Electronics	COM-08530	18.95	-	4	75.8
4	Listening Light	Kingbright	WP154A	1.95	0.83	1	1.95
5	Power/Charge Status light	Kingbright	WP154A	1.95	0.83	1	1.95
6	Power Button	E-Switch	TL2201EEZA1CWHT	0.67	0.37	1	0.67
7	Increment Button	E-Switch	TL2201EEZA1CWHT	0.67	0.37	4	2.68
						TOTAL PARTS COST:	148.58
							*Price/piece (1000pcs)

Table 1. Cost breakdown of prototype

According to Table 1, our estimated prototype cost is $148.58.

According to the IlliniSuccess 2017-2018 annual report [9], the average salary for Computer Engineering graduates was $92,430 and $76,079 for Electrical Engineering graduates. This translates to an hourly rate of $46.22 and $38.04 respectively. As a result, our total development cost can be seen is calculated in (6) to be $48,930.

$$(2 \bullet \frac{\$46.22}{hr} + \frac{\$38.04}{hr}) \bullet 2.5 \bullet \frac{10hrs}{week} \bullet 15\ weeks = \$48,930 \tag{6}$$

3.2 Schedule

Week	Description	**Chris** Focus - Power	**Allan** Focus - Control	**Jason** Focus - Software
10/07/2019	Design Review	Continue work on PCB	Begin looking at mbed and familiarizing with the dev board, Finish ordering required parts	Research NN models and their input features
10/14/2019	Early Orders	Finalize initial PCB design and verify the proper working order of power supply components	Begin work on receiving mic input and processing the signal, start writing main program loop, verify all parts working as intended	Research ML model upload to MCU
10/21/2019	--	Hardware debugging and circuit modification	Continue work on processing audio and main program	Analyse audio dataset & begin creating ML model
10/28/2019	Progress Reports	Hardware debugging and circuit modification	Finalize audio processing code and main program	Test ML model with streaming audio
11/04/2019	Final PCB Orders	Finalize official PCB design	Bugfix audio processing code and main program	Upload ML model to PCB

11/11/2019	--	Construction of housing and assembly	Upload final code to MCU on PCB and bugfix any problems	Testing & Debug
11/18/2019	Mock Demo	Final adjustments	Integrate with power subsystem	Final Assembly
11/25/2019	Fall Break	Prep for final demo	Prep for final demo	Prep for final demo
12/02/2019	Final Demo	Prep for final presentation and start on final paper	Prep for final presentation and start on final paper	Prep for final presentation and start on final paper
12/09/2019	Final Presentation	Finish final papers	Finish final papers	Finish final papers

4. Ethics & Safety

Our projects main ethical concern is the voice recognition system. In recent years, users are becoming increasingly aware of the data that is being collected by large companies such as Google, Amazon, and Facebook; much of that data being collected without the knowledge and consent of the user. Amazon Echo devices have been an especially large privacy and safety concern ever since it was revealed that the device has recorded private conversations and even shared them [8]. This should not have been a surprise to consumers. Not because it is “obvious” that they are recording, but because these companies should have made sure that consumers were aware of the data being collected. According to the IEEE Code of Ethics #1 and #5 [5], engineers must ensure the safety of the public and improve their understanding of emerging technologies. Large tech companies are choosing to ignore these codes and leaving consumers unaware and oblivious to threats to their privacy. Because our project also utilizes a voice recognition system, we need to be cognizant of these facts.

Our project mitigates the dangers of voice recognition in the following manners. First, the length of time that the recording of the user will exist should be no more than a few seconds. By the time the score has been updated, the recording should no longer exist in the system. The system has a limited amount of memory (128kb), with the majority being used by the operating program and KWS model. The leftover amount of memory will likely range from a couple kb to around ten kb. If the system were to keep the voice recordings, it would quickly run out of memory. In addition, the system does not have internet connectivity, unlike Alexa. Alexa operates using a server to handle the voice processing and must upload the user’s voice command to the server where memory is not an issue. Because our system is not connected to the internet, it cannot upload the user’s voice recordings and must delete them to save memory.

In addition to the ethical and safety concerns of voice recognition, our project can also be physically dangerous to consumers due to the lithium-ion battery. Lithium battery failure can result from a range of causes such as puncture, overcharge, overheating, short circuits, and more. Because the battery generates and stores large amounts of energy, it is a fire and explosion risk due to the listed causes and can cause harm to the user [7]. In order to make sure that our product is as safe as possible for users, we need to make sure that our batteries are supplied by a reputable source to minimize manufacturing defects. In addition, we will ensure that there are protections in place against overheating and overcharging. It will also be impossible for users to remove the battery from the device unless they choose to break the device casing.

5. References

[1] D. Villegas, "Scoreboard - On The Go," Version 3.1, *Apple App Store*, 2019. [Online]. Available: https://apps.apple.com/us/app/scoreboard-on-the-go/id1388012712. [Accessed: 19 Sep. 2019].

[2] D. Zrobok, "Volleyball Scoreboard:", Version 2.8, Apple App Store, 2019. [Online]. Available: https://apps.apple.com/us/app/volleyball-scoreboard/id932240112. [Accessed: 19 Sep. 2019].

[3] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello Edge: Keyword Spotting on Microcontrollers," *arXiv*, February 2018. [Online]. Available: https://arxiv.org/pdf/1711.07128.pdf. [Accessed: 22 Sep. 2019].

[4] D. Gotterbarn, "ACM Code of Ethics and Professional Conduct," *Association for Computing Machinery*, 2018. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 19 Sep. 2019].

[5] "IEEE Code of Ethics." *IEEE,* 2019. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 19 Sep. 2019].

[6] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv*, April 2018. [Online]. Available: https://arxiv.org/pdf/1804.03209.pdf. [Accessed: 02 Oct. 2019].

[7] A. Semuels, "When Your Amazon Purchase Explodes," *The Atlantic*, April 2019. [Online]. Available: https://www.theatlantic.com/technology/archive/2019/04/lithium-ion-batteries-amazon-are-exploding/587005/. [Access: 03 Oct. 2019].

[8] C. Osborne, "Amazon confirms Alexa customer voice recordings are kept forever," *ZDNet,* July 2019. [Online]. Available: https://www.zdnet.com/article/amazon-confirms-alexa-customer-voice-recordings-are-kept-forever/. [Accessed: 03 Oct. 2019].

[9] "All Campus Undergraduate 2017-2018 Report," Illini Success. [Online]. Available: https://uofi.app.box.com/s/ml9oh48vawrhm7e019kw5ix4j91j605p. [Accessed: 03 Oct. 2019].

Appendix A - Datasheets

- Texas Instruments Battery PMIC - BQ25896:
  http://www.ti.com/lit/ds/symlink/bq25896.pdf
- PKCell Li-Ion Battery - ICR18650:
  https://cdn-shop.adafruit.com/product-files/353/C450_-_ICR18650_6600mAh_3.7V_20140729.pdf
- Texas Instruments Voltage Regulator - TPS75233QPWP:
  http://www.ti.com/lit/ds/symlink/tps75201-ep.pdf
- Texas Instruments Boost Converter - TPS61089RNRR:
  http://www.ti.com/lit/ds/symlink/tps61089.pdf
- STMicroelectronics Microcontroller - STM32F401RCT6:
  https://www.st.com/content/ccc/resource/technical/document/datasheet/9e/50/b1/5a/5f/ae/4d/c1/DM00086815.pdf/files/DM00086815.pdf/jcr:content/translations/en.DM00086815.pdf
- Texas Instruments Shift Registers - TPIC6C596DRQ1:
  http://www.ti.com/lit/ds/slis093d/slis093d.pdf
- Maxim Integrated Amplifier - MAX9814:
  https://cdn-shop.adafruit.com/datasheets/MAX9814.pdf
- SparkFun Seven-Segment Displays - COM-08530:
  https://www.sparkfun.com/datasheets/Components/YSD-1600AR6F-89.pdf