A Speech and Media Interaction Model for Individuals with Vision and Speech Impairments

by

Mihir Trivedi

S.B., Electrical Engineering and Computer Science,
Massachusetts Institute of Technology, 2021

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2022

Author .....
Department of Electrical Engineering and Computer Science
January 14, 2022

Certified by .....
Stefanie Mueller
Associate Professor
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

A Speech and Media Interaction Model for Individuals with Vision and Speech Impairments

by

Mihir Trivedi

Submitted to the Department of Electrical Engineering and Computer Science
on January 14, 2022, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Advancements in low-cost computing and electronics have created major opportunities in Accessible Technology for individuals with disabilities. Assistive Technology is especially important to introduce to children with disabilities at a young age, as it can have a significant impact on their learning ability.

This thesis presents a device and interaction model for children with vision and speech impairments. The device is a Speech Generating Device that allows children to distinguish between inputs that speak a configurable set of words, as well as control media on connected Bluetooth devices. The device is designed to facilitate easy interaction for use in early childhood education settings, including special needs classrooms and home environments. The device also expands on existing technology by facilitating easy configuration with a mobile and web application.

Thesis Supervisor: Stefanie Mueller
Title: Associate Professor

Acknowledgments

I want to sincerely thank my research supervisor, Professor Stefanie Mueller, for her advice and guidance, both in research and in teaching throughout my time at MIT. Thank you for your support and kindness in this journey. In pursuit of this research, I would like to thank my UROP collaborator, Ria Sonecha, for her work on our preliminary prototype and her valuable advice throughout my thesis work. Thank you to Christyn, Ryan, Peyton, and the Kiley family for their support and feedback throughout the entire design process.

My mentors in the field of Assistive Technology have been incredible, especially Kyle Keane. I appreciate Kyle and the entire PPAT teaching staff's work in teaching the importance of universal and collaborative design. Thank you to members of the HCIE lab, especially Faraz Faruqi, Junyi Zhu, and Yoonji Kim, who have been major supporters and advisors of my work. In building the hardware for the device, the advice and assistance of Bill McKenna and Chris Haynes at the IDC as well as Alec, Anthony, and Dave at the EDS has been indispensable. Thank you to Professor Fadel Adib for your guidance in my academic career as a UROP, Teaching Assistant, researcher, and mentor.

I want to thank Dave Dutton from Bellarmine who has been a continuous supporter of my engineering interests since high school. Finally, a special thank you to my parents, Harsha and Yatin, and my brother Naman whose support made attending MIT a reality for me. Thank you for your unconditional support and encouragement in my pursuits.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Accessible technology has been an area of interest for the research community for several years. The ACM SIGACESS was formed as a Special Interest Group to "empower individuals with disabilities and older adults" [1]. Research into the early childhood education of children with disabilities has found that consistency and repetition help students learn effectively, and that access to assistive technology can greatly improve their quality of education [2]. Existing assistive technology for Augmentative and Alternative Communication (AAC) is able to help users say specific words, but most do not extend to control of modern media devices, such as smartphones and tablets [3]. The advent of low-cost and low-power microcontrollers has made the creation of such devices more available to the community.

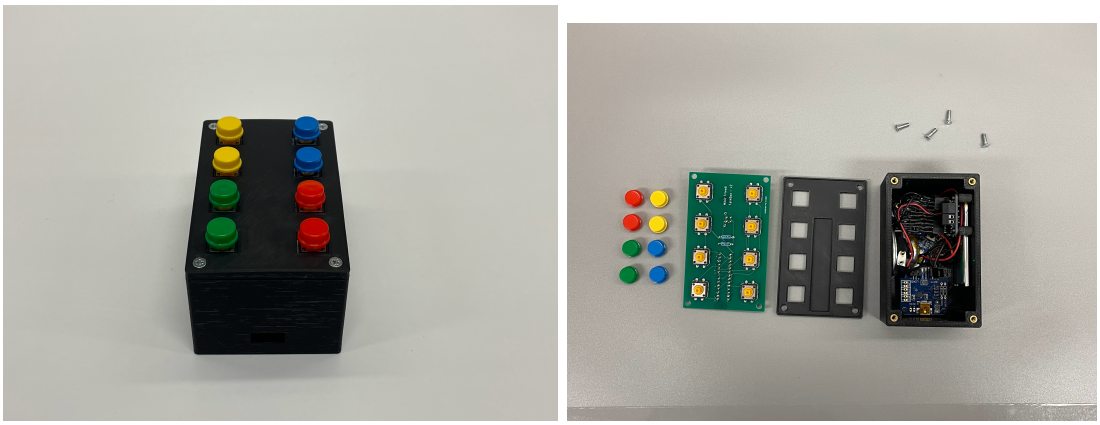

Figure 1-1: Complete device in enclosure (left) and individual device components (right).

Figure 1-2: Device shown with physically distinguishable inputs.

In this project, we present a new, configurable system that helps children communicate words and control media devices via Bluetooth in a compact and affordable device. This device is designed for children with vision and speech impairments that

make traditional communication methods difficult. However, its use case is not limited to children, and could be used by individuals of any age as a speech generating device. The device is designed to increase its functionality as the child learns, allowing them to learn its use over time and use it to communicate in their daily life. We provide users with a mobile and web application that facilitate configuration updates. We prototyped multiple iterations of electronics, microcontrollers, and mechanical designs to create the final design shown in Figure 1-1. Fabrication methodologies including manufactured PCBs, wire-to-board connectors, and threaded inserts were utilized to build a reliable device that is robust to daily use at home and in the classroom.

Chapter 2

Related Work

2.1 Assistive Technology

Significant research has been conducted into the design, use, and efficacy of Assistive Technology (AT) for children [4, 5, 6]. The discussion of Assistive Technology in this thesis follows the definition from the Encyclopedia of Clinical Neuropsychology, "assistive technology device means any item, piece of equipment, or product system, whether acquired commercially, modified, or customized, that is used to increase, maintain, or improve functional capabilities of individuals with disabilities". Existing work on early childhood education for students with disabilities has shown that consistency and repetition are a key component of productive education [7]. The advent of low-cost computing and other electronics have also widened the market and accessibility of AT for individuals with disabilities.

2.2 Vision Impairments

Assistive Technology for children with visual impairments can take many forms, from screen readers for smartphones and computers to refreshable Braille displays that make reading more accessible. Research technology such as Google Glass have also been used in studies for broader access and interpretation of the visual world. [8]. Other publicly available devices include HableOne [9] which allows visually impaired

users to make use of existing accessibility features in smartphones, and type messages using a Braille input mechanism. [10]

2.3 Speech and Communication Impairments

Augmentative and Alternative Communication (AAC) [11] is a category of Assistive Technology that facilitates spoken-language communication for individuals with speech and communication impairments. Recent advancements in machine learning have produced work in personalized AAC methodologies [12]. A large array of products for AAC exist in the public market already, generally controlled by large button-type actuators. Many of these devices are limited in features and prohibitively expensive [3], ranging from $100 to $700, depending on functionality. However, an easily reprogrammable, Bluetooth-enabled, and low-cost AAC device appears to be a difficult find in the existing market. The device presented in this thesis is designed with these issues in mind.

2.4 Assistive Technology in Early Childhood Education

The use of assistive technology has been proven to be effective for early childhood education both at school and at home [13, 14]. Specifically, research on AAC usage in preschools [2] found positive impacts because the devices provide an additional modality for communication with a range of others in their life. Support for universal communication is especially important for childhood development, as only a limited set of individuals close to the child would have learned special communication techniques such as sign language or braille. In the sub-category of Speech Generating Devices (SDGs), which pertains to this project, work [15, 2] has found improved communication outcomes for preschool-aged children using SGDs.

Chapter 3

Implementation

3.1 Preliminary Designs

A prototype of this device was built previously as part of a UROP supervised by Professor Stefanie Mueller and Dr. Kyle Keane at MIT. The goals of the previous device were similar, providing a platform for tactile activation of words spoken by an assistive device. The preliminary device and its designs can be found in Figures 3-1 and 3-2. The author along with UROP student Ria Sonecha built this preliminary device as part of the UROP. The processor for the first prototype was a Raspberry Pi Zero W, a small Linux computer which has WiFi and Bluetooth/BLE capability.

Through the process of building the preliminary device, we found multiple areas for improvement which have guided the design choices for the device described in this thesis. With regards to software, we found that the full Linux-based OS running on the Raspberry Pi created a lot of unnecessary overhead that drained extra battery and caused communication issues over Bluetooth. Due to preexisting driver issues, the Bluetooth advertising and pairing process was inconsistent during testing. In our hardware implementation, we created slots in the lid portion of the 3D printed enclosure, and soldered directly to the exposed pins of the pushbuttons shown. This method of soldering directly to the button pins made for a very fragile connection, which broke regularly. This is why we chose to manufacture PCBs for the current iteration of the device.

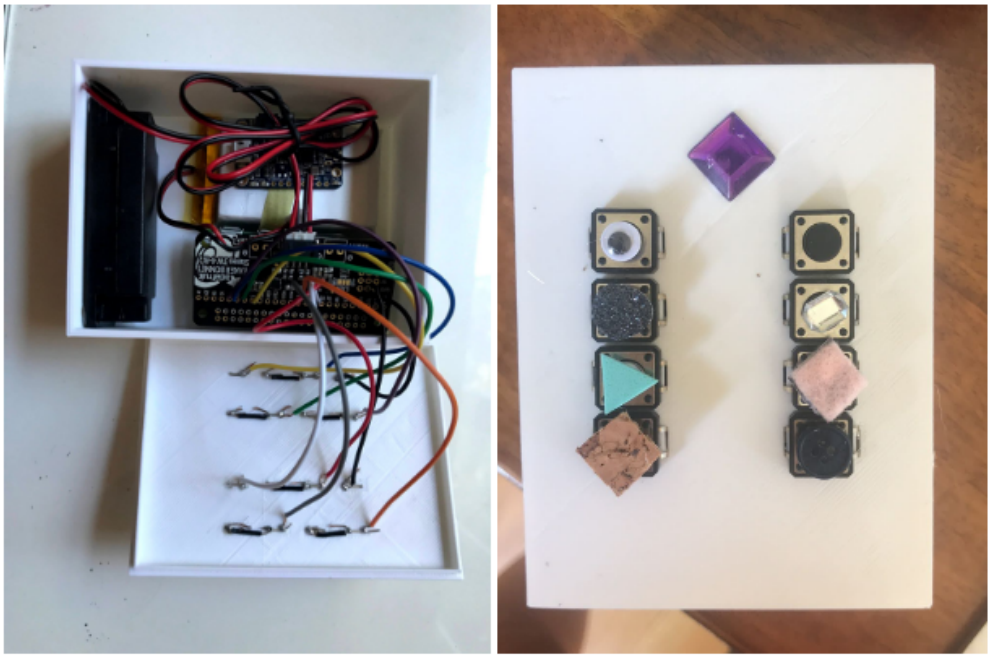

Figure 3-1: Electronics inside the preliminary device (left) and completed device with physically distinguishable inputs (right).

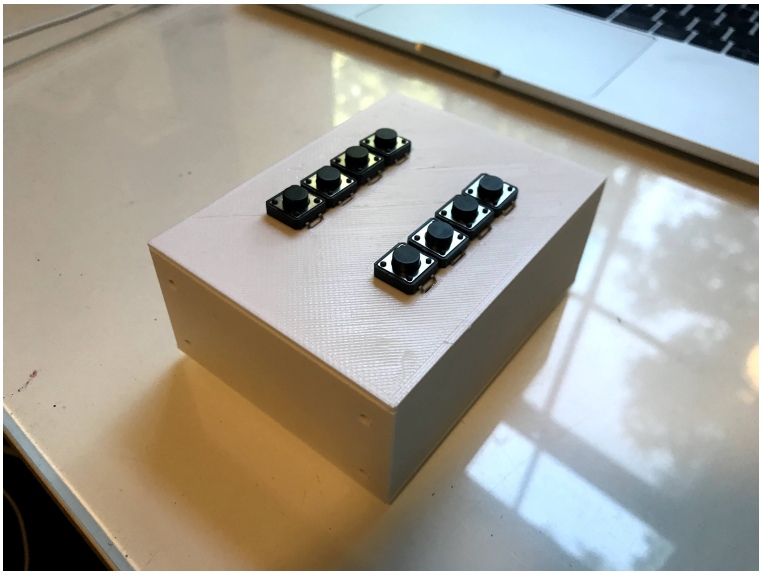

Figure 3-2: Side view of the preliminary device, showcasing button input locations.

3.2 Hardware Design

The hardware design for this project falls into three main categories: processing, electronics, and enclosure.

3.2.1 Processing

The central processor used for Bluetooth communication, processing, and audio driving is the Adafruit Circuit Playground Bluefruit, which utilizes the Nordic Semiconductor nRF52840 microprocessor. It was chosen primarily for its functionality, including: prebuilt libraries for Bluetooth Low Energy (BLE) communication, a power port for LiPo batteries, audio driver functionality, and I^2C communication pins. An image of the board can be found in Figure 3-3.

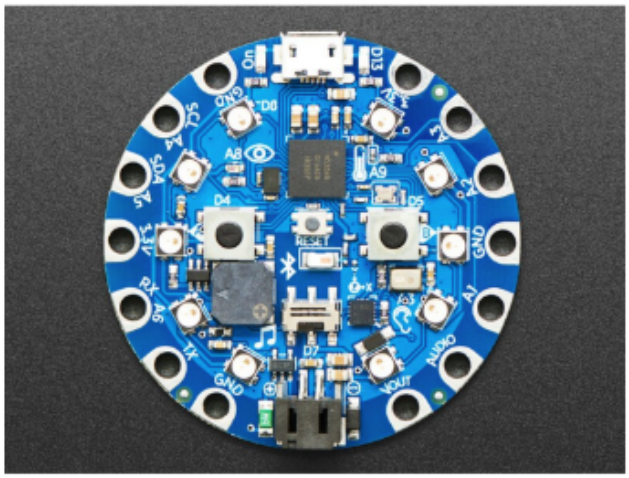

Figure 3-3: The Adafruit Bluefruit board.

Bluetooth Communication

The Adafruit Bluefruit board utilizes the nRF52840 chip's BLE capability to advertise its services to a host device (e.g. a smartphone or tablet). In the context of BLE, the Bluefruit board acts as a peripheral, and the connected smartphone or tablet is the

central. In this device, we utilize two different Bluetooth services which identify the types of information the peripheral can exchange with the central. For Bluetooth-based control of media devices, we utilized the Apple Media Service, which allows the peripheral device to send instructions to the central for play/pause, next/previous track, and volume control of media devices. We chose to focus first on Apple iOS and iPadOS devices, though the service framework could reliably be expanded to control other media sources such as Android smartphones.

For the configuration mode described in 3.3.1, the peripheral advertises the UART (Universal Asynchronous Receiver-Transmitter) service to exchange configuration information using our iOS application.

Audio Control

A core feature of the device is to vocalize requested actions and words. When pressing one of the input buttons depicted in Figure 1-2 , the device speaks the corresponding word and, if applicable, activates the relevant media function. We utilize the audio output feature of the Adafruit Bluefruit to drive a TPA2016 Audio Amplifier, which plays audio via a connected 3 Watt speaker.

I^2C Communication

In order to communicate with the PCB described in 3.2.2, we utilized the I^2C protocol with the MCP23017 chip. The Adafruit Bluefruit support I^2C communication via the SDA and SCL (data and clock, respectively) pins, each with a 4.7Ω pull-up resistor. Details of the circuit can be found in Figure 3-4.

3.2.2 Electronics

The electronics used to create this device consist of four main components: a custom-designed PCB for user interaction, a Lithium-Polymer battery and corresponding charging circuit, an audio amplifier and corresponding speaker, and a microcontroller.

PCB for User Interaction

In order to facilitate robust user interaction, we designed a custom PCB to solder buttons and other electronic components to. The PCB was designed in EagleCAD and was laid out to provide a rigid placement for the interface buttons that protrude from the top of the enclosure. A schematic of the PCB is shown in Figure 3-4.

Due to the limited number of GPIO (General Purpose Input/Output) pins on the selected microcontroller, the PCB was designed to include an extra chip that increases the number of individually readable input pins. The component labeled MCP23017 in the schematic is a GPIO expander from Microchip Technology. This chip was selected because it provides internal pull-up resistors, interrupt capability, and I^2C communication with the microcontroller. The ability to communicate over I^2C greatly reduces the wiring required between the microcontroller and PCB.

The schematic shows one side of each switch connected to ground, labeled GND. The opposite (normally disconnected) side of the switch is connected to a corresponding pin on the MCP23017, in a pull-up configuration. Enabling the built-in pull-up resistor allows the microcontroller to reliably determine whether the switch is pressed or unpressed. The microcontroller is then able to individually read the state of pins via the I^2C protocol [16].

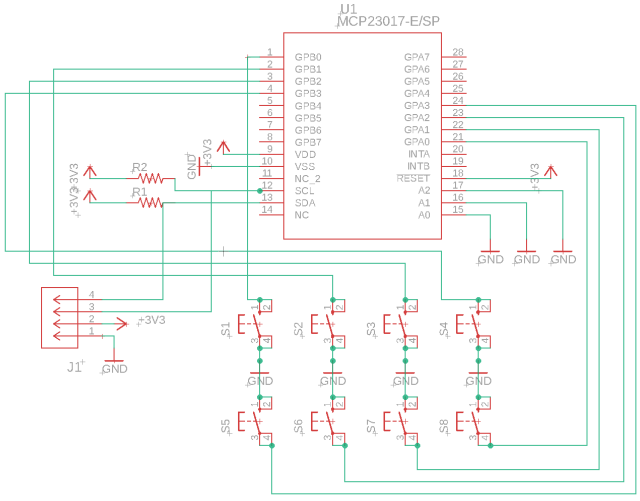

Figure 3-4: PCB Schematic with MCP23017, connector, resistors, and pushbuttons.

The physical layout of the PCB was also designed in EagleCAD with user interaction in mind. The size of and distance between buttons was designed to allow users to effectively differentiate between button caps, while keeping the bounding dimensions of the device wearable. The internal components (MCP23017, SDA/SCL pull-up resistors, and wire-to-board connector) are placed in the middle of the board to conserve space and hide neatly underneath the enclosure.

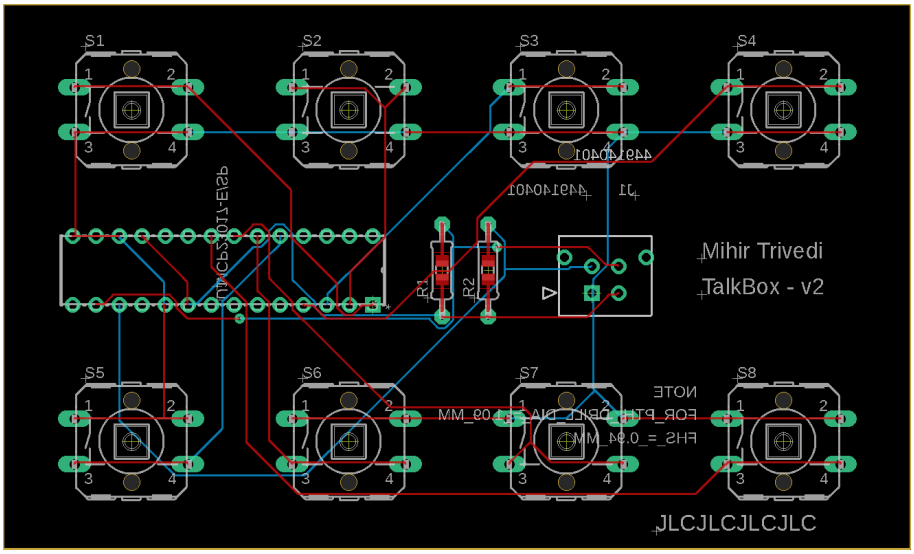

Figure 3-5: PCB layout.

In order to connect to the microcontroller, the PCB utilizes a Molex Micro-fit Wire-to-Board connector, labeled J1 in Figure 3-4 and is seen as the rectangle adjacent "Talkbox - v2" in Figure 3-5. The connector and corresponding clip-on wire provide a robust connection to the PCB that does not require soldering, but can be removed to access the internal components of the device.

The manufactured PCB can be seen with the top input buttons soldered and rear with MicroFit connector in Figure 3-6.

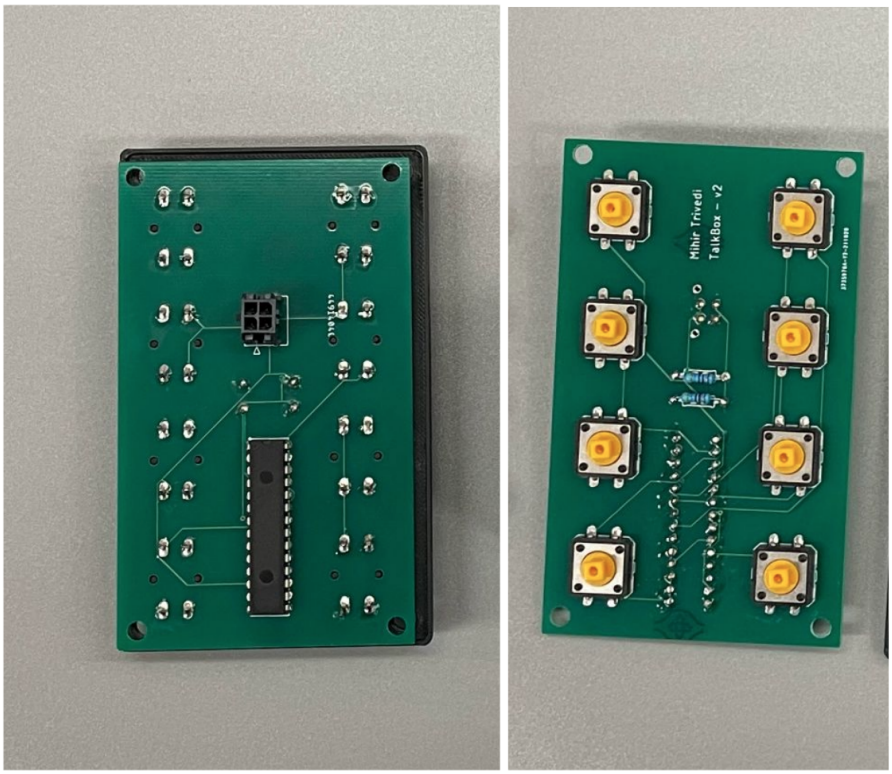

Figure 3-6: Manufactured PCBs after component soldering, bottom (left) and top (right).

Battery and Charging Circuit

Powering the entire device is a 3.7v, 1200mAh Lithium-ion polymer battery. This battery was chosen based on calculations made with power draw information for the nRF2840 SoC which powers the Adafruit Bluefruit microcontroller. The current draw in multiple modes is detailed in Table 3.1.

The current draw of the TPA2016 amplifier with 4Ω speaker at maximum volume draws about 400mA. The duration of each word spoken via the amplifier is about 1 second. Based on the information provided in the datasheets, with a rate of 250 uses (inputs on the device) a day, the worst-case lifetime of our battery would be 5 days and 3 hours. In trials, the battery has lasted an average of 6 days with similar usage.

Table 3.1: Power Draw for operating modes of nRF2740.

Mode	Current Draw
BLE Transmitting	6.40 mA
BLE Receiving	6.26 mA
CPU Running (no BLE)	3.3 mA
CPU Running + BLE transceiving	8.6 mA

The charging circuit is configured for 500mA charging, a range which is safe for the specific LiPo battery chosen and prevents excessive heating during the process. By using the MCP73833 chip to manage charging, we are able to provide power to the microcontroller both while connected to just the battery as well as during charging. A sample of the charging board is seen in Figure 3-7. The charging circuit also utilizes a USB mini-B port for easy charging of the battery. A hole in the enclosure is provided so a user may plug a charging cable directly into the device.

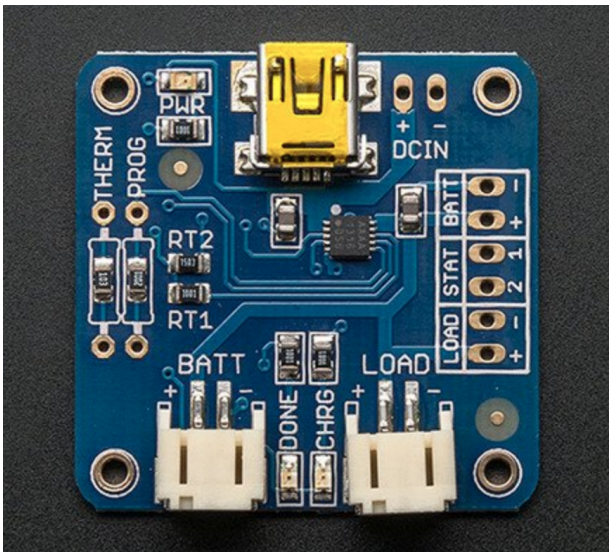

Figure 3-7: LiPo Charging Board with MCP73833.

Audio Circuit and Components

In order to effectively play audio clips from the Adafruit Bluefruit microprocessor, we utilized a 4 Ohm, 3 Watt speaker, driven by a Class D Amplifier, the TPA2016

(Figure 3-8). The TPA2016 amplifier supports stereo amplification, which provides an opportunity to add an additional speaker if required to increase the volume of the device. We utilize the shutdown pin of the amplifier in order to conserve power when the device is not in use. The audio input to the amplifier is fed from the Audio pin (AO) of the Bluefruit board, and powered via the 3.3V pin from the same board. While the Bluefruit board can produce an audio signal, it is not designed to power a speaker itself. The amplifier allows us to safely activate the chosen speaker with a source signal from the Bluefruit.

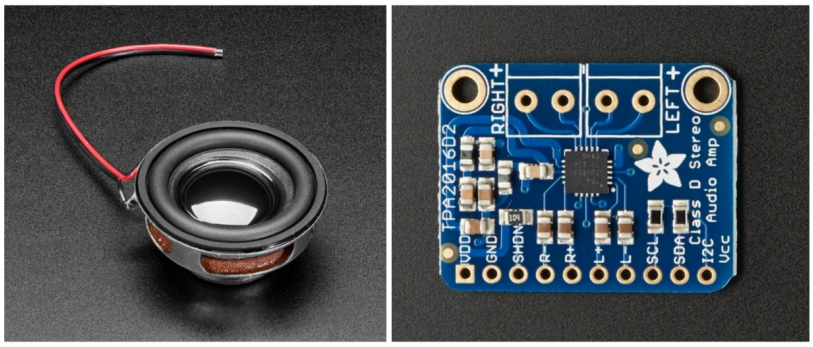

Figure 3-8: 3 Watt speaker (left) and Class D amplifier board (right).

3.2.3 Enclosure

The enclosure for the device was designed in a CAD tool, OnShape, to produce the minimal size that would enclose the electronics and interaction points. The enclosure is built in two parts, a container and removable cover, for easy assembly and packaging of the device. CAD images of the container, cover, and total assembly can be seen in Figure 3-9.

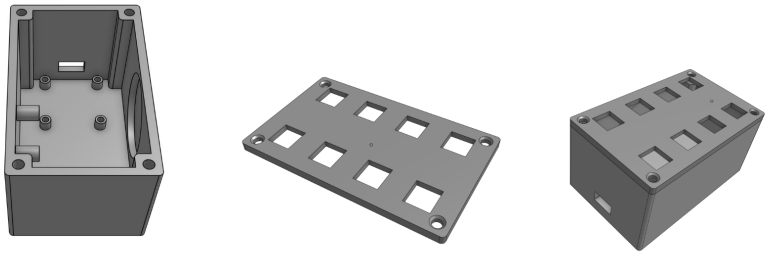

Figure 3-9: Enclosure container (left), cover (middle), and complete assembly (right).

The enclosures were printed via a Prusa i3+. To ensure the rigidity of the container, it was printed with a high infill (75%) – this produced dense walls around the object so it could not be easily bent by hand or by impact from a short fall (tested as a drop from 5 feet above ground). The 3D printer used for manufacturing can be seen printing the lid in Figure 3-10.

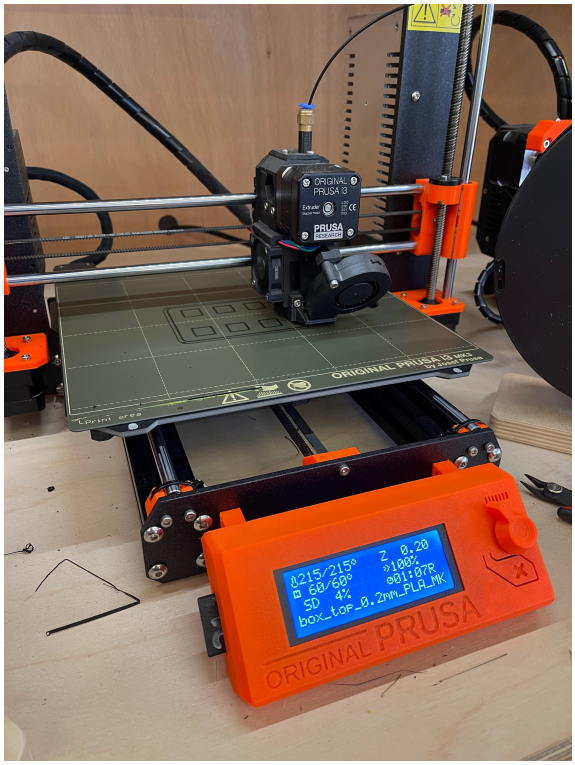

Figure 3-10: The Prusa i3+ printing the lid of the enclosure, using PLA filament.

Enclosure Design Details

The container base contains multiple features for sturdy mounting and robust encapsulation of the electronics. Two mounts are provided for securely mounting components: four holes on the base are used to mount the charging board, providing a stable platform for multiple plugs and removals of the charging cable. The board faces an exterior wall with a cutout, seen at the top of Figure 3-9 (left); this allows access to the USB mini-B charging port. The other mounting holes can be seen along the top of the left side wall. These are used to hold the amplifier chip described in Section 3.2.1. The holes are extruded far enough to allow the battery to be mounted behind the amplifier board.

In order to reliably screw boards and other components into the 3D printed enclosure, we utilized heat-set threaded inserts (Figure 3-11, left). These inserts are placed in holes created by the 3D print (Figure 3-11, right), and melted into the print with a soldering iron. Once in place, the inserts prevent wear on the plastic build material of the enclosure, allowing for multiple insertions and removals of screws.

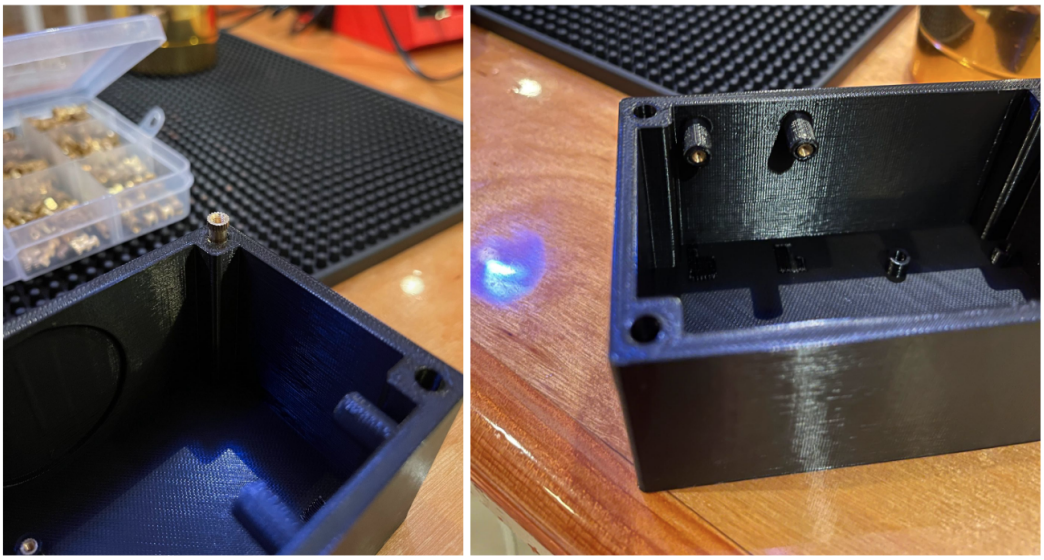

Figure 3-11: Heat-set threaded insert before heating (left) and after insertion (right).

An inset for mounting the speaker is seen opposite the amplifier mount points. The inset allows the speaker to be mounted along its edges and reduces the amount of

plastic between speaker and the outside, decreasing the muffling effect of the plastic.

The lid of the enclosure is designed to provide access only to the eight input buttons atop the PCB. After attaching the lid with four corner screws, the user can attach replaceable button caps. The button caps can be customized by texture, braille, or other physically distinguishable features.

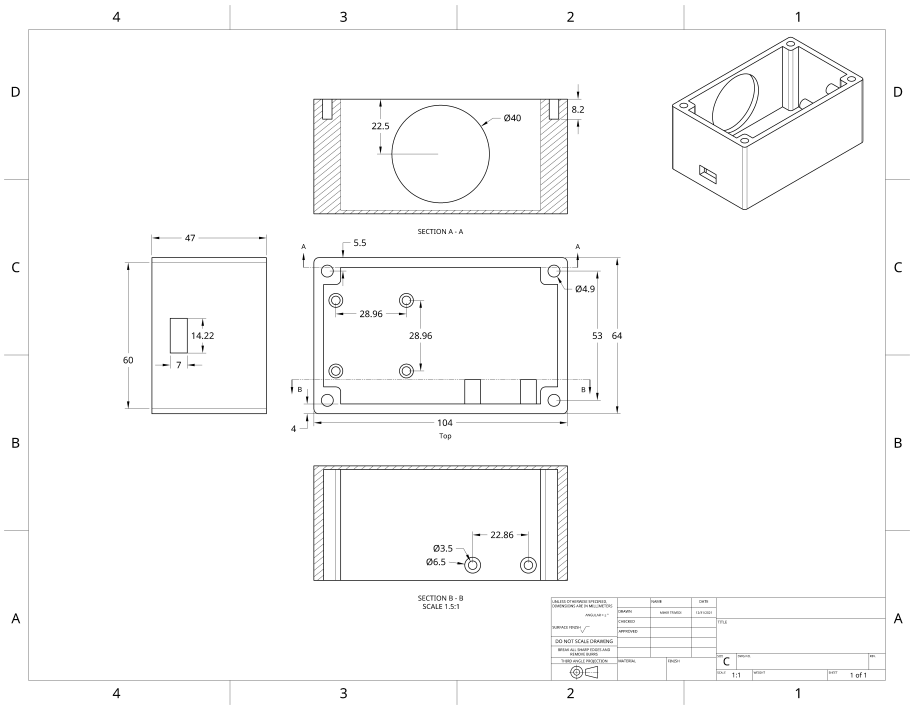

Figure 3-12: Technical drawing of the container base.

Input Distinguishability

In order to improve the customizability of our device, we utilized buttons with replaceable and modifiable caps, shown in Figure 3-14 (left). Because the use case of our device is to provide a Speech Generating Device for use by the visually impaired, especially children, we attached distinguishable, common materials to the button caps instead of braille. The device with its button attachments can be found in Figure 3-14 (right). A selection of the materials used include: felt, construction paper, buttons, and sanded plastic.

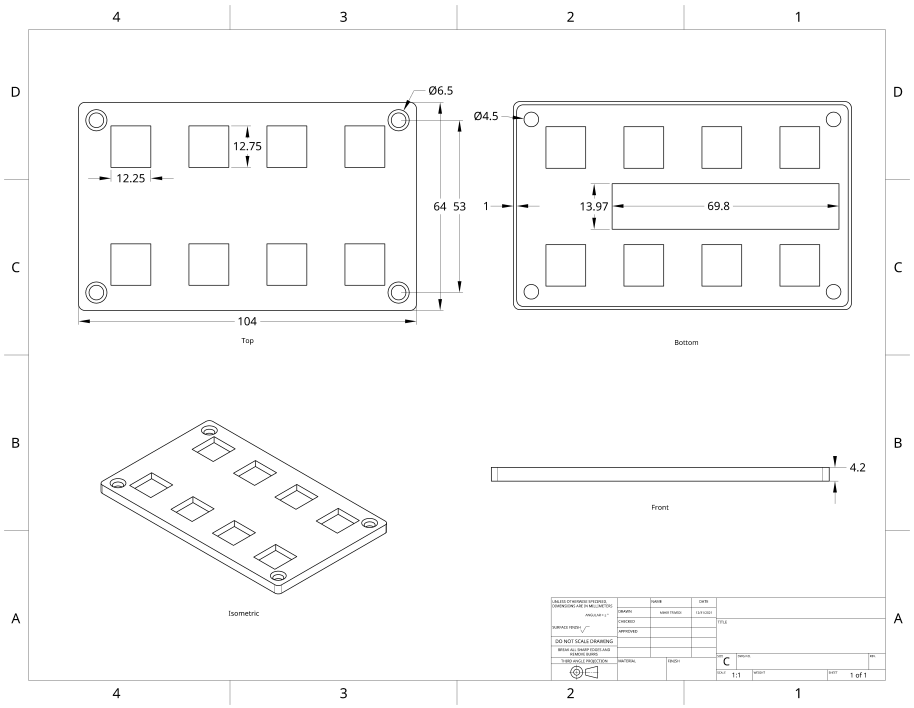

Figure 3-13: Technical drawing of the enclosure lid.

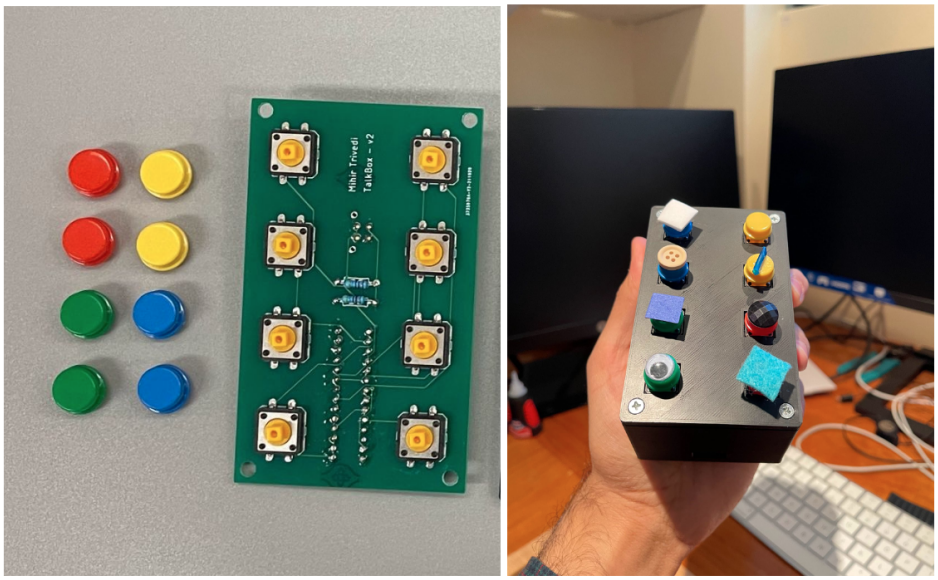

Figure 3-14: Individual button caps (left) and complete device with modified caps attached (right).

Table 3.2: Itemized cost for a prototype device. Numbers reflect average prices for items bought at the time of design.

Item	Cost
Bluefruit Microcontroller	$24.95
Battery	$7.99
Speaker	$4.95
Amplifier	$9.95
Charging Board	$14.99
PCB	$5.00
Total	**$67.83**

3.2.4 Cost Analysis

A guiding principle of the design for this SGD was a low-cost product that could be easily built from commercially available hardware and electronics. Table 3.2 describes the individual COTS (commercial off-the-shelf) cost for the device, totalling under $70. In comparison, many existing SGDs on the market cost between $250 and $500. Most are limited in their configuration ability and have limited external interface capabilities.

Chapter 4

Software Design

4.1 Usage Note

This chapter refers to two types of "users" – configuration users and end users. In this context, the *configuration user* refers to a parent, caretaker, or teacher who helps a student learn to use the accessible device. The individual who is learning to use the device is the *end user*.

4.2 System State Machine

The actions of the device can be modeled as a Finite State Machine, with two primary modes and actions available in each mode. On startup, the device begins in playback mode, which allows the end user to immediately begin using the functionality of the device. Based on the configuration of the device at the time, pressing an input button can: have no response (inactive), speak a word, or speak a word and perform a Bluetooth action. Upon completion of the configured action, the device returns to the base playback state and waits for another input. By holding buttons 1 and 2 at the same time, the configuration user can switch the device to configuration mode, to be used with the mobile application. The message received over BLE from the mobile application then dictates which configuration should be updated and saved: either toggling the audio functionality of an input button or saving a word assignment to a

specific button. A diagram showing this workflow can be found in Figure 4-1.

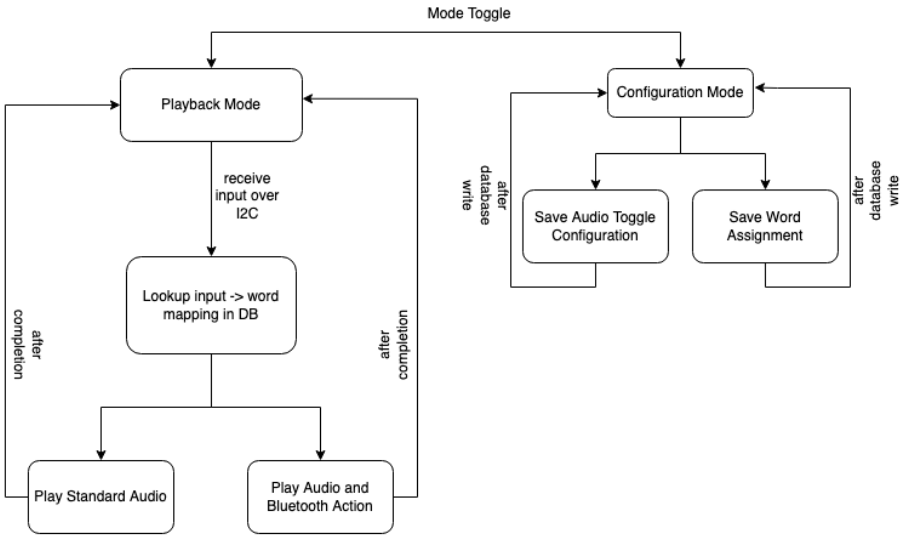

Figure 4-1: State machine of the device's software.

4.3 I²C Communication

The MCP23017 GPIO Expander was used in order to individually access the state of all 8 input buttons. The inputs could not be directly connected to our microprocessor due to the limited number of GPIO pins. The chip communicates over I^2C , which the Bluefruit board provides hardware support for. Using a compatible I^2C communication library, the software running on the Bluefruit instantiates a connection to the GPIO Expander chip, and registers the relevant pins as input pins with internal pull-ups. After initial setup is complete, the playback mode loop repeatedly polls each input for its value over the I^2C bus.

4.4 Bluetooth for Media Control

In playback mode, the device is able to control media playback devices (specifically iOS devices) via the Apple Media Service. This is a BLE profile developed by Apple that allows for media control with the following functions: play, pause, next track,

previous track, volume control, and loop mode (repeat and shuffle). The service also exposes commands for liking, disliking, and bookmarking tracks which we did not make use of in our implementation. The Apple Media Service exposed by iOS devices is composed of three *characteristics*: Remote Command, Entity Update, and Entity Attribute. [17]. In order to make a command request to the connected iOS device, we utilize the Remote Command characteristic to send an integer-mapped command as described in the reference.

4.5 Bluetooth for Device Configuration

In configuration mode, the device is able to communicate with our mobile application over UART. In order to reliably communicate standardized information over the Bluetooth radio, two concepts are used: Python *struct* packing and 2-bit *checksums*. The use of the Python struct library allows us to type-cast incoming Bluetooth packets and action based on the type of information being delivered. Checksums verify that a received packet contains all the information that the sender intended to transmit.

Struct Packing

In order to standardize the types of messages sent between the mobile application and the device, we utilize the Python struct library to check which type of information is being received over UART. The basic structure follows the format of the `Packet` abstract class built by Adafruit. We implement three types of `Packets`: enable/disable, volume, and word selection.

Header and Checksum

The packet header is always 2 bytes, "!" followed by a letter that identifies the packet type, e.g. `!S`. The checksum is always 1 byte, as described in further detail below.

Enable/Disable

Header: S

The payload of the enable/disable packet is `s,s`, which identifies a 1-byte button number followed by True or False: `<BUTTON>,<T/F>`.

Volume

Header: V

The payload is `f`, a float representing a volume on the range 0,1.

Word Selection

Header: W

The payload is `s,10s` representing a 1-byte button number and a word up to 10 characters long.

Checksums

Because the packets we transmit over BLE are short, a simple checksum algorithm sufficed for our case. Our checksum is based on Adafruit's original checksum provided in the Bluefruit Connect application. To compute the checksum, we cast the string to a byte format (`UInt8`), sum the bytes using a wrap addition [18] and invert the result. This resulting checksum byte is added to the end of the packet before it is sent over BLE. Once the packet is received by the device, a similar process is followed to verify the checksum. Since the checksum is known to be one byte long and always at the end, the first `n-1` bytes are run through the same checksum algorithm, and the result is compared to the sent checksum. If the packet's checksum is missing or does not match the generated one, the packet is discarded.

4.6 Configuration Persistence and Storage

In order to maintain saved configurations, we utilized the onboard storage capability of the Adafruit Bluefruit board. The board contains a 2MB on-board SPI Flash storage. SPI Flash storage devices are commonplace on small form-factor microcontrollers

because they are small, low-cost, and able to hold non-volatile memory that persists between reboots and power loss on the board. The 2MB of storage is shared between the code, libraries, audio files, and configuration files.

The write state of the Bluefruit board can be in one of two mutually exclusive states: external writes from another computer to the local filesystem (e.g. PC saving code to Bluefruit), or an internal write from the processor to its own filesystem. When in the external write mode, the Bluefruit cannot save any text files to its local filesystem, but is able to access and read the file's contents. During testing and development, the mode was switched using a switch lever on the board itself. In production usage, the device remains on internal filesystem write mode, as the external write is only useful for uploading code to the board.

4.6.1 Audio Toggle

The configuration for use of specific input buttons is saved as a dictionary, mapping button numbers (1-8) to boolean values. A "false" mapping means that the button is disabled – no end user input can activate the corresponding word or Bluetooth action, if relevant.

4.6.2 Word Choice

Similar to the audio toggle menu, each input button's corresponding spoken word is saved to the persistent filesystem. The dictionary maps selectable words to input button numbers after a save command is received from the mobile application.

4.7 Mobile Application

In order to easily update the configuration of the device, a mobile application was created to update and save device saved configurations. The mobile application is built for the iOS ecosystem using Swift, although it could be extended to other platforms such as the Android app store. The underlying technology for UART

communication over BLE can be utilized with similar libraries for Android.

Our mobile application is a fork of the existing Adafruit Bluefruit LE Connect mobile application. Using this existing application greatly sped up the development process, as Adafruit has already built out and included libraries for the core capability of Bluetooth Low Energy communication and UART messaging.

4.7.1 User Interface

The User Interface ("UI") is the interaction point for the configuration user to help set the device up for the end user's learning and usage.

Audio Toggle Page

The Audio Toggle page allows the configuration user to configure the activation of specific buttons as well as the volume of the device's internal speaker. The design is intended to contain as few controls as possible for ease of use and potential compatibility with iOS accessibility features, such as VoiceOver. To toggle the audio functionality of any button on the device, the configuration user clicks on the corresponding number, changing its color from blue to gray or vice versa. A gray number indicates that a button is inactive, while a blue button indicates that it is activated. Clicking the Save button saves the configuration to the device.

The volume slider below the toggle buttons provides a method of controlling the volume of the internal device speaker.

Word Assignment Page

The Word Assignment page allows the configuration user to reassign which buttons correspond to which spoken words. Using two scrollable selectors, this user is able to choose a button number and word, then click Save to update the device's configuration.

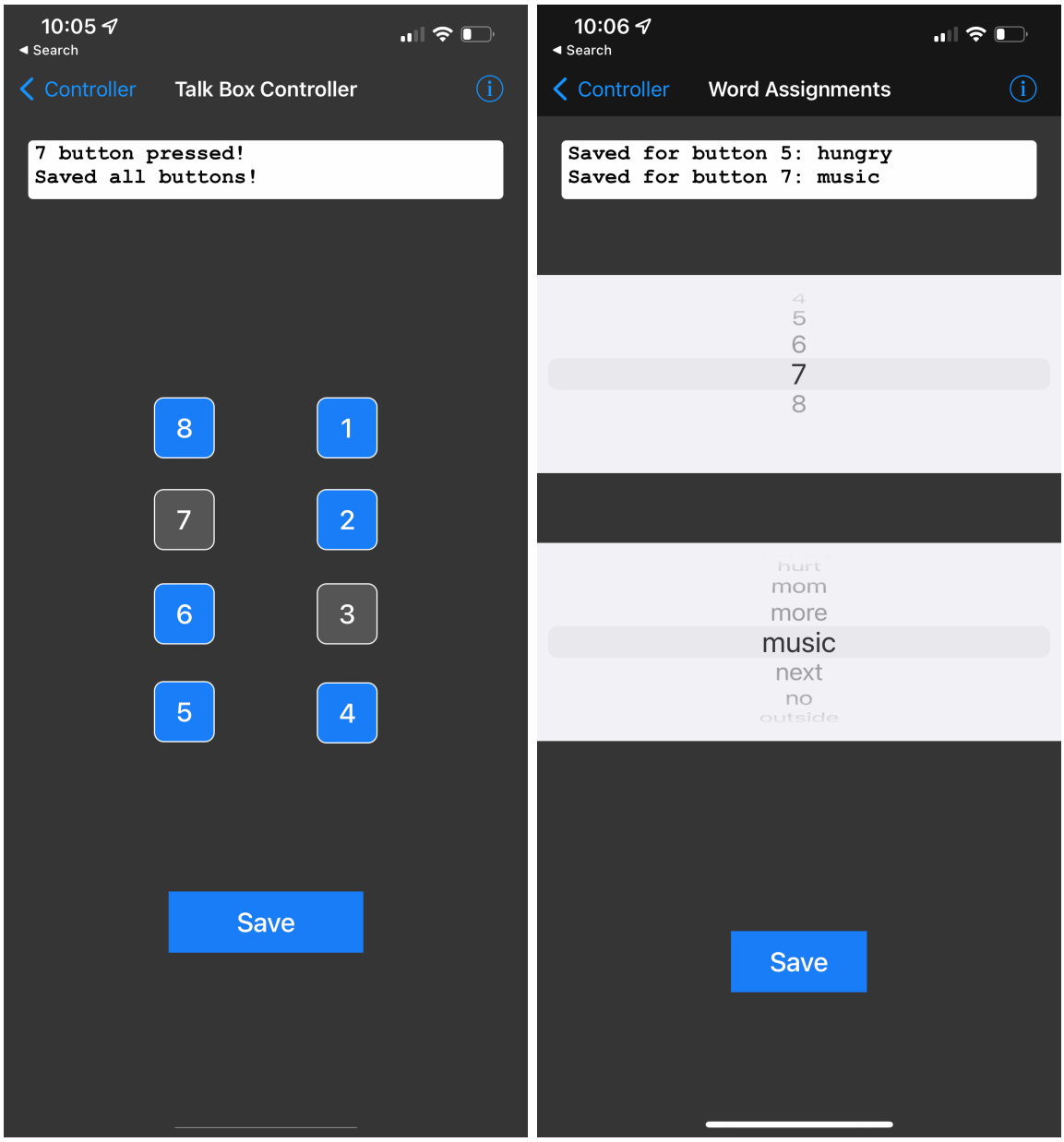

Figure 4-2: User Interface for the Audio Toggle (left) and Word Selection (right) pages.

Chapter 5

Discussion

The primary use case of this device is to supplement existing methodologies for AAC, incorporating modern media interactions into a speech generating device (SGD). Daily use cases include utilization of the device for speech and communication therapy, for example with sign language instruction. For children with vision and speech impairments, teaching alternative forms of communication includes repetition to allow the student to practice a given word or phrase. One example of incorporating an SGD into an education routine would be a teacher or parent saying a word, helping the student sign that word, then helping them press the respective button on the device. This repetitive cycle helps the student learn how to communicate the word by reinforcing the device's usage with an existing language (sign language). Once the association between a specific input and a given word has been formed, the student can utilize the SGD to communicate with a wide variety of other individuals, instead of just those who understand sign language.

5.1 Limitations

5.1.1 Bluetooth Capability

The current implementation requires the user to manually switch the device from Playback mode to Configuration mode with a specific button press. This is due to

the Bluetooth limitations of the library used, which only allows a single service to be advertised at a time. As a result, the pairing process for switching from Playback to Configuration mode Bluetooth connections requires the target smartphone or tablet to be manually unpaired and reconnected. In order to have one connection support both configuration and playback, modifications to the used BLE library may be necessary.

5.1.2 Click Suppression

Although the Class D amplifier we utilized in the device has an Automatic Gain Control (AGC), we found that the beginning and end of certain audio playbacks caused clicking noises that were not present in the source audio files. Clicking or popping sounds are a common issue when changing playback state to and from an idle state. We attempted multiple solutions, including utilizing the TPA2016's configurable gain to reduce the unwanted noise. While we were able to reduce the intensity of the clicking, we were unable to remove it entirely.

5.2 Future Work

A few additional features would increase the functionality of this device greatly, expanding its utility for those with a range of visual and communication impairments. Notably, the current implementation pre-selects 50 words that a user may choose to use in the mobile and web applications. However, allowing the user to record or upload their own word or audio file would greatly expand the usability of the device. Using the existing UART BLE implementation, a user could record a word directly on the mobile application and transfer the audio file directly to the device.

Chapter 6

Conclusion

In this thesis, we presented a speech generating device that allows children with vision and speech impairments to more effectively communicate with others and control their own media devices. The hardware design provides a simple interface for the user to learn, configurable via a mobile application that can be used by an instructor, therapist, or parent. The device is designed to last about 5 days on a single charge, and perform simple media control interactions for connected Bluetooth devices. We tested this device in simulated usage conditions, and measured its robustness through that usage. The work covered in this thesis represents a step forward in accessible technology for children and provides a framework for future low-cost developments of similar Accessible Technology.

Bibliography

[1] ACM SIGACCESS Homepage. https://www.sigaccess.org/.

[2] R. Michael Barker, Aanae Akaba, Nancy C. Brady, and Kathy Thiemann-Bourque. Support for AAC Use in Preschool, and Growth in Language Skills, for Young Children with Developmental Disabilities. *Augmentative and alternative communication (Baltimore, Md. : 1985)*, 29(4):334–346, December 2013.

[3] Bright Big Red Switch to Increase Visual Perception. https://enablingdevices.com/product/bright-red-switch/.

[4] Austin M. Mulloy, Cindy Gevarter, Megan Hopkins, Kevin S. Sutherland, and Sathiyaprakash T. Ramdoss. Assistive Technology for Students with Visual Impairments and Blindness. In Giulio E. Lancioni and Nirbhay N. Singh, editors, *Assistive Technologies for People with Diverse Abilities*, Autism and Child Psychopathology Series, pages 113–156. Springer, New York, NY, 2014.

[5] Saira Saleem and Shahida Sajjad. The Scope of Assistive Technology in Learning Process of Students with Blindness. *International Journal of Special Education*, 31(1):46–54, 2016.

[6] Meng Ee Wong and Janet S. P. Law. Practices of Assistive Technology Implementation and Facilitation: Experiences of Teachers of Students with Visual Impairments in Singapore. *Journal of Visual Impairment & Blindness*, 110(3):195–200, May 2016.

[7] Nonverbal Communication in Children Who Are Blind or Visually Impaired. https://familyconnect.org/multiple-disabilities/communication/delayed-communication/nonverbal-communication-in-children-who-are-blind-or-visually-impaired/.

[8] Ales Berger, Andrea Vokalova, Filip Maly, and Petra Poulova. Google Glass Used as Assistive Technology Its Utilization for Blind and Visually Impaired People. In Muhammad Younas, Irfan Awan, and Irena Holubova, editors, *Mobile Web and Intelligent Information Systems*, Lecture Notes in Computer Science, pages 70–82, Cham, 2017. Springer International Publishing.

[9] Homepage. https://iamhable.com/home.

[10] Cássia Cristiane de Freitas Alves, Gelse Beatriz Martins Monteiro, Suzana Rabello, Maria Elisabete Rodrigues Freire Gasparetto, and Keila Monteiro de Carvalho. Assistive technology applied to education of students with visual impairment. *Revista Panamericana De Salud Publica = Pan American Journal of Public Health*, 26(2):148–152, August 2009.

[11] Augmentative and Alternative Communication (AAC). American Speech-Language-Hearing Association. https://www.asha.org/public/speech/disorders/aac/.

[12] Rúbia E. O. Schultz Ascari, Roberto Pereira, and Luciano Silva. Computer Vision-based Methodology to Improve Interaction for People with Motor and Speech Impairment. *ACM Transactions on Accessible Computing*, 13(4):14:1–14:33, October 2020.

[13] Fernando H. F. Botelho. Childhood and Assistive Technology: Growing with opportunity, developing with technology. *Assistive technology: the official journal of RESNA*, 33(sup1):87–93, December 2021.

[14] Brian Burne, Valerie Knafelc, Maureen Melonis, and Patricia C. Heyn. The use and application of assistive technology to promote literacy in early childhood: a systematic review. *Disability and Rehabilitation. Assistive Technology*, 6(3):207–213, 2011.

[15] David Trembath, Susan Balandin, Leanne Togher, and Roger J. Stancliffe. Peer-mediated teaching and augmentative and alternative communication for preschool-aged children with autism. *Journal of Intellectual & Developmental Disability*, 34(2):173–186, June 2009.

[16] Pull-up Resistors. https://learn.sparkfun.com/tutorials/pull-up-resistors/all.

[17] Reference. Apple Developer Library.

[18] Apple Developer Documentation.