# ALEC: Audio Learning and English Companion

## 1 Statement of Work

### 1.1 Molly Gibson

As a co-lead for keyboard development, I was first responsible for the complete hardware design of the keyboard. I led the design and implementation of the custom keyboard, starting with selecting and sourcing components tailored to the needs of our application. I designed the PCB layout and schematic, incorporating features such as the key-switch matrix, pull-up resistors, multiplexer, power regulation, and diodes to prevent ghosting and key-blocking. I soldered and assembled the keyboard components, ensuring all connections were functional and optimized for durability and ease of use. In addition to hardware assembly, I designed the keyboard to be visually engaging for young learners after it was printed, implementing Spongebob keycaps and painting it so that it appeared very child-friendly. For another hardware component, I also designed the simple LED circuit schematic and layout for providing visual feedback to the user.

For the software side of the project, I helped implement key features for the keyboard functionality, including embedded C code to manage matrix scanning, key mapping, and debouncing. I developed specific application-level features such as functionality for special keys, including "Start", "Help", "Delete", and "Enter", to enhance the user experience. A major challenge I addressed was audio timing issues caused by simultaneous processes within the microcontroller. I resolved this by implementing atomic operations and carefully managing global variables, ensuring concurrency and smooth interaction between the keyboard and audio playback systems. This resulted in reduced delays and improving overall performance and detection of key presses. Lastly, I collaborated with the team to integrate the keyboard with other subsystems, including the LCD screen, speaker, and LEDs. I helped debug potential problems with the overall system, addressing edge cases in the code and any potential problems in hardware connections.

### 1.2 Phi Lu

As a co-lead for audio playback development, I was responsible for designing and implementing the audio system for our project. I wrote embedded C code to manage the playback of audio files stored on a USB flash drive, ensuring compatibility with the STM32F407G1-DISC1 microcontroller and Digital-to-Analog Converter (DAC). My work included uploading audio files for three languages to the flash drive and developing system workflows to facilitate communication between the flash drive, DAC, and microcontroller. This required creating efficient methods for iterating through audio files based on user interactions and integrating these flows into the overall system design.

Additionally, I conducted unit and system-level testing to verify the functionality of the audio subsystem before its integration with the keyboard. This included debugging edge cases, such as scenarios where the flash drive was unplugged mid-playback, which could disrupt the system's operation. I implemented solutions to handle these cases gracefully, ensuring reliability and user satisfaction. My contributions played a pivotal role in delivering a robust and functional audio system, which forms a key component of the project's user experience.

**1.3 Daniel Sarria**

As a co-lead for keyboard development, I integrated the keyboard hardware with the embedded system. I wrote embedded C code to implement a robust matrix scanning mechanism, leveraging a hardware timer interrupt for row scanning and polling for columns to ensure no key press was missed. I also implemented functions for the delete key and spacebar, enabling the deletion of characters and cursor movement for added functionality. Additionally, I created the character mapping for the keyboard to ensure each letter was displayed on the screen using a 2D array of characters. I also integrated the keyboard with an LCD screen to display characters for user interaction and facilitate debugging. Using an existing LCD driver, I wrote embedded code to initialize the LCD in 4-bit mode with a blinking cursor. I also implemented multiple scenes for the LCD screen, such as the choose language screen and the start screen. Lastly, I refactored code to improve readability and debugging. On the hardware side, I installed keyboard stabilizers and managed the wiring for the peripherals to ensure a clean and organized setup. I also resolved edge cases related to USB drive removal.

Additionally, I also conducted unit and system-level testing to verify the functionality of the keyboard subsystem before its integration with the audio system. This includes cases such as repeated characters, missed inputs from a key press, and incorrect character outputs on the screen. I helped implement solutions to handle these cases carefully, ensuring proper functionality to provide our users with a great user experience.

**1.4 Andrew Vithoulkas**

As a co-lead on the audio playback development for the project, I took charge of both integrating and designing the audio system. My contributions included writing embedded C code to enable the playback of .wav files stored on a USB drive. This required a deep dive into the STM32F407G1-DISC1 microcontroller and its interfacing with the Digital-to-Analog Converter (DAC) to ensure high-quality audio output. To manage audio data efficiently, I developed a file management system capable of mapping specific .wav files to corresponding text-based word entries. This involved implementing a mechanism to read .txt files from the USB flash drive, parsing their content, and matching them against user input for precise file retrieval. I also played a pivotal role in integrating the audio system components, ensuring smooth communication between hardware and software elements, and optimizing the user experience with the interaction tool we developed.

Beyond the technical aspects, I spearheaded the physical design of the system. Leveraging my expertise in 3D printing, I took on the responsibility of creating the device's external case. This involved detailed measurement, CAD modeling, and prototyping to ensure the enclosure was both functional and aesthetically pleasing. Once the design was finalized, I led the printing process and assembled the components to deliver a polished, ready-to-use device. My work combined technical innovation with hands-on design to create a fully functional and user-friendly audio playback system that met the project's goals.

# 2 Table of Contents

## 2.1 Table of Figures

## 3 Abstract

The Audio Learning and English Companion project, also known as ALEC, aims to develop an offline, interactive tool to enhance English vocabulary and comprehension skills for non-native English-speaking primary school-aged children without internet access. The device integrates a custom-made keyboard, XMSJSIY Mini Computer Speaker, and a Midas Displays LCD screen controlled by an STM32F407G1-DISC1 microcontroller all in a 3D-printed plastic case, printed from PLA filament. This tool allows the users to practice listening comprehension exercises for single-word translations and typing on a keyboard without the need for an internet connection. The project addresses the inequality of language access, particularly in underfunded, non-English-speaking communities, by offering a low-cost solution for improving English listening comprehension skills offline.

## 4 Background

The English language is the official language of 57 sovereign countries, making it the most popular official language in the world [10]. It is the dominant language in countries such as the United States, the United Kingdom, Australia, Canada, and several other nations. English also holds the status of an official language in countries like India and South Africa, making it a crucial medium of communication in international business, science, and diplomacy. Approximately 1.5 billion people worldwide are able to speak English [4]. However, not everyone can speak English fluently. In fact, in 2021, approximately 5% of school-age children both spoke a language other than English at home and had difficulty speaking English [3]. This is highlighted below in Figure 1:

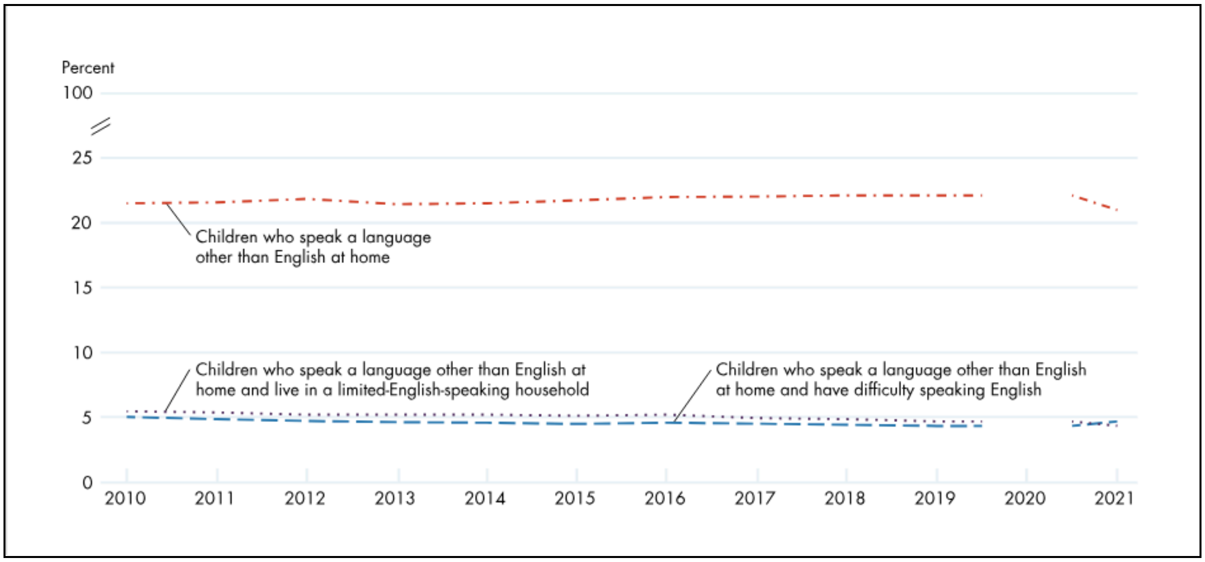



*Figure 1. Percentage of children ages 5–17 who speak a language other than English at home and who have difficulty speaking English or live in a limited-English-speaking household, 2010–2021*

Projects have been developed to improve English comprehension among children, such as online platforms like Duolingo and Rosetta Stone, which offer English learning listening exercises, but these rely on constant internet connectivity, limiting their use in offline home settings. This is important because two thirds of the world's school-age children have no internet access at home [9]. Existing offline tools such as the Speak and Spell, a popular educational tool from the late 1970s and 1980s, have outdated features, such as limited vocabulary, lack of customization for other languages, and minimal feedback, compared to the modern learning tools mentioned above.

The motivation behind ALEC stems from the growing need for accessible language-learning tools, particularly for non-native English-speaking primary school children who may not always have sufficient English speakers at home and have unreliable internet access. This tool aims to provide offline listening exercises to enhance English comprehension skills, which is crucial for English learners in remote or under-resourced areas. What sets this project apart is its ability to function entirely without an internet connection and to provide a direct translation from a user's native language to the English language to improve comprehension, while still providing interactive feedback to users through the use of LEDs and an embedded system.

Our coursework in Introduction to Embedded Computer Systems has equipped us with hands-on experience in using the STM32 microcontroller (MCU) and programming in C to manage user input and output. This will specifically apply to the custom-designed keyboard where each key needs to be handled appropriately using embedded C code on the MCU, as well as the speaker system where specific audio files need to be chosen from the correct directory and sent to the speaker to convey speech to the user. Additionally, the knowledge gained from the Electrical and Computer Engineering Fundamentals series has provided us with the skills to design appropriate circuits, handle audio signal processing, and troubleshooting. This applies to a custom keyboard we will make where PCB design is needed to send signals to the MCU and the speaker system that will be used where signal processing needs to be applied to send audio from the MCU to the speaker.

## 5 Project Description

### 5.1 Performance Objectives and Specifications

ALEC aims to support primary school children who are non-native English speakers in learning vocabulary spelling and translations, especially in environments without internet access or exposure to English speakers. The tool will feature a custom-made keyboard designed for user-friendly input, XMSJSIY Mini Computer Speakers that pronounce phrases in the user’s native language and translate them into English, and an alphanumeric 20x4 LCD screen from Midas Displays to display the user's input along with yellow and green feedback LEDs to signal correct or incorrect inputs. The overall design prototype can be seen below in Figure 2. An STM32F407G-DISC1 microcontroller will integrate the keyboard, speaker, and screen into a unified system. The custom-made keyboard will include all 26 letters of the English alphabet, along with start, end, help, delete, and enter keys for ease of use. The functioning of these buttons aligns with our goals to make an easy-to-use listening comprehension aid for younger students. The speaker will be driven by a 12-bit Digital to Analog Converter (DAC) with an internal output buffer for sound clarity to ensure words are spoken clearly to be interpreted. Lastly, the MCU and the dual stereo speakers will be powered by a Voltaic Systems V25 always-on battery power bank. These specifications did not change from our proposal.

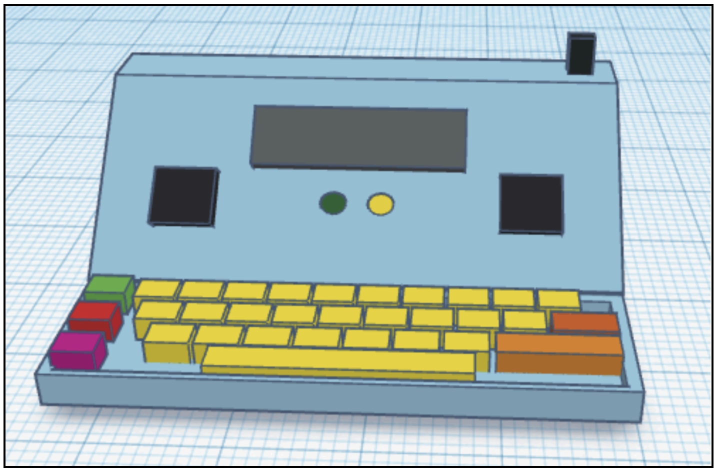


*Figure 2. Basic CAD Design*

**5.2 How It Works**

**Hardware:**

The keyboard is designed as a custom key-switch matrix with columns driven high to 3.3V VCC. A multiplexer (MUX) determines the active column, with its address pins (S0-S3) configured as GPIO inputs and its COM pin configured as a GPIO output to the STM microcontroller. Rows are configured as GPIO outputs, all driven high except one at a time during scanning. The STM32 scans the matrix using a hardware timer interrupt, sequentially driving each row low. When a key is pressed, it connects the corresponding row and column, shorting the column to logic 0, which is read by the microcontroller via the COM pin of the MUX. This setup allows the row and column addresses to map to 32 specific characters or function keys, processed through embedded logic for the user program. Figure 3 illustrates the key-switch matrix schematic, while Figure 4 shows the MUX schematic and global connection configuration.

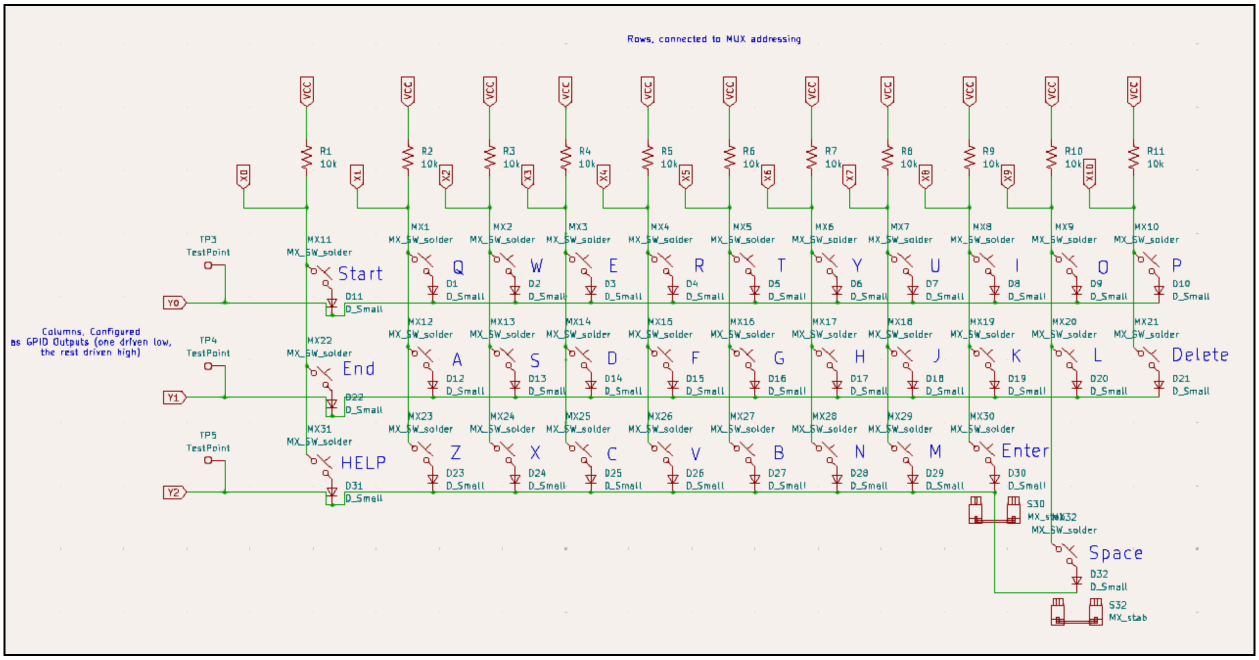



*Figure 3. Key-Switch Matrix Schematic*

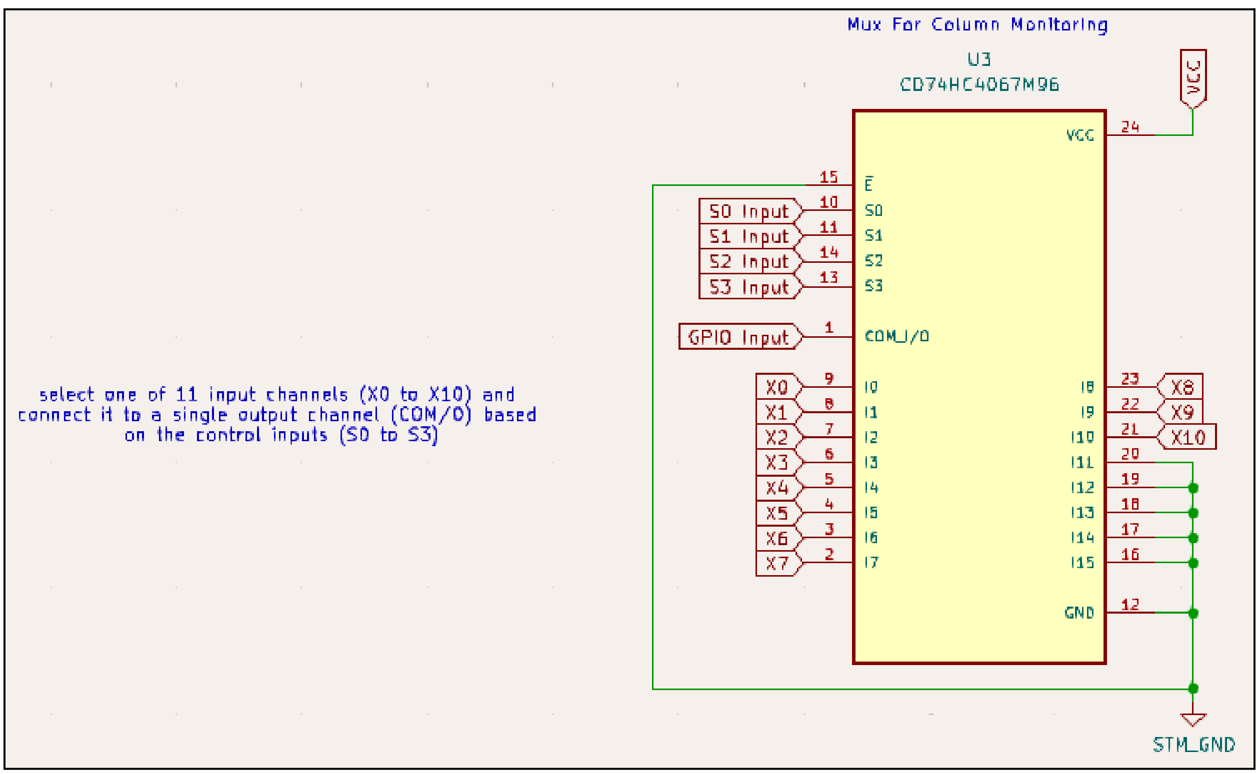
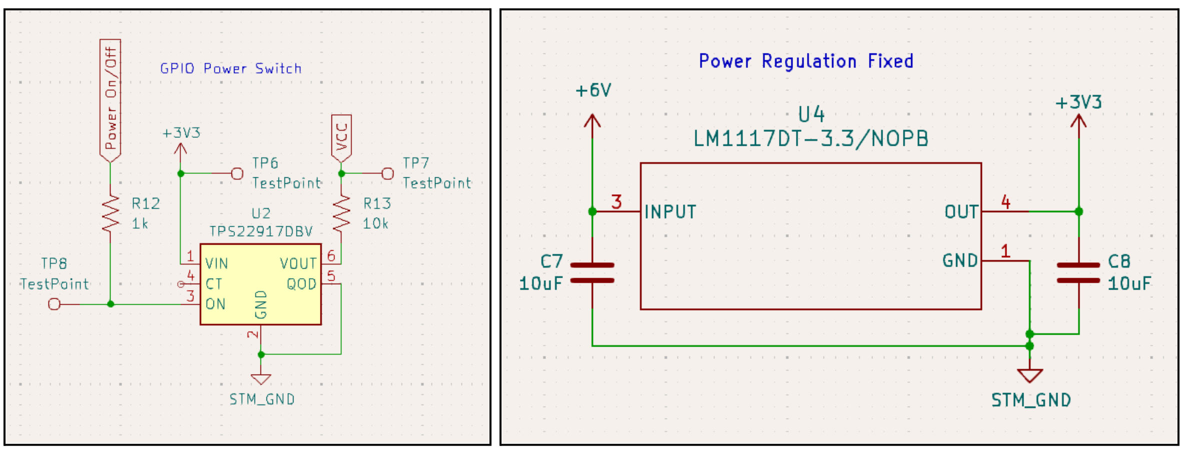



*Figure 4. Multiplexer Schematic and Global Connection Configuration*

The keyboard PCB includes three input connectors for rows, four inputs for addressing columns through the MUX, and a common input/output port for column detection (COM). Additional inputs include one for the common ground connection to both the STM32 and power supply and another configured as a GPIO output for enabling or disabling the power switch. The power supply is fixed as 3.3V from 5V portable charging using a linear regulator. The GPIO power switch pin provides power to the key-switch matrix when logic 1 is set, otherwise it is disabled with logic 0. Key columns are held at VCC with 10kΩ pull-up resistors, while diodes between switches help prevent ghosting or key-blocking during simultaneous key press events. Schematics for the GPIO power switch and 3.3V linear regulator are shown in Figures 5 and 6, respectively. The keyboard PCB layout is depicted in Figure 7. A 3D view of the keyboard PCB from an alternate angle is presented in Figure 8.




*Figures 5-6. GPIO Power Switch and 3.3V Linear Power Regulator Schematics*

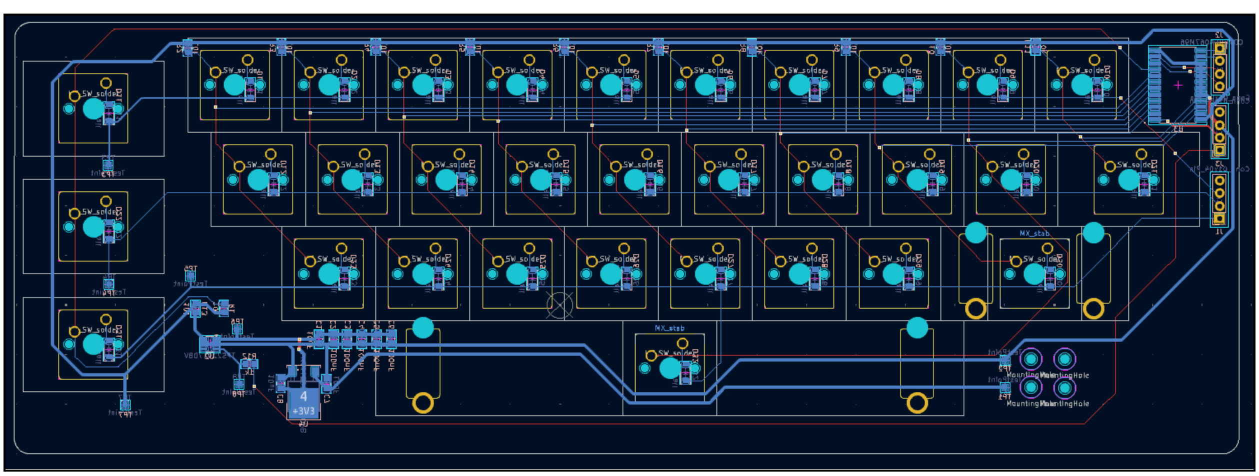


*Figure 7. Keyboard PCB Layout*

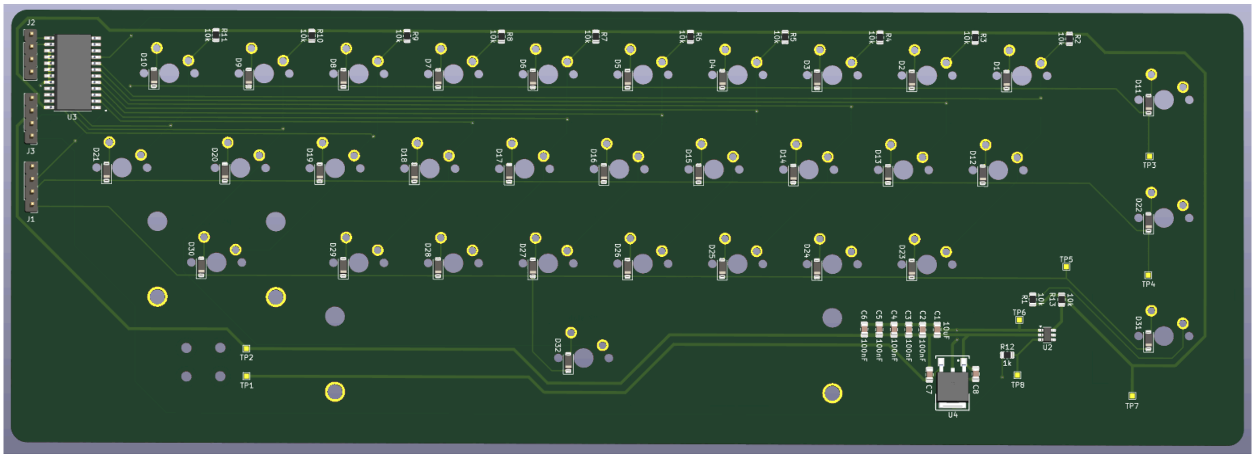
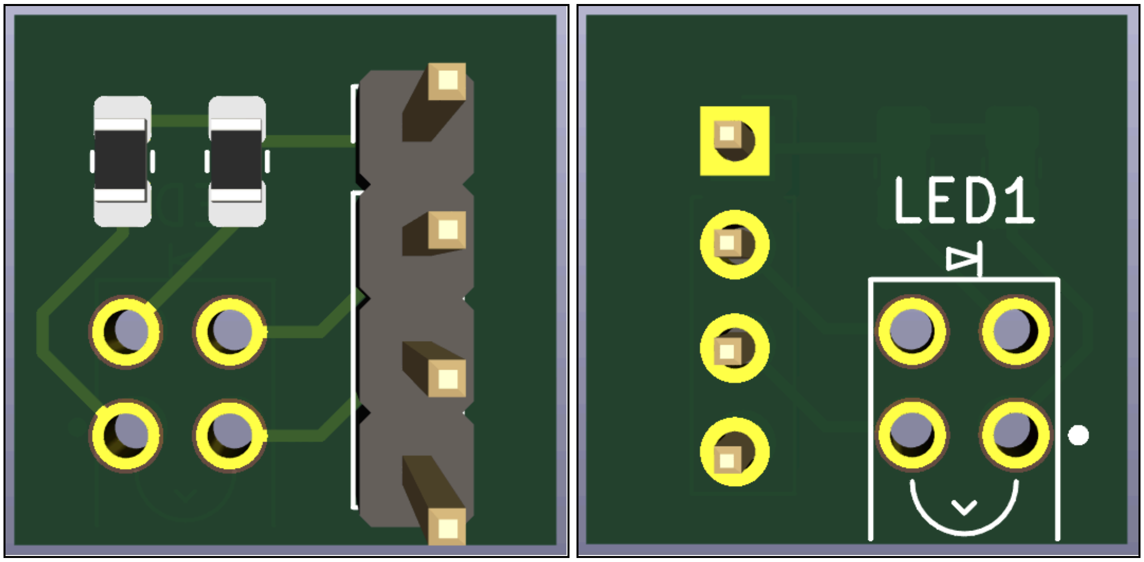


*Figure 8. Keyboard PCB 3D View, Alternate Side*

The speaker system connects to the STM32F407G-DISC1 microcontroller via an auxiliary port, utilizing its 12-bit buffered DAC channels to convert digital signals to analog outputs. The microcontroller features an integrated Class D speaker driver, allowing compatibility with most speakers using an auxiliary cable. PCM audio data is transmitted from the USB drive to the DAC using C, I2C, and I2S protocols.

The USB drive connected via the OTG USB port on the STM32 stores directories and audio files for different native languages and words. The system selects the appropriate files based on user input during the initial language selection process.

The system provides immediate feedback to users via LEDs. A green LED lights up to indicate correct input, while a yellow LED signals incorrect input, enhancing the interactive experience. The simple circuit for the LED feedback system is shown in Figures 9 and 10, which depict 3D front and back views of the PCB. Resistor values of 68Ω were used for both LEDs, as specified by their ~2V forward voltage drop and 5V input.




*Figures 9-10. LED Simple Circuit PCB 3D Front and Back Views*

Volume adjustments are managed through the on-board speaker controls, providing flexibility to users. Additionally, a physical power switch allows the user to turn to the device on or off.

**Software:**

The STM32 microcontroller scans the key-switch matrix using a hardware timer interrupt. As rows are driven low sequentially, key presses connect rows and columns, which are read as logic 0. The corresponding addresses are mapped to characters or special function keys processed by the embedded program. Using C programming in STM32CubeIDE, audio data in PCM format is read from the USB drive and transmitted to the DAC. Proper I2C and I2S settings enable the microcontroller to play audio through the speaker, providing instructions and feedback to the user. A custom file system coded in C organized the audio files by language directories. During the initial selection, the user chooses a native language, prompting the system to load the corresponding directory and files from the USB drive.

The user interacts with the system through various buttons. After selecting a language via the startup instructions, the "Start" button begins the application. The "Help" button repeats audio cues and provides spelling hints when pressed multiple times. The "End" button stops the session, reveals the score or number of words correctly entered, and returns the user to the language selection screen. Upon typing a word, correct spelling results in positive feedback via a green LED and an audio cue. Incorrect spelling prompts a yellow LED and allows the user to retry, eventually revealing the correct word after five failed attempts.

The system uses the LCD screen to display the user's typed words, hints, and other prompts. LED feedback reinforces learning by providing immediate visual cues for correct or incorrect inputs. When the USB drive is removed, the system halts audio playback, disables the keyboard, and resets the initial language selection screen. Reinserting the USB restores functionality and allows the user to resume.

For overall system control flow, the user begins by selecting their native language from a menu on the LCD screen. After pressing the "Start" button, the system guides the user through a series of words. Audio instructions are played, and user input is evaluated. Correct spelling triggers the next word, while incorrect spelling provides feedback and additional hints. The system ensures seamless operation and adaptability through its integrated feedback mechanisms. An overview of the high-level system design and initial control flow is provided in Figures 11 and 12, respectively.

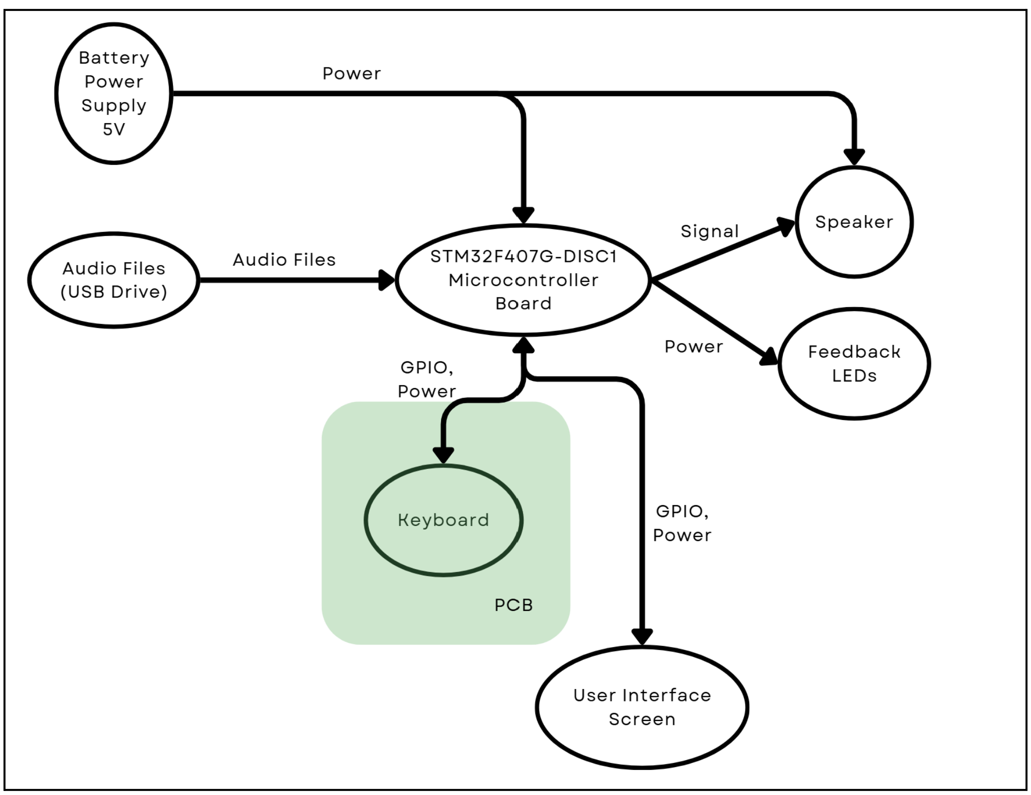


*Figure 11. High-Level Overview of System*

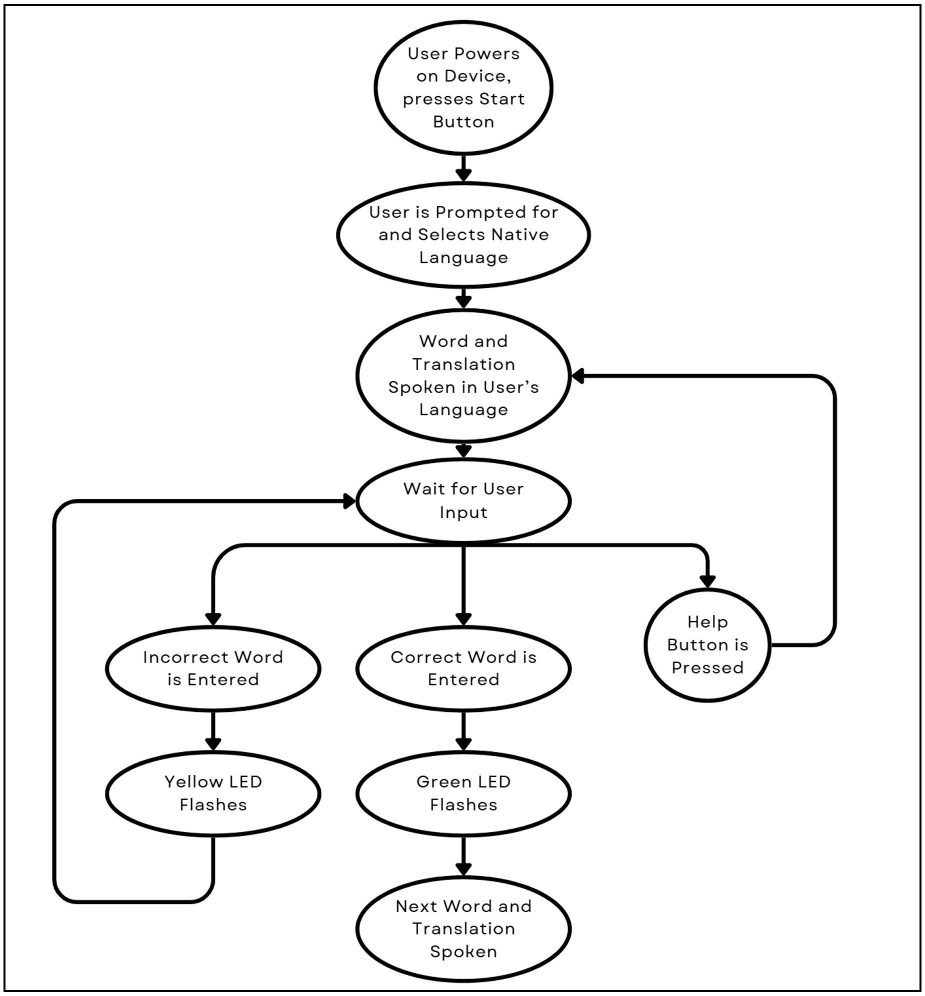



*Figure 12. General Control Flow*

**5.3 Technical Details**

The STM32F407G-DISC1 microcontroller is an ideal choice due to its availability of around 100 pins, which are essential for interfacing with the custom-designed keyboard, the user interface screen, and the speaker system [8]. This allows us to utilize the MCU as the main power source for all of the peripherals. Moreover, the MCU's built-in audio DAC, integrated Class D speaker driver, and stereo headphone output jack [8] make it well-suited for audio signal processing. The MCU also features 4 user LEDs and an on-board ST-LINK/V2-A debugger/programmer, which can be leveraged for debugging [8]. The MCU also comes equipped with 2 5V pins and 2 3V pins that would allow us to power all of our deliverables from the MCU alone, as long as the MCU is supplied with power. Additionally, it is fully compatible with the STM32CubeIDE, an IDE we are already familiar with, making development more streamlined.

The XMSJSIY Mini Computer Speaker is a dual stereo speaker that works well for our project because of its size, availability, and the ability to adjust the volume. Given its limited dimensions, it will fit well on our 3D-printed design on both sides of the screen. Its main important feature is that it is fed audio signal by a normal auxiliary jack, which our board outputs, making configuration between the two components extremely easy. Not only this, but it works well within our budget to give us the performance we need. Coming from a supplier like Amazon, availability and order times should not be an issue if the speaker stops functioning and we need another. The speaker is also powered from a USB type A cable, allowing us to power it straight from the battery pack, furthering the convenience of applying it to our design.

We wanted a design that used a keyboard as opposed to a touchscreen for user input primarily because we wanted our target demographic to get more practice on a keyboard, as it is vital for their future success in a digital world [1]. We chose to design our own keyboard for several different reasons. Primarily, we did not need the added complexity of a full keyboard, as the purpose of our keyboard is to aid the user in spelling words. Therefore, numbers and other characters, such as punctuation marks, were not needed. While it is also true that it is very difficult to find reduced keyboards with only letters, we wanted the added benefit of being able to include our own buttons for the purpose of the listening comprehension aid, such as start, stop, and help buttons. Additionally, designing our own keyboard would allow us to use the Kailh Mechanical Key Switches, which are ideal because they provide clear tactile feedback when pressed, which can help kids understand when a keypress has registered. Also, these switches are designed to withstand heavy usage, which is ideal for kids who might press keys with more force or use them frequently. Lastly, the use of the Akko SpongeBob Keycap Set features bright, vibrant colors and familiar characters from SpongeBob SquarePants, a show beloved by many children. This makes the keyboard visually exciting and can capture kids' attention, making learning or interaction more engaging.

The LCD display and LEDs are other important components, as they provide feedback to the user along the way to aid the learning in practice process. We want the kids to know when they are making the correct decision, and feedback, even as simple as a light turning on, is an incredibly important part of the learning process [2]. The alphanumeric 20x4 LCD display from Midas Displays that we have chosen is relatively inexpensive, widely available, and is large enough to display multiple lines of characters to the user. Additionally, it is fully configurable with 4-bit or 8-bit mode to display characters using a microcontroller. Although it is possible for our project to use a graphic display, it is much more feasible and sensible to utilize an alphanumeric display because our project only involves characters. The LEDs were easy to purchase and work with, as they are also widely available and inexpensive. They can be integrated into our design very flexibly and worked perfectly for our needs. A small PCB was designed and manufactured, attaching the LEDs to the case along with 68Ω resistors to ensure the LEDs received the correct amount of current. This was wired to the microcontroller board to deliver 5V to the LEDs.

The MCU is powered by a Voltaic Systems V25 always-on battery power bank, connected through a USB power switch. This setup allows users to conveniently turn the device on and off while ensuring a consistent and reliable power supply to the microcontroller, eliminating the need for manual intervention with the battery pack. Unlike most battery packs, which shut off when there is no current draw, this power bank remains active. This feature is crucial because, with an ordinary battery pack, turning off the power switch on our device would eventually cause the battery pack to shut down completely, requiring manual intervention to turn it back on. Additionally, users can recharge the power bank via a USB-A connection when its battery is depleted. The Voltaic Systems power bank is ideal for our application as it is specifically designed for microcontrollers requiring an always-on power source, effectively addressing potential issues with power interruptions. This always-on capability is particularly important for our target audience of children, who may inadvertently leave the device on for extended periods. By incorporating this power solution, we ensure uninterrupted operation and provide a user-friendly design that minimizes the need for frequent manual resets or battery changes, making it both practical and durable for everyday use.

The external case for this learning tool was designed using Autodesk Inventor. It was then 3D printed on Ultimaker 3D printers using PLA filament. This type of filament is very inexpensive and easy to use for prototyping, as it sticks to the 3D printer's build plate much easier than other types of filament. This made it the ideal choice to create this prototype. The case is printed in separate pieces to fit in the 3D printer's print area. Once the pieces were printed, they were attached using bolts on pieces that should be taken apart for repairs (e.g. the keyboard) and J-B Weld on pieces that needed a permanent connection (e.g. the main case). J-B Weld is a specialized, high-temperature epoxy adhesive that creates a permanent bond between surfaces. This ensured that the external case best fit all parts for ease of interaction by the user. Since our target user is children, we want to make sure that the case fits together robustly, which is why we chose J-B Weld. To make the case attractive for the children, we spray painted it with Island Girl COLORSHOT paint. This gives the case a softer blue color that is more attractive than the printed grey and allows us to use acrylics to paint flowers on it. This was heavily influenced by Spongebob, a popular kid's show today. This gives the target user a familiar look, increasing comfortability for the first interaction with the tool.

Much of our design can be attributed to Texas Instruments' original Speak and Spell design. In fact, Texas Instruments' original design consisted of an integrated circuit that included a controller chip (shown below in Figure 13), which is similar to our use of the STM32F407G. It also had an external speaker and a display, which will also be utilized in our design. However, Texas Instruments' implementation of the audio system was much more complex. It had a synthesizer to manipulate electrical signals to sound, word storage ROM to store a multitude of words, and a solid state speech module [6]. In comparison, our design will utilize a USB drive as the form of audio storage, which will already contain the audio samples of the words. This USB drive will help us and the user by allowing us to potentially put different words in each USB drive, furthering access to the language. Our product allows kids who are new to English to be able to practice listening comprehension and spelling words in the English language with the translation from their original language. After all, word association practice is an extremely important part of learning a new language [7]. In order for the user to know how to internally translate words, it must be practiced.

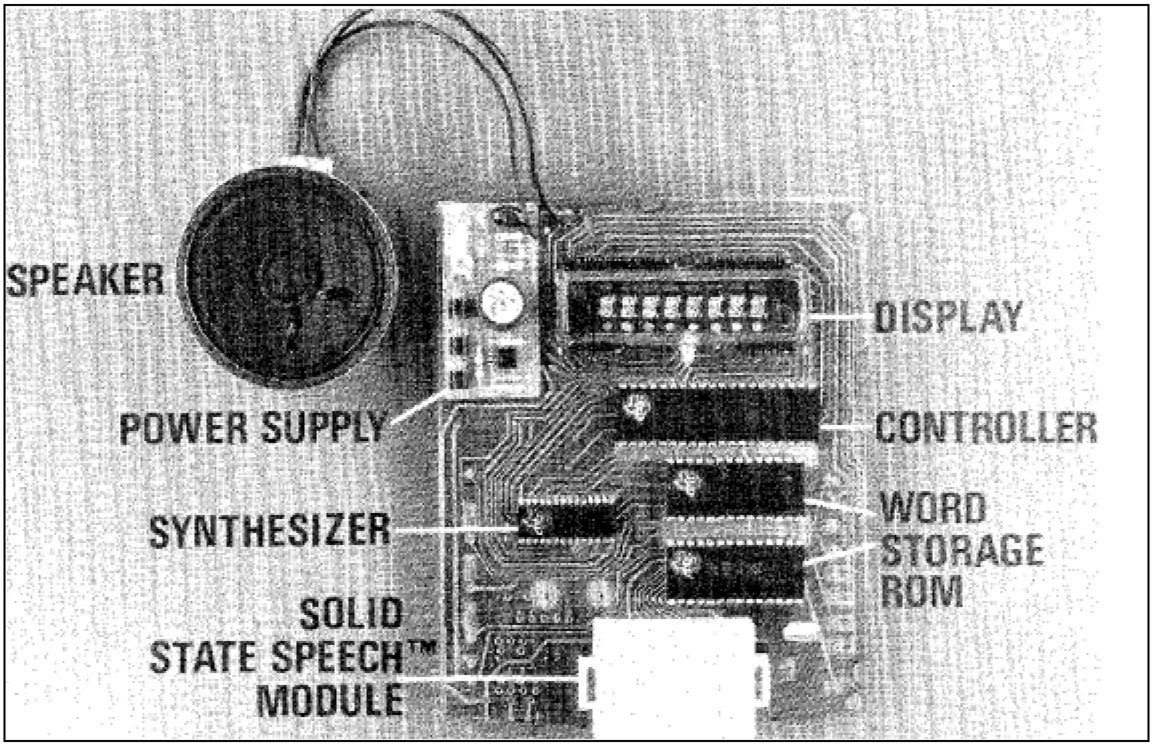



*Figure 13. Texas Instruments Original Design of Speak and Spell*

**5.4 Test Plans**

In order to verify the functionality of this device, the parts underwent block testing to ensure each part worked before assembling the system. Both the audio system and the personalized keyboard were created and tested in parallel.

The keyboard was tested using the MCU. Originally, the MCU was going to run embedded C code using external interrupts to detect each keystroke that is pressed, which is outputted to the console using UART. This was to be repeated for every button on the keyboard to ensure every input is detected. However, after thorough inspection of the MCU datasheet, it was discovered that the MCU utilized a virtual COM port. This meant that the only feasible way to utilize UART on the MCU was to obtain external hardware to connect USB on our computers to the UART port pins on the MCU. To maneuver around this problem, we simply utilized a multimeter to measure the test points on each key switch. Specifically, we measured the voltage drop from 3.3V to 0V when the key switch is pressed because this indicates a key press in the corresponding row and column in the matrix.

The audio system was tested by running a portion of test code that is written to fetch a specific language audio file. The file system chooses the correct audio file to translate from and automatically selects the English language audio file as well. The selected audio file then was outputted to the speaker where the selected audio file is spoken first, followed by the corresponding English translation. In the early stages of audio playback, a button press was used to mimic the user typing words and pressing 'Enter'. After an audio file would play, the DAC would be turned off. Once the button was pressed, the next audio file was selected and played through the speakers. This allowed us to ensure that we could jump from audio file to audio file in between words quickly.

The feedback components (LEDs and LCD screen) were then tested only after the keyboard passed all testing. The LCD screen was tested with the keyboard and the MCU. For each keystroke pressed on the keyboard, the LCD screen outputted the correct character. This was repeated for every key on the keyboard to ensure the LCD screen displays the correct output. The character mapping was coded in the STM32Cube IDE using a 2D array. After initial testing of the keyboard with the LCD screen, we discovered that the keyboard had a repeat rate that was too fast when pressing a key for a very short time. As a result, one design change we made on the keyboard was to only register the key press after the key was released. This prevents any repeated letters from occurring when the key is held down for any amount of time and prevents children, who tend to be slow typers, from spamming letters on the screen accidentally. Additionally, this allows children to slowly take their time spelling each of the words that they listen to.

The LEDs were tested with the keyboard and the MCU. Initially, the design was going to utilize a green and red LED. However, after further consideration, we believed having a red LED to indicate an incorrect answer would be too harsh for children. As a result, we decided to modify the red LED to a yellow LED instead for a softer response. Originally, for every correct keystroke pressed, the green LED should be activated, and for every incorrect keystroke, the yellow LED should be activated. However, we decided that it would be much easier to indicate LED feedback after the user presses the enter button on the keyboard to indicate that the user has finished spelling the word. This was accomplished by creating a portion of test code that sends a sample word to spell, such as the word 'Hello'. If the correct word is entered on the keyboard and the user presses the enter button, the green LED lights up and the yellow LED remains off. If the incorrect word is entered on the keyboard and the user presses the enter button, the yellow LED should light up and the green LED should remain off.

After each component was tested individually, the device was tested as a whole before assembly. This meant that when we started the game, an audio file was played, and the user had to spell the word correctly to move forward, just as the game was defined. We ran into several timing issues, resolved with atomic variables, which will be discussed in a later section. Once these issues were resolved, the game worked very well on the full scale. After this, the device could be assembled and painted. A team member took ALEC to young students that were family friends learning English and keyboarding to test the device. The initial plan was to test in a team member's elementary school, but this did not work, as the elementary school would not allow data collection without a rigorous application that would be looked at after the capstone course was completed. Because of this, testing in young children's homes was ideal, since this is the environment that the tool is meant to be used in.

## 6 Physical Constraints

### 6.1 Design and Manufacturing Constraints

One of the main design constraints that our project faced is CPU limitations on the MCU. The CPU of the MCU needed to be able to handle scanning the matrix of the keyboard, register characters on the screen, update the screen to new scenes, read and output the correct word file for the audio system, and emit the correct LED. These processes can simultaneously happen at once which can lead to timing issues. For instance, pressing certain buttons on the keyboard may not register immediately, which interrupts the learning process. One way to solve such timing issues was the utilization of FreeRTOS to create different threads for each process. However, no one in our team has any experience with FreeRTOS. As a result, we utilized atomic variables to resolve timing issues. Atomic variables ensure that operations on shared data are completed as a single, indivisible unit, preventing race conditions and ensuring data consistency even when multiple processes access or modify the same global variables. In our application, we used these variables to synchronize shared resources, prevent interrupt interference, efficiently manage flags, and avoid delays caused by locking mechanisms.

Another constraint our project faced was with naming the audio files on the USB drive. During initial testing, we realized that the FAT File System that we used on the STM32 only allows for an 8 character file name limit on audio files. This means that we had to keep the file names down to just a few characters. To get around this, we numbered each .wav file, such as

"1.wav," "2.wav," etc. While this may not seem like a large deal, it is a constraint that we faced on naming our files to pull from, as we had to create a file manager to know which audio file had which word on it.

Part availability is a constraint that is difficult to get around. In our project, we initially looked for LEDs that were larger than the ones we ended up with, but with none of them being in stock, we had to end up with the smaller LEDs that were available. This ensured that we could get them in time to finish by the end of the semester.

A final constraint we faced during the manufacturing process was 3D printing the case. The 3D printer we had access to could not fit the entire case on the build plate, so we had to get around this by cutting up the design in software and printing it in pieces. This led us to get more creative with bolt placement so that the entire case could be assembled correctly.

**6.2 Tools Utilized**

KiCad was utilized for PCB production. Similar to our use of Multisim and Ultiboard in the past, this tool served as a schematic design and PCB layout tool for the keyboard and LED simple circuits. The schematic editor allowed us to create detailed and accurate circuit diagrams. The PCB layout editor enabled precise placement of components and routing traces along with the tool's Design Rule Check functionality to ensure the layout met manufacturing constraints for final production. KiCad's 3D viewer provided a visual representation of the PCB and its generation of industry-standard Gerber files were used by the manufacturer (JLCPCB) to fabricate the PCBs.

To aid in the KiCad development of the keyboard PCB, Keyboard Layout Editor (KLE) was used to facilitate the spatial design of the keyboard layout. It allowed us to specify the position, size, labeling, and coloring of each key. KLE allows users to explore the keyboard layout in JSON format. Using this data from the KLE export and importing the necessary part libraries for switches, KiCad was able to map the keys onto the PCB layout. This ensured the physical arrangement of the keys matched the intended design.

The STM32CubeIDE served as the primary development environment for writing and flashing embedded code to the MCU. It was seamlessly integrated with our GitHub repository to facilitate team collaboration. Through this process, we gained proficiency in using the IDE's user interface for pushing, pulling, and merging code within our project repository. Additionally, we learned to organize the extensive codebase effectively and refactor it to maintain clean, structured, and efficient code. Lastly, we utilized the IDE to debug and determine errors in our source code.

The C programming language was the primary language used to write code in the STM32CubeIDE. Using C enabled us to develop low-level embedded code, providing precise control over hardware resources. Careful attention was given to optimizing code efficiency, managing memory usage, and ensuring proper timing for critical processes. This approach allowed us to achieve robust and reliable functionality while working within the constraints of the microcontroller.

GitHub was the primary version control system used to enable team collaboration within the STM32CubeIDE. This setup allowed each team member to maintain a local repository and resolve any code conflicts efficiently. We also learned how to integrate our Git repository into the STM32CubeIDE, enabling us to push, pull, and merge code seamlessly into the remote repository from the IDE.

Other tools utilized included those for the manufacturing of the case. The first of which was Autodesk Inventor. This software allowed us to model our case so that it could be 3D printed. Having a teammate with experience using this software meant that modeling and prototyping was straightforward and took place towards the end of the project. Once the design was modeled in Inventor, it was cut up and transferred to our next software tool, Ultimaker Cura, as a .stl file. Cura is a software that transfers .stl files .gcode files, which contain instructions for the printer that leads to the 3D print as we know it. This leads to the final tool in this process, the Ultimaker S3 3D Printer. This printer was free to access for us, and it took the .gcode file created from the Cura software combined with PLA filament to create the pieces of the case. The combination of these tools meant that manufacturing the case from 3D printing was straightforward.

**6.3 Cost Constraints**

The project's cost constraints are influenced by multiple components, including the STM32 microcontroller, keyboard PCB, mechanical switches, keycaps, stabilizers, dual stereo speakers, LCD screen, battery pack, LED PCB, and USB drives. A detailed projected cost for a single production unit is provided in Appendix A.

Other minor components were relatively inexpensive and had minimal impact on the overall budget. Notably, the keyboard keycaps and mechanical switches can be replaced with more affordable alternatives without compromising functionality. For instance, mechanical switches are available for as low as $12.00, and keycaps can be sourced for as little as $20.00. These substitutions can significantly reduce the total cost of the project.

**6.4 Production of the Prototype**

In order to create a production version of ALEC, several modifications would need to be implemented across the entire design, including materials, manufacturing processes, and scalability. The focus should be on simplifying the design, reducing costs, and improving hardware and software reliability.

The current prototype uses a 3D-printed casing, which, while functional, would need to be optimized for durability and scalability. A larger casing should be designed to securely house all internal components, reducing the risk of damage during daily use. Using higher-quality materials, such as ABS plastic, would improve the casing's structural integrity, making it better suited for repeated handling by children.

The keyboard PCB and keycaps would require reinforcement to withstand heavy use. Mechanical switches and stabilizers should be rated for durability, and the assembly should be secured to prevent damage from liquid exposure, drops, or frequent typing. Reinforcing these components would ensure the device's longevity in home environments.

For cost efficiency, components such as key switches, diodes, and LCD screens should be sourced from bulk producers. Transitioning from small-scale PCB production to large-scale manufacturing would involve ordering PCBs in bulk from manufacturers like JLCPCB or PCBWay. This change would significantly reduce unit costs and streamline the production process.

Before entering the market, the firmware should be further refined to handle edge cases gracefully, such as unexpected power loss, USB disconnections, or incomplete inputs. Improving the reliability of software will enhance the overall user experience and ensure that the device operates smoothly under various conditions.

To increase market reach, the production version should include support for additional languages and advanced educational levels. This expansion would allow the device to cater to a broader audience and grow with the user's skill level, increasing its value as a long-term educational tool.

By implementing these modifications, ALEC could transition from a prototype to a scalable product suitable for production. These enhancements would not only improve its functionality and durability but also align it with market and user needs, ensuring its success in providing accessible language education.

## 7 Societal Impacts

The project is aimed to help bridge the gap of educational inequality, particularly in low-income communities where access to the internet and English-speaking households is limited. By providing an interactive, offline, and affordable solution, the device seeks to empower young students to improve their foundational skills and English literacy, bridging the digital divide with respect to education. The primary stakeholders of this project are educators and parents or guardians within these communities, with the end-user being the students.

**Positive Impact:**

*Educational Equity and Accessibility*

The target audience, students who lack internet access and English-speaking environments, often face challenges in keeping up with peers due to lack of resources and a language barrier. In fact, ESL students are frequently marginalized within the school environment, often being viewed as "liabilities" rather than valuable contributors to classroom dynamics. The language barrier, coupled with a lack of high-quality learning materials, prevents them from fully participating in academic and social activities [5]. To combat this issue, the device serves as a low-cost tool that can be deployed with relative ease in both schools and homes. Therefore, in marginalized communities without access to high-quality educational materials, this device could serve as a vital resource. It is a step in the right direction in ensuring that all students, regardless of economic standing, can receive equitable learning opportunities.

*Global Relevance*

The device also holds global relevance, as it can be used as a model to serve developing nations and rural areas that face the same challenges. Around the world, there are children facing issues of inadequate educational infrastructure, language barriers, and limited access to the internet. The device has the potential for scalability, due to its affordability and easy distribution, and widespread implementation in these areas. In a later version, it has the potential to be adapted to support multilingual communities in assisting with the learning of languages other than English.

*Public Health and Safety*

An additional perk to the design and its offline use is it ensures the safety of the children who use it by removing the potential for exposure to inappropriate content, online predators, and sharing private information. This secure and controlled learning environment is crucial for young children who often do not have the knowledge to navigate the internet safely.

*Community and School Integration*

The device is designed to be easily integrated into schools and marginalized or underserved communities. Due to its compact size and affordability, it can be adopted as a good, low-cost, solution to enhance literacy programs. It can be distributed through libraries, local community centers, or non-profit organizations to work outside of formal education environments. Communities in need can utilize this to address educational inequity and support academic growth.

*Long-Term Economic Growth*

Finally, a common side effect of improved educational literacy is reduced poverty. By fostering educational programs for children in low-income communities, they are more likely to find higher-paying employment along with an increased range of job opportunities. In fact, education increases earnings by roughly 10% per each additional year of schooling [15]. In the long term, this should foster economic growth and development in disadvantaged areas.

**Potential Risk:**

*Production and Distribution Costs*

Even with affordability as a key feature, initial production, and distribution costs could be prohibitive for some schools or organizations. If the cost burden falls on low-income communities, it may strain already limited resources. Long term maintenance, such as replacing parts or addressing hardware issues, could also become a financial burden.

*Social and Cultural Concerns*

The focus on English literacy might inadvertently devalue other languages or dialects in communities where cultural preservation is a challenge. Also, in an increasingly digital world, reliance on offline tools could be seen as a "second-class" solution, potentially stigmatizing the communities that use them. Relying on external tools such as this could divert attention from systemic issues and broader policy reforms.

*Ethical Considerations*

While the device is offline, if future iterations include online capabilities for updates or tracking progress, there could be privacy concerns with user data or misuse. In cases where students are unsupervised, the device might be used in ways that detract from its educational purpose, such as repetitive use without meaningful engagement.

*Physical Hazards*

One significant hazard is the presence of sharp edges or protruding components, which could cause cuts or scrapes during use. Another hazard arises from detachable parts, such as keycaps or LEDs, which could pose a choking risk when ingested. Electrical safety is also a consideration. Exposed wires or damaged circuits could lead to electrical shocks or burns to the user. Similarly, prolonged use of electronic components might lead to overheating, potentially causing burns or fire hazards. Another potential hazard is the noise level of ALEC, which can cause harm to the user if noise levels exceed normal hearing thresholds. Lastly, if ALEC is dropped and damaged, fragile parts like the LCD screen or the casing might shatter, creating sharp debris. The rechargeable battery pack may also overheat, swell, or leak, posing fire, chemical, or explosion hazards. Compliance with regulations such ASTM F963 and EN71 should be prioritized for future production.

*Environmental Impacts*

Lastly, over time, discarded or non-functional devices could contribute to electronic waste. Additionally, manufacturing at scale could increase demand for raw materials, raising concerns about the ethical sourcing of components.

## 8 External Standards

As a final product consisting of electrical components meant to be used by primary school children, it is important to note the standards it must adhere to throughout the design process. NEMA: 250 - ‘Enclosures for Electrical Equipment’ [11] is relevant because the device is expected to be durable and safe for use by children with all electronic components stored within the 3D-printed casing. The IPC standards, specifically IPC-2221 [12], would apply to the design and manufacture of the custom-made keyboard's PCB. These standards ensure quality, reliability, and consistency in PCB manufacturing. Regarding the user interface, ISO-9241-210 [13] ensures ALEC device is designed with a user-centered approach, optimizing usability and comfort for young ESL learners when interacting with the keyboard, screen, and speakers. UL 60950-1 [14], which ensures the electrical safety of information technology equipment such as the MCU, is a critical consideration for the device intended for use by children in educational settings.

## 9 Intellectual Property Issues

ALEC is not patentable due to the use of pre-existing, copyrighted elements, such as SpongeBob keycaps and a SpongeBob floral design for the casing. These design choices rely on intellectual property owned by third parties and are not original enough to claim as novel or non-obvious under patent law. Of course, this feature was incorporated in order to increase engagement with our target demographic, but it may be adjusted with simple colorful keycaps and paint for mass production to reduce costs and increase patentability.

Additionally, since the ALEC project stems from Texas Instruments’ Speak-and-Spell device from the 1980s, the argument could be made that the general functionality of audio playback and user interaction for spelling exercises does not encourage novel ideas for design. However, ALEC incorporates a new method of teaching using translation with the addition of the native language audio compared to the Speak-and-Spell showing novelty.

US Patent 4,209,836 [16] regarding Texas Instruments’ Speak-and-Spell describes a device for teaching spelling and pronunciation, providing audio playback of a word and prompting the user to type the word, and includes functionality for detecting user input errors and providing corrective feedback. The general functionality of ALEC aligns with this patent's claims, particularly in its audio playback and typing prompt mechanisms. However, the patent does not cover bilingual features, such as presenting a word in the user's native language followed by English. ALEC’s unique bilingual sequence introduces a new teaching improvement that differentiates it from the scope of this patent.

US Patent 6,565,358 [17] regarding a Language Teaching System covers a system that facilitates language learning by providing recorded expressions in a target language, allowing students to hear and practice pronunciation. This patent specifies the inclusion of translations in the student's native language to aid comprehension. This patent outlines a language teaching system that uses native and target language recordings, similar to ALEC. However, it does not include user interaction via typing prompts, which is a core feature of ALEC. Additionally, ALEC’s integration of sequential bilingual playback (native language first, then English) combined with typing exercises offers a tailored educational approach not directly covered in the dependent claims of this patent, leading to an increased patentability.

US Patent 8,032,384 [18] describes a portable device that provides translations and pronunciations of words between multiple languages, facilitating language learning. It includes features for interactive user engagement, such as typing exercises. This patent includes translation and interactive features similar to those in ALEC. However, ALEC’s focus on ESL students and its unique workflow of presenting words in the native language, then in English, followed by user input, distinguishes it from the claims here.

Based on the claims of these related and overlapping patents, ALEC has an opportunity of patentability if the physical design and keycaps were changed to simple colors or unique designs. ALEC accumulates the ideas of these patents into one unique device that encourages a novel method of language learning at home.

# 10 Timeline

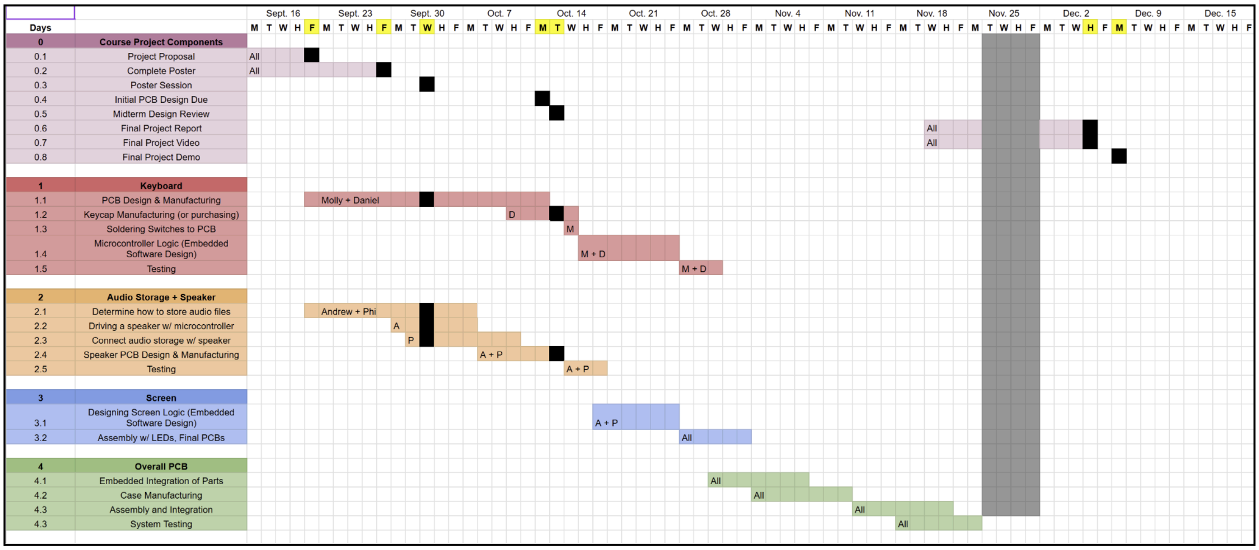



*Figure 14. Old Complete Gantt Chart*

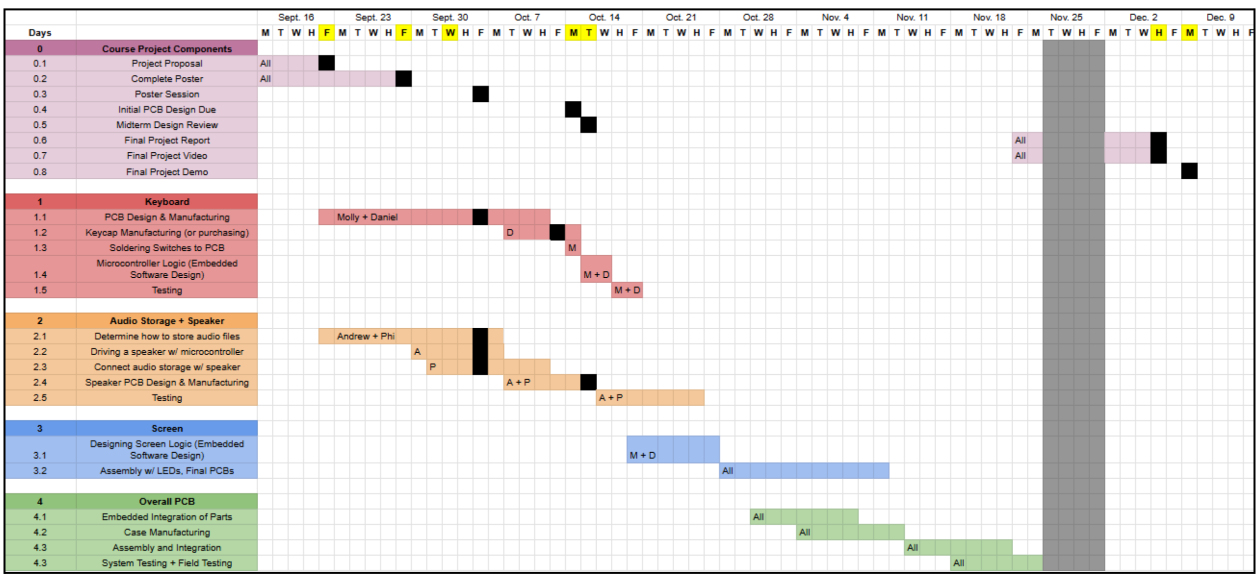



*Figure 15. Updated Complete Gantt Chart*

The Gantt chart displayed in Figure 15 displays the progression and development of the main course project components, the keyboard, the audio storage and speaker, the screen, and the overall system implementation with off-days for university Fall Break and national holidays designated with the black/gray blocks. The main differences between the two Gantt charts are:

keyboard development took less time than expected, audio system and screen development both took longer, and overall implementation testing took longer.

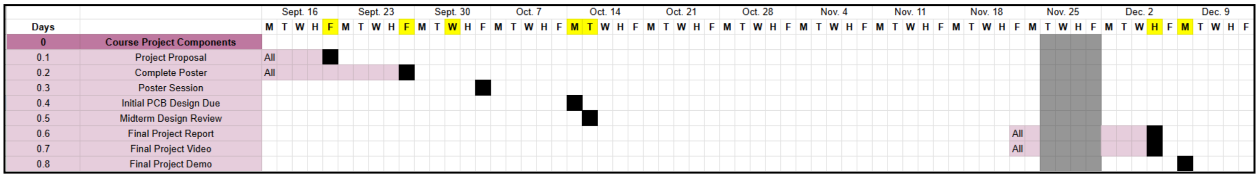



*Figure 16. Course Project Components Timeline*

For our course project components, our group stayed on time with the original Gantt chart, beginning our final report and final video demo a little later than expected.

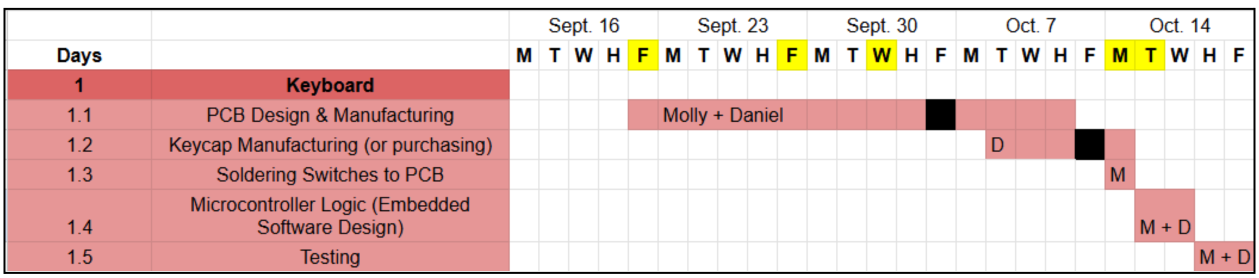



*Figure 17. Keyboard Design Timeline*

Molly and Daniel's keyboard development for embedded logic and testing took less time than expected because they were able to split up the PCB design and soldering as well as the MCU logic, leading to less implementation time.

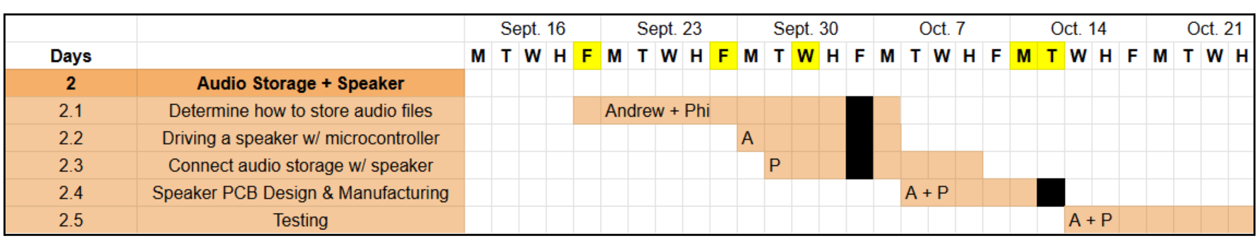



*Figure 18. Audio Storage and Speaker Design Timeline*

Phi and Andrew's audio system development took longer than expected due to a lengthened debugging process, especially with specific edge cases prior to the keyboard integration.

| | | Oct. 14 | | | | | Oct. 21 | | | | | Oct. 28 | | | | | Nov. 4 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Days** | | **M** | **T** | **W** | **H** | **F** | **M** | **T** | **W** | **H** | **F** | **M** | **T** | **W** | **H** | **F** | **M** | **T** | **W** | **H** | **F** | **M** |
| **3** | **Screen** | | | | | | | | | | | | | | | | | | | | | |
| 3.1 | Designing Screen Logic (Embedded Software Design) | | | | | M + D | | | | | | | | | | | | | | | | |
| 3.2 | Assembly w/ LEDs, Final PCBs | | | | | | | | | | | All | | | | | | | | | | |

*Figure 19. Screen Design Timeline*

The screen took longer to design and implement into our final design because we switched screens. The shipping time for the new MIDAS screen set us back a couple of days, but it was roughly the same logic allowing us to implement it once in hand quickly. This pushed back the overall design debugging back a couple of days.

| | | Oct. 21 | | | Oct. 28 | | | | | Nov. 4 | | | | | Nov. 11 | | | | | Nov. 18 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Days** | | **W** | **H** | **F** | **M** | **T** | **W** | **H** | **F** | **M** | **T** | **W** | **H** | **F** | **M** | **T** | **W** | **H** | **F** | **M** | **T** | **W** | **H** | **F** | **M** |
| | | | | | | | | | | | | | | | | | | | | | | | | | |
| **4** | **Overall PCB** | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.1 | Embedded Integration of Parts | | | | | | All | | | | | | | | | | | | | | | | | | |
| 4.2 | Case Manufacturing | | | | | | | | | All | | | | | | | | | | | | | | | |
| 4.3 | Assembly and Integration | | | | | | | | | | | | | | | | All | | | | | | | | |
| 4.3 | System Testing + Field Testing | | | | | | | | | | | | | | | | | | | All | | | | | |

*Figure 20. System Implementation and Testing Timeline*

For the final deliverable, we extended system testing and added field testing to tackle debugging errors and observe and gather live results before the demo day. Overall, our group stayed very close to the timeline, where we finished our design and tested it the week of November 25th, allowing us to allocate more time towards other project components.

## 11 Costs

The cost analysis of the ALEC project considers the expenses for the prototype, as well as the projected cost for scaling production to 10,000 units. The total cost for a single prototype was $228.58, while the estimated cost for manufacturing each unit at scale is reduced to $212.40 without manufacturing overhead considerations.

In bulk production, it is important to estimate the external costs as well. Each of the most notable external costs given common manufacturing costs and time assumptions per unit are estimated below in Table 1:

| Cost Component | Cost per Unit | Total for 10,000 Units |
|---|---|---|
| Labor (assembly & soldering) | $10.00 | $100,000 |
| Facility rental and utilities | $1.50 | $15,000 |
| Equipment amortization | $5.00 | $50,000 |
| QA testing | $1.67 | $16,700 |
| Packaging and shipping | $2.00 | $20,000 |
| **Total Production Cost** | **$20.17** | **$201,700** |

*Table 1. External Production Costs for 10,000 Units*

Adding the $20.17 in external costs to the $212.40 bulk component cost results in a **final per-unit cost of $232.57** when scaled to 10,000 units.

Therefore, the total cost for 10,000 units, including components and external production costs is estimated at:

$$232.57 \text{ USD/unit} \times 10{,}000 \text{units} = \mathbf{2{,}325{,}700 \text{ USD}}$$

This comprehensive cost includes all aspects of production and assembly, making the device economically viable for its intended audience while allowing for potential pricing adjustments based on market and funding considerations.

The prototype cost and the bulk production cost differ slightly due to the use of in-house materials and labor during prototyping, as well as the additional shipping costs associated with acquiring individual components for the prototype. Considering the shipping was an additional ~$40 for all materials for our purposes, the bulk cost of $232.57 is still cheaper than the $228.58 + ~$40 prototype cost.

Appendix A shows the total bill of materials and compares the estimated material costs of prototyping and bulk manufacturing.

## 12 Final Results

Our final prototype met our design goals very well. The power button is on the back of the 3D-printed case. When it is turned on, it displays the welcome screen with three different language options. Once an option is selected, the screen shows the instructions of the tool in the chosen language, and then the first word in the language and its translation in English are spoken through the speaker. The volume is on the side of the case and can be adjusted at any time. If the user types in the correct spelling of the English word and presses enter, a green LED shines, a nice chime is played, and the next word is presented. If the user gets it wrong, a yellow light is shone and the user has the chance to try again. At any point, the user can press the help button, which will repeat the word and its translation. If the user presses the help button 3 times, a hinting letter is given to the user. If the user continues to press the help button, each letter will be given until the word is completely spelled. Once this happens, a yellow light shines and the correct word is again presented. If the user presses the enter button 5 times and the word is still spelled incorrectly, the same thing happens, where the correct word is presented to the user along with a yellow light and an encouraging message. The user can go through all 50 words in their flash drive forever, as they reshuffle once they have all gone through. If the USB is removed at any point during the exercise, the device cuts back to the main screen and the keyboard is disabled.

This proves that our final results for our project were very successful. The USB can be replaced with another USB with harder words or full phrases for furthering language practice with no problem. This proves that our device is capable of expanding the listening comprehension learning process, our ultimate goal. In our proposal, we defined the following criteria:

- An audio system that sources audio files from a USB drive and plays them through an external speaker
- A working keyboard that shows inputs on an LCD display
- LED feedback for each input from the keyboard
- A battery power source for the device

- A fully functioning listening comprehension and spelling practice game for at least one language (Spanish)

Our project met all of these criteria. We expanded on the last criteria and created a fully functioning listening comprehension and spelling practice game for three languages.

## 13 Engineering Insights

Working on ALEC has provided invaluable experiences and lessons in technical skill development, engineering processes, and collaboration.

### 13.1 Technical Skills

We gained proficiency in embedded C programming, specifically using the STM32CubeIDE to manage peripherals like the screen, keyboard, and DAC. Honing skills from ECE 3430: Intro to Embedded Computing Systems while also learning to manage concurrency issues and implementing atomic operations were key challenges that enhanced our understanding of low-level programming and timing.

Regarding our keyboard, we gained experience in circuit design and analysis through the development of our keyboard and LEDs. In the short time frame, we had to ensure our PCB designs were completely correct because shipping took 1-2 weeks, while also ensuring we had thought about different components needed for all uses in the future. For example, the implementation of diodes to prevent ghosting was an important aspect to keep in mind.

Developing the audio storage and playback systems taught us about implementing file systems and using FATFS, using I2S and I2C communication protocols, and understanding DACs. Handling edge cases such as removing the USB flash drive mid-playback helped us design robust, fail-safe systems. We also conducted unit and system-level testing, strengthening our problem-solving and debugging skills.

### 13.2 Engineering Processes

The most important aspects learned throughout the project were time management and communication within our team. The project is only a semester but contains many steps such as brainstorming an idea, finding necessary parts, ordering and shipping for these parts, assembling your design, etc. Creating a Gantt chart was one of the most helpful skills because our group had a timeline to follow and gave us the opportunity to allocate sufficient time wherever needed.

Within our group, dividing up into teams (keyboard and audio) based on individual strengths and maintaining regular team meetings ensured smooth collaboration where each person contributed. Weekly meetings with our advisor and reports encouraged group members to have clear communication and documentation for their contributions.

While working on this project, we encountered many bugs and errors that took days to fix such as timing issues with missed key presses. It was important for team morale to take needed breaks and celebrate the small milestones. Taking a step back and slowly walking through the code helped us tackle our major problems.

**13.3 Advice for Future Students**

In the future, an important consideration would be to establish the roles and goals for each individual team member. Establishing roles and goals for each team member is essential for effective collaboration and project success. By identifying each member's strengths and assigning roles accordingly, teams can leverage individual expertise while ensuring accountability. Clear goals tied to these roles help maintain focus and track progress. Regular check-ins enable the team to monitor progress, adapt as needed, and address unforeseen challenges. While roles provide structure, fostering collaboration and flexibility encourages cross-functional support, ensuring seamless integration and problem-solving. Defining shared goals, such as delivering a functional prototype by a set deadline, aligns the team toward a common purpose, enhancing efficiency, accountability, and overall project quality.

Additionally, allotting time for debugging is a critical aspect for the development process, ensuring that software or hardware functions as intended and meets user needs. It allows the team to identify and resolve errors, improving system reliability, performance, and security. Effective debugging not only prevents small issues from escalating into major problems, but also saves time and resources in the long run. It fosters a deeper understanding of the code or system, enabling the team to write more robust and maintainable solutions. Moreover, debugging enhances user satisfaction by delivering a polished product free of frustrating glitches or malfunctions, ultimately contributing to the project's success.

Lastly, team bonding is essential for fostering trust, collaboration, and a sense of unity within the team. It helps individual team members understand each other's strengths, weaknesses, and communication styles, creating a foundation for smoother collaboration. Effective team bonding can boost morale, increase motivation, and enhance problem-solving by promoting open communication and mutual respect. It also helps prevent misunderstandings and conflicts, making the team more resilient in the face of challenges. Whether through shared activities, regular check-ins, or celebrating successes together, team bonding cultivates a positive environment where everyone feels valued, ultimately improving productivity and achieving collective goals.

## 14 Future Work

In the future, this project can be expanded upon in a few different ways. The project currently works for three different pre-set languages: Spanish, French, and German. The reason these three languages were chosen is because English characters can be used for the encouraging messages in between words, when the user spells the word correctly. If a group were to take over this project in the future, a valid extension to this would be to use a different screen that would allow for different characters to be portrayed, such as Arabic letters. This would extend the reach of the device into more communities that don't have equal access to the English language for listening comprehension practice. An add-on that would benefit this would be a larger display that is a bit clearer. This could be in the form of an LED graphic display instead of an LCD alphanumeric display. One difficulty of doing this, as we saw during our project, is satisfying the power requirements. Luckily, an LCD display can run off of the 5V coming from the MCU board and is very easy to display characters using GPIO pins to transmit the data. An LED display would need +5V and -5V power pins, which we could not run straight off of the MCU board. This would also need a different communication protocol to show the correct words.

Another extension to the project could be to expand the language reach to other languages. As previously mentioned, the tool currently can help with Spanish, French, and German native speakers. A new graphic display would open the tool up to many languages that don't have English letters. With this, another layer of difficulty is added. The start-up screen that is currently portraying the three languages would have to read from the USB which languages are installed. We tackled part of this problem by reading the text files that have which words are being played, but this would mean adding another text file that includes the languages on the USB drive. This would prove difficult using C on the STM32, as this is a file that has to be ignored during shuffling so that the system doesn't try to find a file that doesn't include audio. Expanding upon this project would be an exciting and worthwhile task for any future students.

## 15 References

[1] A. M. Burford, "Keyboarding: An Important Skill for the Office of the Future," The Journal of Business Education, vol. 55, no. 7, pp. 290–293, Apr. 1980, doi: https://doi.org/10.1080/00219444.1980.10772629.

[2] A. Poulos and M. J. Mahony, "Effectiveness of feedback: the students' perspective," Assessment & Evaluation in Higher Education, vol. 33, no. 2, pp. 143–154, Apr. 2008, doi: https://doi.org/10.1080/02602930601127869.

[3] "Childstats.gov - America's Children: Key National Indicators of Well-Being, 2021 - Language Spoken at Home and Difficulty Speaking English," *www.childstats.gov*. https://www.childstats.gov/americaschildren/family5.asp#:~:text=SOURCE%3A%20U.S.%20Census%20Bureau%2C%20American

[4] E. Dyvik, "Most Spoken Languages in the World," *Statista*, Jul. 04, 2024. https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/

[5] "English Language Learners Marginalized in Classrooms – Undercurrents." https://blogs.umb.edu/undercurrents/2019/08/20/english-language-learners-marginalized-in-classrooms/

[6] G. Frantz and R. Wiggins, "The Development of 'Solid State Speech' Technology at Texas Instruments," *Ieee.org*, Mar. 1981. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1165777 (accessed Sep. 19, 2024).

[7] I. Dagilienė, "Translation as a Learning Method in English Language Teaching," Studies About Languages, vol. 0, no. 21, Dec. 2012, doi: https://doi.org/10.5755/j01.sal.0.21.1469.

[8] STMicroelectronics, "Discovery kit with STM32F407VG MCU," User Manual DM00039084, Nov. 2011. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00039084-discovery-kit-with-stm32f407vg-mcu-stmicroelectronics.pdf. [Accessed: 19-Sep-2024].

[9] UNICEF, "Two Thirds of the World's School-Age Children Have No Internet Access at Home, New UNICEF-ITU Report Says," *UNICEF*, Nov. 30, 2020. https://www.unicef.org/press-releases/two-thirds-worlds-school-age-children-have-no-internet-access-home-new-unicef-itu

[10] WorldAtlas, "The Most Popular Official Languages Of The World," *WorldAtlas*, Jun. 14, 2017. https://www.worldatlas.com/articles/the-most-popular-official-languages-of-the-world.html
[11] "NEMA 250-2018 Enclosures for Electrical Equipment (1000 Volts Maximum)." Available: https://www.nema.org/docs/default-source/standards-document-library/nema-250-2018-contents-and-scope.pdf?sfvrsn=96e44a99_1
[12] "IPC-2221 Generic Standard on Printed Board Design The Institute for Interconnecting and Packaging Electronic Circuits," THE INSTITUTE FOR INTERCONNECTING AND PACKAGING ELECTRONIC CIRCUITS, Feb. 1998. Available: https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC_2221.pdf
[13] "ISO 9241-210 Ergonomics of human-system interaction - Human-centred design for interactive systems," International Organization for Standardization , Mar. 2010. Available: https://richardcornish.s3.amazonaws.com/static/pdfs/iso-9241-210.pdf
[14] C. Bayhi, "UL/CSA 60950-1 UL/CSA 60950-1 2nd Edition 2nd Edition," Jun. 24, 2008. https://ewh.ieee.org/r6/ocs/pses/60950-1,2ndEd-1.pdf
[15] GPE Secretariat, "5 ways education can help end extreme poverty," Globalpartnership.org, Oct. 17, 2016. https://www.globalpartnership.org/blog/5-ways-education-can-help-end-extreme-poverty
[16] R. Higgins and G. Brantingham, "SPEECH SYNTHESIS INTEGRATED CIRCUIT DEVICE," Jun. 24, 1980 Accessed: Dec. 04, 2024. [Online]. Available: https://ppubs.uspto.gov/dirsearch-public/print/downloadBasicPdf/4209836?requestToken=eyJzdWIiOiI4MWU0YmI3MC1hNjVmLTQ0ZmQtOWQ0My1hMmU4OWExYjhiNzciLCJ2ZXIiOiI4YzIyZmY1ZC1lNzcyLTRlYWEtODIxYS00NjhkMjYyODQzMzQiLCJleHAiOjB9
[17] M. Thomas, "LANGUAGE TEACHING SYSTEM," May 20, 2003 Accessed: Dec. 04, 2024. [Online]. Available: https://ppubs.uspto.gov/dirsearch-public/print/downloadBasicPdf/6565358?requestToken=eyJzdWIiOiI4MWU0YmI3MC1hNjVmLTQ0ZmQtOWQ0My1hMmU4OWExYjhiNzciLCJ2ZXIiOiI4YzIyZmY1ZC1lNzcyLTRlYWEtODIxYS00NjhkMjYyODQzMzQiLCJleHAiOjB9
[18] J. Rylander, "HAND HELD LANGUAGE TRANSLATION AND LEARNING DEVICE," Oct. 04, 2011 Accessed: Dec. 04, 2024. [Online]. Available:

https://ppubs.uspto.gov/dirsearch-public/print/downloadBasicPdf/8032384?requestToken=eyJzdWIiOiI4MWU0YmI3MC1hNjVmLTQ0ZmQtOWQ0My1hMmU4OWExYjhiNzciLCJ2ZXIiOiI4YzIyZmY1ZC1lNzcyLTRlYWEtODIxYS00NjhkMjYyODQzMzQiLCJleHAiOjB9

# 16 Appendix

Appendix A: Detailed Material Cost Analysis for Prototype and Bulk Production

| Component | Supplier | Unit Price (Prototype) | Quantity | Prototype Cost | Bulk Unit Price (10,000) | Bulk Cost |
|---|---|---|---|---|---|---|
| STM32F407G 1-DISC1 microcontroller | Digikey | $20.91 | 1 | $20.91 | $209,100.00 | $209,100.00 |
| Custom Keyboard PCB | JLC PCB | $2.46 | 1 | $2.46 | $24,600.00 | $24,600.00 |
| XMSJSIY Mini Computer Speaker | Amazon | $14.99 | 1 | $14.99 | $149,900.00 | $149,900.00 |
| Midas Displays LCD screen | Digikey | $19.30 | 1 | $19.30 | $74,800.00 | $74,800.00 |
| Voltaic Systems V25 always-on battery power bank | Amazon | $44.00 | 1 | $44.00 | $440,000.00 | $440,000.00 |
| LED PCB | JLC PCB | $0.40 | 1 | $0.40 | $4,000.00 | $4,000.00 |
| GATERON PC Screw in Crystal Stabilizers | Amazon | $11.99 | 1 | $11.99 | $119,900.00 | $119,900.00 |
| Kailh Mechanical Key Switches | Digikey | $6.95 | 4 | $27.80 | $69,500.00 | $278,000.00 |
| Akko SpongeBob Keycap Set | Akko | $64.99 | 1 | $64.99 | $649,900.00 | $649,900.00 |
| SanDisk 32GB Cruzer Glide USB 2.0 Flash | Amazon | $5.77 | 1 | $5.77 | $57,700.00 | $57,700.00 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Drive | | | | | | |
| 1N4148W-TP General Purpose Diode | Digikey | $0.07 | 32 | $2.36 | $357.80 | $11,449.60 |
| LM1117DT-3.3/NOPB Linear Voltage Regulator | Digikey | $1.69 | 1 | $1.69 | $8,448.28 | $8,448.28 |
| TPS22917DBVR Power Switch | Digikey | $0.41 | 1 | $0.41 | $2,020.47 | $2,020.47 |
| CD74HC4067M96 16:1 Mux | Digikey | $0.58 | 1 | $0.58 | $2,472.30 | $2,472.30 |
| CL21A106KOQNNNE 10uF Ceramic Capacitors | Digikey | $0.08 | 3 | $0.24 | $88.00 | $264.00 |
| CC0805KRX7R9BB104 0.1uF Ceramic Capacitors | Digikey | $0.08 | 5 | $0.40 | $152.00 | $760.00 |
| RNCP0805FTD10K0 10k Resistors | Digikey | $0.10 | 12 | $1.20 | $92.70 | $1,112.40 |
| RC0805FR-071KL 1k Resistors | Digikey | $0.10 | 1 | $0.10 | $30.60 | $30.60 |
| USB to Mini USB Cable | Amazon | $8.99 | 1 | $8.99 | $89,900.00 | $89,900.00 |
| **Total Cost of Prototype** | **$228.58** | | | | | |
| **Total Cost of One Bulk** | **$212.44** | | | | | |