Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

User's Manual

Portable Language Translator

Team 23
Portable Language Translator

Team Members

Andrew Nguyen aynguyen@bu.edu
Cristian Palencia cris0211@bu.edu
Ryan Liao ryanliao@bu.edu
Yohan Kim yohank@bu.edu

Submitted: 4/18/2025

Portable Language Translator
Table of Contents

Executive Summary

Communication is one of humanity's most fundamental needs, yet language barriers and lack of widespread knowledge of ASL hinder our ability to connect. Our project aims to provide an all-in-one translation device that enables real-time translation between spoken languages and seamless conversation between ASL and speech. Our device will use speech recognition and translation, a camera, and a screen, all in a portable form factor to bridge communication gaps. It will enable users to engage in natural conversations, translating spoken words bidirectionally and interpreting ASL gestures into speech, fostering inclusivity and accessibility for diverse communities.

1. Introduction

In today's increasingly interconnected world, language barriers continue to hinder effective communication—especially in situations where immediate understanding is crucial. This challenge is even more pronounced when it comes to American Sign Language (ASL), where accessible and seamless translation tools remain limited. While many devices exist to translate spoken language, few offer reliable solutions for interpreting ASL gestures, leaving a significant gap in support for the Deaf community. This project aims to bridge that gap with the Portable Language Translator, a device designed to recognize and translate both spoken language and ASL in real time.

One of the teams' members, Andrew, previously worked at a popular matcha cafe. With some non-English speakers also visiting the cafe, it was difficult to give them the service they deserved in the time frame that is expected when the customer and Andrew did not understand each other. With the Portable Language Translator, this communication would have been simplified and prevented any misunderstandings. This device is meant to help anyone in the service industry or anyone surrounded by people that don't share their language.

Our approach integrates speech and image-based gesture recognition using a combination of hardware and software such as Google Cloud and machine learning. The portable translator uses a Raspberry Pi 5 along with a camera and microphone that captures user input. Spoken languages are transcribed using Google Cloud's Speech-to-Text API and translated with Google Cloud's Translation API. For ASL, we trained a Long Short-Term Memory (LSTM) model using TensorFlow to classify 5 different gestures (Hello, Thank You, Bathroom, Help and Yes), using OpenCV for video capture and MediaPipe to collect landmarks from the user's hand and body. The TensorFlow model was converted into a TFLite model in order to host the model locally on the Raspberry and still have smooth performance. This approach solves a user's key issue and enables real-time communication between speakers of different languages and even signers and non signers.

From a user's perspective, several features of our project stand out. Our device allows for real-time ASL recognition as the camera captures hand gestures and classifies them with an 80%+ accuracy. Another feature of our device is the spoken language translation as the device will detect the spoken language between English, Spanish and Korean and translate into the user's preferred language in under 2 seconds using cloud-based APIs. Our device enables hand-free communication as it is housed in a 3D enclosure that is worn with a pair of shoulder straps. Fluid communication is enabled as only one user needs to wear the portable device and text output is converted back to audio with text-to-speech conversion which removes the need to pass a device back and forth.

Internet connectivity is required in order to use the spoken language translation as translation is done using Google Cloud's APIs. While the system runs efficiently, the combined use of camera, microphone and wifi results in a battery life of about 3.125 hours.

The remaining sections in this manual will walk you through hardware components, set up instructions, software architecture and troubleshooting steps. Detailed diagrams and usage examples will help users understand how to use the Portable Language Translator.

2. System Overview and Installation

2.1. Overview block diagram

The PLT (Portable Language Translator) device is primarily held and contained within a physical encasing. The encasing provides mountings for the camera, microphone, and touch screen which all receive relevant input in the form as user interaction, ASL gestures, and language to be translated. All of these inputs are taken in by relevant hardware that is processed by a Raspberry Pi which handles translations that will be outputted in the form of visual text on the User Interface or audibly through the device's speakers.

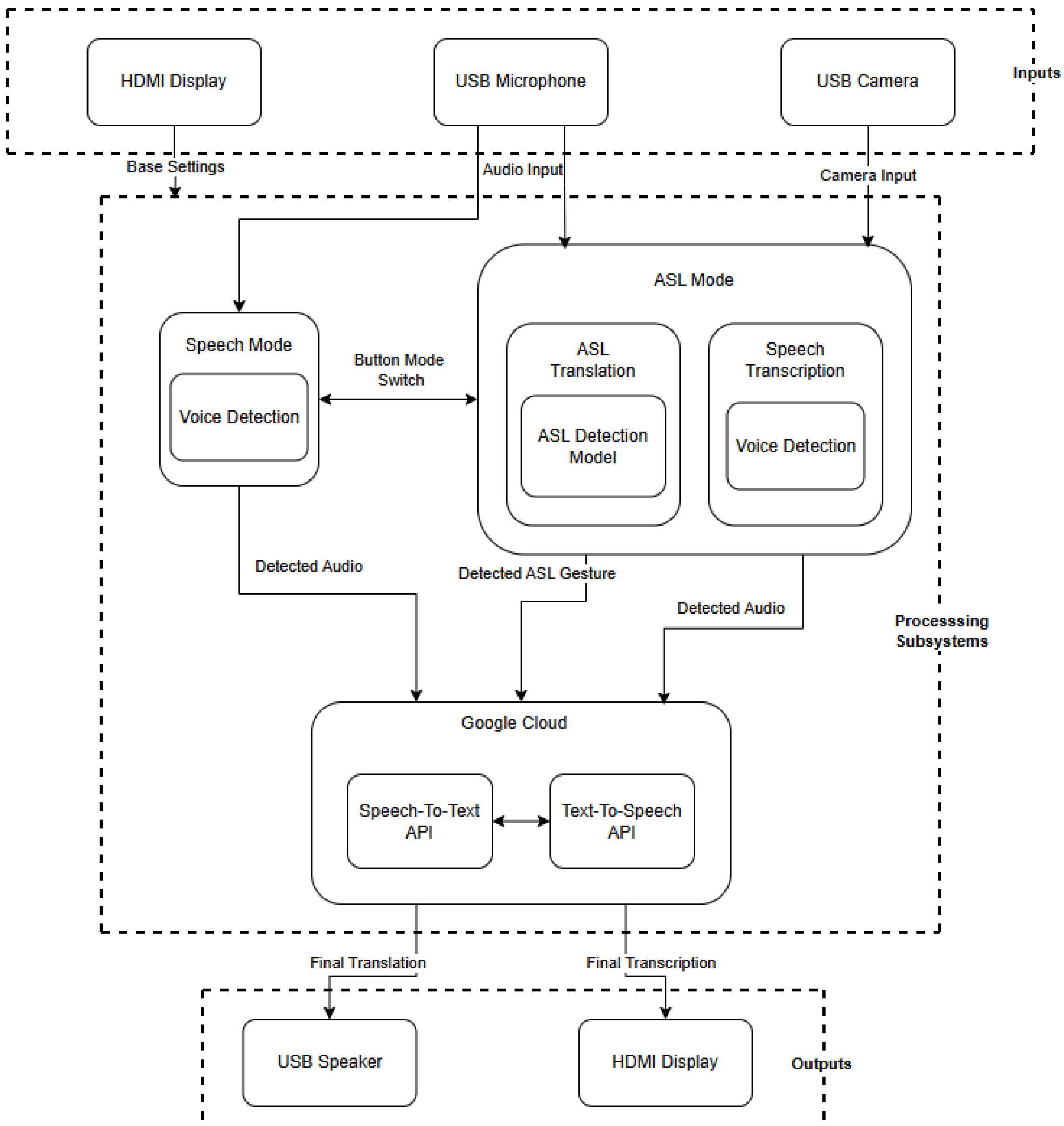
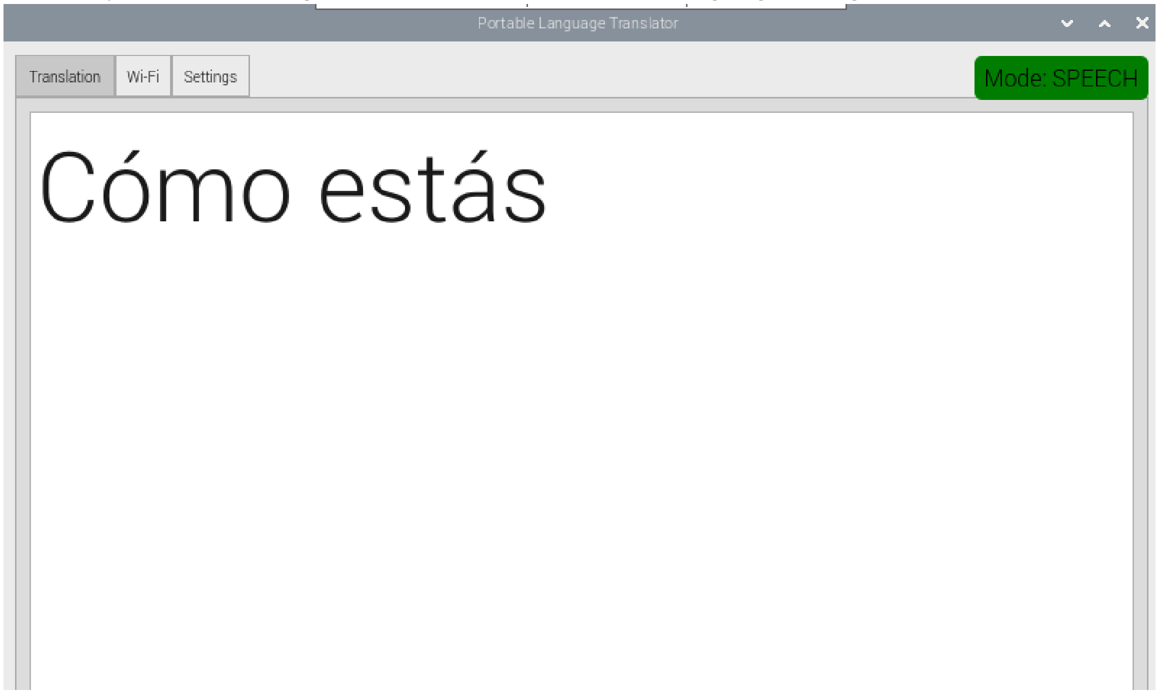
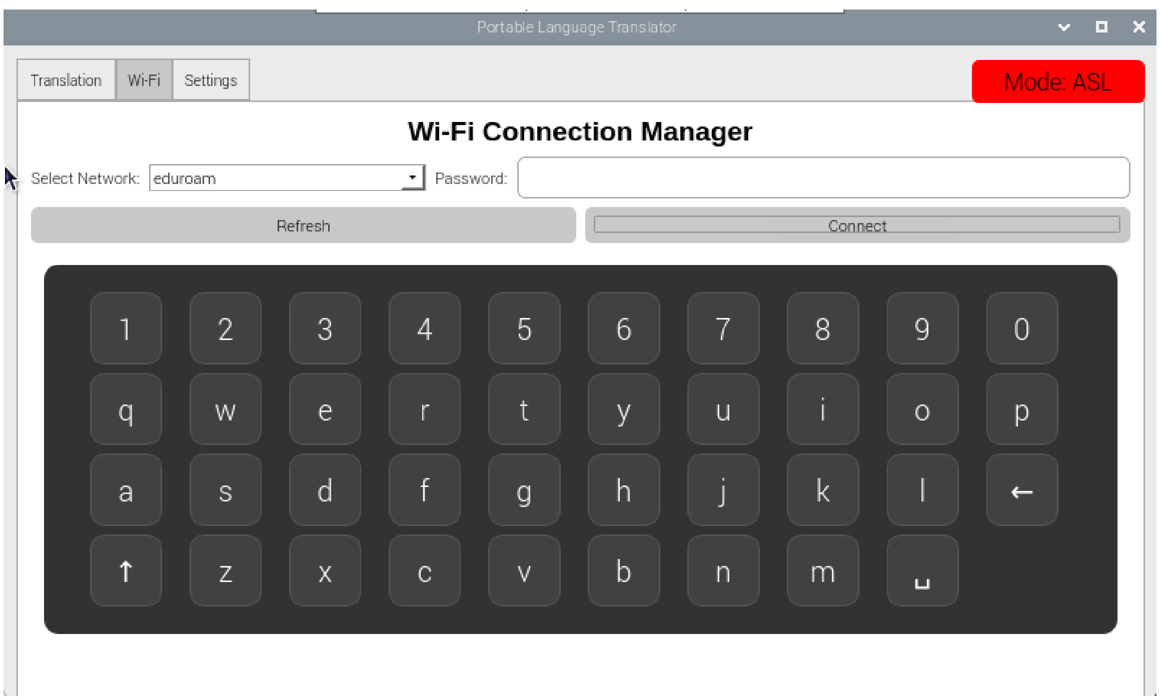


Figure 2.1 System Block Diagram

2.2. User Interface.

The User Interface consists of three major tabs. The first tab includes all relevant functionality for ASL and Language Translations. The second tab is in charge of allowing the user to connect to any available WiFi connections. This includes regular router connections or user provided hotspots. Finally, the third tab is used for allowing the user to set any desired settings for the device such as the language being translated.





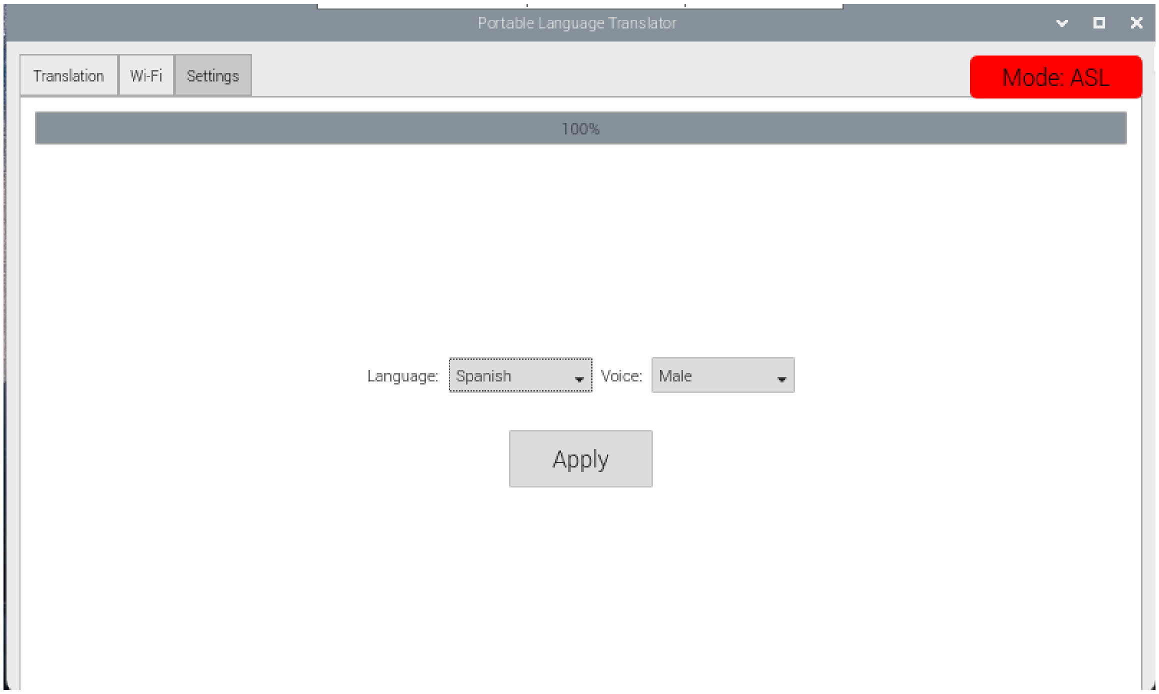


Figure 2.2 User Interface Tabular Organization

2.3. *Physical description.*

The device is composed of 3D printed components that mount the various components of the device. The device includes three components. One, the main frame directly allows the mounting of the Raspberry Pi, touch screen, and camera. Two, the back plate that permits ventilation and the mounting of the battery pack. Three, the top plate that encloses the system speakers. Finally, the user should be aware of the included straps that allow the device to be worn using shoulder support.

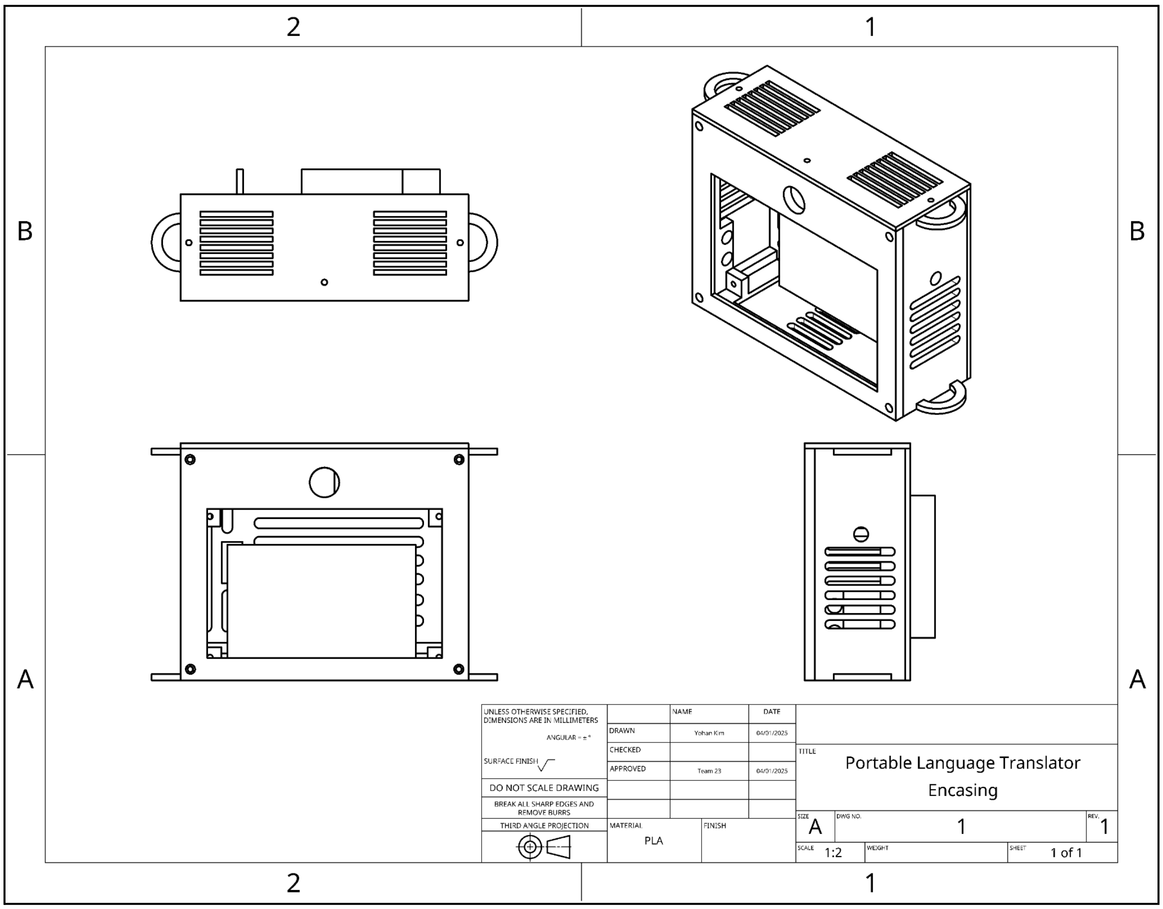


Figure 2.3 3D Drawings

Figure 2.4 Physical Design

2.4. Installation, setup, and support

Installation and Setup

Every PLT device already comes fully assembled along with all relevant software already included. Some relevant considerations when first obtaining the PLT include the following:

1) *Battery pack charge*: The included battery pack may not come charged already. If this is the case please use a micro-usb cable to connect to the battery pack and let it charge so that the device can properly operate.

2) *Wearing the device*: To put on the straps, start by clipping both straps onto the two bottom rings of the device. Then, take the bottom left strap and clip it onto the opposite top right ring. Next, put your left arm and head through the loop you've created. Finally, take the bottom right strap, bring it across your back and over your left shoulder, and clip it onto the top left ring. Adjust strap lengths as necessary.

3) *Internet connection*: The user needs to ensure that the device is connected to the internet. To do this please navigate to the second tab on the user interface and check for available networks that you may connect to.

3. Operation of the Project

3.1.Operating Mode 1: Normal Operation

In order to use the PLT:

1. Turn on the device by pressing on the power button on the battery pack once.
2. Once the device has turned on and the GUI has loaded, the language translation mode will be on by default.
3. Connect to the wifi network using the wifi tab by selecting a network and inserting a password if necessary using the touch screen.
4. Change desired settings in the settings tab using the touch screen again.
5. Once connected to wifi and updated the settings, the user is ready to use the device and may switch back to the Translation tab.
6. While in speech to speech mode, the user may speak and the device will start translating back and forth.
7. Press the button on the left side of the device to switch into ASL mode. The user can check which mode they are in with the indicator on the top right of the GUI.
8. While in ASL mode, the GUI will show the camera feed and start detecting ASL. Once the device finishes detecting, it will play the detected ASL gesture as speech. After the device finishes playing the audio, the GUI will switch to a text box and start listening for speech. After it has finished detecting speech, it will display the transcribed speech onto the text box. After 3 seconds, the GUI will switch back to the camera feed.
9. While using the device in either mode, the volume of the voice out can be adjusted with the two buttons on the right side of the device. The top button will raise the volume while the bottom button will decrease it.

3.2. Operating Mode 2: Abnormal Operations

One abnormal operation would be if the battery pack powering the device runs out of power and the device shuts off. If this happens, charge the battery pack and then reconnect the battery to the device when sufficient power is available. Next, turn the device back on by pressing on the power button once on the battery pack and the GUI will automatically load and the device will be ready to run once again. The device cannot operate while the battery is charging.

Another abnormal operation is the case of disconnection from the internet. If this happens, click on the wifi tab and reconnect to a network by selecting a wifi network and inserting a password if necessary using the onscreen keyboard. The device will be ready to translate once it has been reconnected.

3.3. Safety Issues

The device contains several openings for ventilation and access to hardware components such as the battery pack and cooling system. These openings are not sealed against moisture or water ingress. Exposure to rain, high humidity, or accidental splashes can lead to short circuits, damage to internal electronics, or even pose a fire or electric shock hazard. For safe operation, always use the device in dry, indoor environments and store it in a protective case when not in use.

4. Technical Background

4.1. Software Component

There are three major sections in the software component of the project: the language translation, ASL gesture recognition model, and the user interface software.

Language Translation:

The Language Translation subsystem processes spoken language inputs and converts them into another spoken language. The process begins by capturing audio using the systems usb microphone, which is processed in chunks to detect speech using the WebRTC VAD (Voice Activity Detection) library. This ensures that only relevant speech segments are processed. Once speech is detected, it is converted to text using Google Cloud's Speech-to-Text API. The text is then translated into the desired language using Google Cloud's Translation API. Finally, the translated text is synthesized into speech via Google Cloud's Text-to-Speech API, which is played back to the user through a USB speaker. This subsystem enables seamless bidirectional speech-to-speech communication between users speaking different languages.

ASL Gesture Recognition:

For the ASL gesture detection and prediction, the external camera (USB 2.0 Camera) will record a live video stream and the script will initialize this video feed with OpenCV. First processing camera frames is done with Mediapipe's hand landmarks model to extract hand landmarks and then grouping these landmarks from the last 30 frames into a sequence. This sequence is fed into our LSTM (Long Short Term Memory) model that outputs the probability for each gesture. When a specific gesture's confidence is greater than the threshold value, that gesture will be then chosen as the detected gesture. An asynchronous thread handles this inference process, ensuring that consistent and confident predictions are added to a running sentence, which is eventually converted into speech for translation. Once the sentence is done playing, the script will then listen for the audio and save the transcription in a .txt file where it will be displayed in the UI. Then the script will go back to detecting ASL gestures and the cycle will restart.

User Interface:

The user interface was programmed using a UI library called PyQt5. This allows us to create relevant UI objects that are linked to key components of the project. For example, it allowed the creation of labels, buttons, text boxes to support translations, wifi connectivity, and configuring relevant settings.

4.2. *Hardware Component*

The hardware for the project included several components including the Raspberry Pi 5, relevant USB devices (microphone, speaker, and camera), touch screen, and battery.

Raspberry Pi 5

The Pi 5 acts as the main processing unit of our project. It handles receiving relevant input from the USB devices and by interfacing directly with the touchscreen display. It includes relevant software for processing the inputs to be either translated or recognized as a gesture. Afterwards it handles properly outputting to either the system's speakers or back onto the touchscreen.

Microphone

USB device connected to the Pi. It allows the users to speak directly into the device for appropriate translation.

Speaker

USB device connected to the Pi. Outputs relevant translation processed by the Raspberry Pi

Camera

USB device connected to the Pi. When in gesture recognition mode it reads in the camera feed such that the Pi can process supported ASL gestures

Touch Screen

Connected directly to the Raspberry Pi such that the user can interact with user interface for wifi connectivity and settings via touch input.

Battery

The battery pack is mounted to the back of the device and provides the devices with at least 3.125 hours of battery life at max performance.

4.3. *Physical Component*

The 3D print allows the user to interact directly with different components of the device. Specifically. it allows the user to interact with the touch screen along with three relevant buttons on the side of the device. The three included buttons are the volume control (two buttons for volume up and volume down) and one button for mode toggling between ASL translation mode and Language translation mode.

5. Relevant Engineering Standards

The Portable Language Translator was developed with both industry standards and real world engineering practices in mind. This section outlines the most relevant hardware, software, communication and usability standards that were applied during development. Each contributed to the system's accuracy, safety and user accessibility. These standards were reinforced through testing, iteration and refinement. By intentionally aligning the Portable Language Translator with these standards, the team ensures that the product is secure, scalable, portable and accessible while upholding the best practices in electrical, software and systems engineering.

Hardware and Electrical Standards

- IEC 62368-1 - Audio/Visual and ICT Equipment Safety
  Applied to the Raspberry Pi 5, which powers the USB peripherals (camera, speaker, microphone and HDMI Display) safely within low-voltage design limits. Power draw was tested at 1.6A and a 5000mAh battery supports up to 3.125 hours of safe runtime which exceeds the 2 hour design requirement.
- IEEE 802.11 - Wifi Communication
  The device connects securely to the internet through a user's phone hotspot using a touchscreen-based virtual keyboard. This allows for secure transmission of audio/image data to cloud services.

Software Engineering and Machine Learning Standards

- PEP 8 - Python Code Style
  Used throughout the development of scripts managing gesture recognition (OpenCV, MediaPipe, TensorFlow) and speech processing (Google Cloud APIs), improving readability and collaborative debugging.
- IEEE 829 - Test Documentation
  Formal testing of both spoken language and ASL recognition was structured in tabulated accuracy/error reports.
- TensorFlow Training Guidelines
  ML model training followed recognized practices: using sufficient training epochs, a clear confidence threshold and asynchronous threading to separate inference from data collection.

Accessibility and Human Interface Standards

- Section 508 Compliance
  This project was built with inclusivity in mind. The Portable Language Translator converts gestures to spoken language and spoken phrases to text. This device is usable by individuals who are deaf, hard of hearing or non-verbal.
- W3C Web Accessibility Initiative (WAI)
  The UI tab on the built-in touch display allows users to configure key settings such as base language and voice gender. The interface was designed following accessibility aware practices, including clear labeling, intuitive layout and touch-friendly elements to ensure usability for individuals with disabilities. While not web based, the UI adheres to core WAI principles.

Operating System and File Standards

- POSIX Compliance
  The Raspberry Pi 5 runs a Linux operating system, which is POSIX compliant. This ensures robust scripting with predictable file permission and process control and compatibility with standard tools for logging, threading and scheduling.

6. Cost Breakdown

Project Costs for Production of Beta Version (Next Unit after Prototype)				
Item	Quantity	Description	Unit Cost	Extended Cost
1	1	Raspberry Pi 5	84.99	84.99
2	1	Raspberry Pi 5 Active Cooler	4.99	4.99
3	1	Touch Screen	32.99	32.99
4	1	Microphone	4.50	4.50
5	1	Speaker	13.99	13.99
6	1	Camera	14.99	14.99
7	1	Battery Pack	8.45	8.45
8	1	Shoulder Straps	17.98	17.98
			Beta Version-Total Cost	182.88

Table 6.1 Cost breakdown

7. Appendices

7.1. Appendix A - Specifications

Specification	Value, Range, Tolerance, Units
Case Dimensions	167.6 mm x 43.8 mm x 111.7 mm
Power	5000 mAh battery pack
Latency (Speech)	< 3 sec
Latency (ASL)	< 3 sec
Speech Detection Range	50 - 70 dB within 1 m
ASL Detection Range	1 m
Battery Life	3.125 hours

7.2. Appendix B – Team Information

Cristian Palencia

I am a Computer Engineering major from Framingham, Massachusetts. From a relatively young age I've always enjoyed working with computers and computer parts. For that reason I decided to go into Computer Engineering in college and pursue a career that allows me to dive deeper into those interests. Specifically, throughout my time in Boston University I have discovered a main interest in Embedded Systems. For this reason I have posed myself as such for our project. I enjoy working with hardware components and being able to write code to actualize them.

Contact Information

- email: cris0211@bu.edu
- phone: 508-630-6468

Andrew Nguyen

I am a Computer Engineering major with an emphasis on full-stack development, machine learning, and computer graphics. I am passionate about technology as a whole but I am really interested in games and VR.

Contact Information:

- email: aynguyen@bu.edu

Yohan Kim

I am a senior majoring in Computer Engineering. Throughout my college career I have been interested in software engineering with an emphasis on backend engineering. I am passionate about creating impactful solutions and continuously expanding and learning new skills in the area of software engineering.

Contact Information

- email: yohank@bu.edu
- phone: 408-809-6250

Ryan Liao

I am a senior Computer Engineering student pursuing a concentration in Machine Learning from Beacon, NY. I want to use my experiences from college to create solutions that drive positive societal impact.

Contact Information

- email: ryanliao@bu.edu
- phone: 845-542-8689