Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Semester Report

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: 2024-12-07

Table of Contents

Executive Summary

Authored: Andrew Nguyen
Portable Language Translator
23 – PLT

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.

1.0 Introduction

Authored: Ryan Liao

Language barriers can hinder effective communication, especially where there is a need for instant and reliable translation of spoken languages or American Sign Language (ASL). This can create challenges in accessibility, inclusivity and seamless interactions. There are many language devices that address spoken language, but very few offer solutions for translating ASL gestures which leaves a gap in supporting the Deaf and Hard of Hearing communities. Also, there is a need for a portable, efficient and easy to use solution that integrates both spoken and gesture-based translations.

One of the team's members, Andrew, worked at a popular matcha cafe this previous summer. The lines were very long and went out the front door. Andrew needed to work very efficiently to ensure each customer received their orders in a quick manner. There were times where Spanish speakers came in and it was difficult to communicate with them. With this difficulty communicating, there was potential for wrong orders which would take more time to fix. With the Portable Language Translator, this communication would be simplified and prevent any misunderstandings.

Our project aims to solve this problem by developing a portable language translator that bridges the communication gap by enabling seamless translation of both spoken languages and ASL into spoken output. This will promote accessibility as well as inclusivity in personal and professional interactions. Our device will combine a microcontroller, video input for ASL gestures, and Google Cloud services to process spoken and gesture based inputs. We are using Google's text-to-speech, translation and speech-text APIs and an ML model to recognize ASL letters and gestures. Preprocessed input data will be sent to the Google Cloud for translation and the results will be returned as an audio or visual output. In terms of hardware, the device will utilize a camera, microphone and an OLED display. The camera and microphone will be used to take in ASL as a visual input and spoken language as an audio input respectively. The OLED screen will be used in the ASL mode for the deaf/hard of hearing user as the spoken language will be displayed on the screen for them to read.

Our device will solve the user's problems and needs by providing a unified solution for both spoken and ASL translations which will address the needs of people who encounter language barriers in real time. Its portability ensures convenience along with accurate translations in real time.

Special Features of our project include:

- Dual Translation Modes: Support of both spoken language translation and ASL translation
- Portable Design: Compact and lightweight device that will be able to be held in one hand for easy on-the-go use
- Cloud Integration: Leverages Google Cloud for high accuracy translations as well as hosting our future ML model
- ML Model: Trained model that uses keypoints on your hand to identify ASL letters

2.0 Concept Development

Authored: Andrew Nguyen

2.1 Customer Problem

1. Real-Time Bidirectional Translation: Desires a device that must process and translate spoken input of specified languages and provide smooth translations between those languages.

2. Language Detection: The device should be able to automatically detect what language is spoken, as to minimize user input when using it.

3. Controlled Voice Detection: The device should not detect everything spoken around it and should stay constrained to the users' conversation and voices in the immediate vicinity.

4. ASL Translation: Desires a device that can process and translate ASL and process that into audio for the user to understand.

5. Speech-to-text: In order to communicate with someone with hearing impairments, transcription of speech and a display is necessary.

6. Portability: A translation device should be compact, lightweight, and able to be carried around in daily life without being intrusive.

7. User-Friendly Operation: A device should be easy to use with intuitive design choices as to reduce any learning curve and allow immediate use.

8. Multimodal: A device with both speech and ASL capabilities should allow the option to switch between these modes and also be able to change various settings to personalize the users' experience.

2.2 Conceptual Approach

Hardware:

The proposed solution is a compact, clip-on device with the following hardware modules:

- Microcontroller: Manages core processing tasks, acting as the device's brain for coordinating data input, processing, and output.

- Camera: Captures ASL gestures, allowing the device to interpret sign language in real-time. This inclusion addresses the gap in communication tools for the deaf and hard-of-hearing community.
- Speaker: Delivers clear, audible translations for spoken language users, enabling bidirectional conversation.
- Microphone: Detects speech input within a conversational range, enabling smooth, hands-free operation without requiring the user to hold or adjust the device.
- Screen: Displays text translations for users who prefer visual output or for those interacting with ASL users.
- Buttons: Physical buttons provide intuitive controls for power, mode switching, and other settings.
- LED Lights: Offer visual indicators for power and mode, keeping the user informed about the device's functionality at a glance.
- Bluetooth/Wi-Fi Connectivity: Enables the device to communicate with a web application and utilize cloud services for high-accuracy translation processing.
- Battery: A rechargeable battery provides portable power for 2–6 hours of continuous use, ensuring the device meets the demands of daily life.
- Encasing: The encasing is lightweight, durable, and designed with a wearable clip-on mechanism, ensuring comfort and convenience for users in various scenarios.

Software:

The software is modular, with three primary components designed to address the diverse functionality of the device:

1. Language Translation Module
    - Handles speech-to-speech and speech-to-text translations using GCP.
    - Utilizes automatic language detection and translation APIs to minimize user input.
    - Ensures hands-free operation, allowing users to communicate naturally without interruptions.

2. ASL Translation Module
    - Processes video input from the camera to detect and interpret ASL gestures using machine learning models.
    - Supports the creation of words and sentences from ASL letters, enabling meaningful and fluid communication.
    - Bridges the gap for the deaf and hard-of-hearing community, offering a communication tool that is often overlooked in the market.

3. Web Application Module
    - Provides a user-friendly interface for customizing device settings, such as selecting languages, voices, and modes.
    - Enables easy configuration and updates, ensuring the device adapts to the evolving needs of its users.
    - Accessible from any internet-connected device, enhancing flexibility and user control.

This approach prioritizes inclusivity, adaptability, and ease of use, ensuring the device meets the diverse needs of users across different languages and communication modes. By integrating ASL alongside spoken languages, it addresses a critical gap in communication tools and fosters connections between signing and non-signing users. Automatic language detection and hands-free operation make the device intuitive and accessible, eliminating the need for constant interaction. The modular design of the software and hardware ensures scalability and adaptability for future updates, while its lightweight, wearable design guarantees portability and comfort for everyday use. By streamlining translation processes and offering seamless, real-time interaction, the device enables meaningful, barrier-free communication, making it a practical and impactful solution for a wide range of users.

Alternative Approaches

1. Sole Language Functionality Device: This alternative dropped the ASL functionality and focused on language translation between two parties. This would maximize portability of the device as we would not need certain hardware modules. This was dropped because of the presence of a multitude of competing technologies already on the market.

2. Portable, but non-wearable design: This alternative is in a larger form to increase the quality and speed of each of the hardware components. This could be a handheld or device that sits on a flat service. This was not pursued as it only served a part of a customer base. If we were to serve the purpose of helping travelers, having a device you need to hold would decrease the usability.

3.0 System Description

Authored: Yohan Kim

3.1 System Architecture

The Portable Language Translator system integrates three primary processing subsystems: Language Translation, ASL-to-Speech Translation, and Speech-to-Text Translation. These subsystems collectively ensure accurate and efficient real-time communication between users. Below is a detailed explanation of each subsystem.

The Language Translation subsystem processes spoken language inputs and converts them into another spoken language. The process begins by capturing audio using an I2S MEMS microphone, which is processed 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.

The ASL-to-Speech Translation subsystem focuses on translating American Sign Language gestures into spoken language. A USB 2.0 camera captures live video streams of the user's hand gestures. Using OpenCV, each video frame is processed with MediaPipe Hands to detect 21 key hand landmarks. These landmarks are normalized and fed into a pre-trained KeyPointClassifier model, which identifies the corresponding ASL letter. A stability timer ensures that only stable gestures are classified. Detected letters are concatenated to form words, which is further fixed using an integrated autocorrection system to correct spelling, grammar, and punctuation errors. Once the sentence is finalized, it is converted into speech using Google Cloud's Text-to-Speech API, allowing for direct audio output of the translated sign language.

The Speech-to-Text Translation subsystem provides transcriptions of spoken language for users relying on visual text outputs. Similar to the Language Translation subsystem, this process begins with audio input captured by the I2S MEMS microphone. The WebRTC VAD library detects speech, and the audio is processed by Google Cloud's Speech-to-Text API to generate a text transcription. The resulting text is displayed on the device's OLED screen, enabling users who rely on text-based communication to understand the spoken input.

Finally, a React-based web app allows users to set parameters, including the base language and voice preferences.

3.2 System Block Diagram

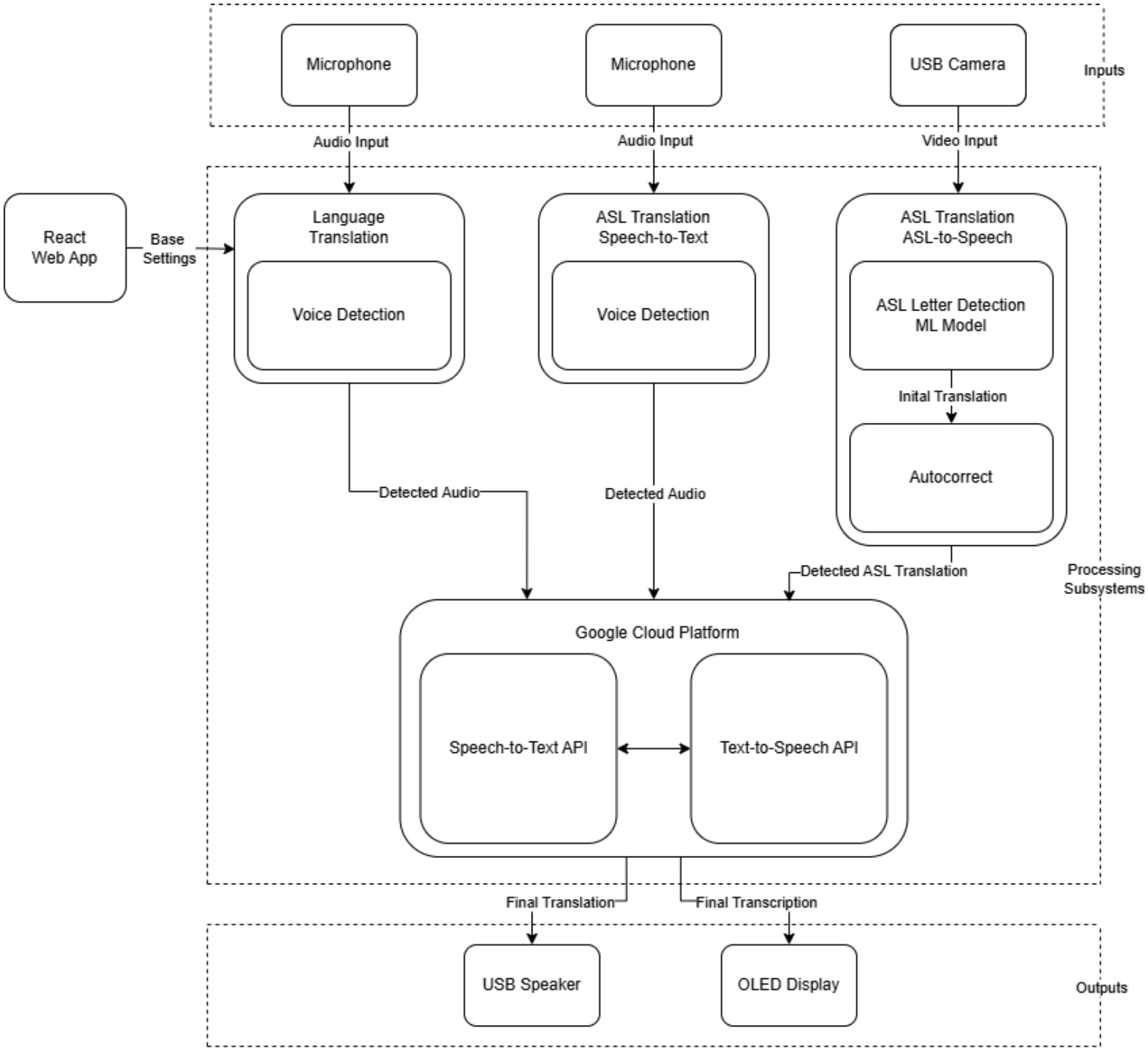


Figure 1. System Block Diagram of Translation Process

The system block diagram illustrates the key components and data flow of the device. The system is divided into three main stages: Inputs, Processing Subsystems, and Outputs.

Inputs

Audio input is captured by a single microphone and is used for both the Language Translation subsystem for translating spoken languages and the ASL Translation (Speech-to-Text) subsystem for converting spoken language to textual output. Video input is captured by a USB camera, which sends a live feed to the ASL Translation (ASL-to-Speech) subsystem to detect and classify American Sign Language gestures.

Processing Subsystems

The Language Translation subsystem includes voice detection, which isolates relevant audio segments and processes them for translation. The ASL Translation (Speech-to-Text) subsystem similarly uses voice detection to process audio for generating text transcriptions. The ASL Translation (ASL-to-Speech) subsystem uses an ML model for ASL letter detection and initial translation. It refines results with an auto correction module to fix any spelling, grammar, and punctuation mistakes before converting the output into speech. All three processing subsystems rely on the Google Cloud Platform, leveraging its Speech-to-Text API for transcription and Text-to-Speech API for speech generation.

Outputs

Translations and transcriptions are presented through two output devices: a USB speaker for audio output and an OLED display for visual output. The speaker plays back synthesized speech for both language translation and ASL-to-speech translations, while the OLED display provides real-time feedback, such as text transcriptions.

4.0 First Semester Progress

Authored: Cristian Palencia, Andrew Nguyen, Yohan Kim, Ryan Liao

Language Software

The language software logic for bidirectional translation is complete. The software listens for voice activity using a voice activity detection library. When it hears voice activity and then none, it knows to send the audio chunk the Google for processing. The Google Cloud transcribes while detecting the language, translates the text, and then performs text-to-speech in the target language. The audio is then played back through the speaker. The software automatically sets the language mode based on settings from the web application and the language detected to facilitate a conversation without manual input from the user. Testing data can be found in the appendix 7.3, tables 1-7

Web Application

The web application is currently able to set the base language of the user and the voice gender of the audio playback and send that to the language software via flask API.

ASL Software

This semester, the main goal for the ASL translation was to create a working system that could translate ASL letters into English sentences. This goal was achieved with the essential functionality of processing ASL input and translating into meaningful text.

The ASL translation was done by using a live video feed from a USB camera. Then MediaPipe Hands was used to detect and process hand landmarks which were further used as input in a pre-trained key point classifier model. The model identifies the corresponding ASL letter and then adds the letter into a string of words. Once no more letters are being detected, the initial output goes through an autocorrect script that utilizes Google's Gemini AI to fix any spelling, grammar, and punctuation mistakes.

Testing results for the ASL translation, outlined in appendix 3, 7.3, tables 8-9 shows the current translation capabilities. The raw output from the recognition model sometimes had errors such as incorrect letters, spacing problems, or substitutions. These errors were more common when the user was positioned 1 meter away from the camera, indicating that distance impacts recognition accuracy. Nevertheless, the integrated autocorrection system effectively resolved most of these issues, significantly improving the quality of the final output. Even at the maximum tested distance, the system reliably produced accurate sentences. The average time taken to process and correct a sentence was recorded at 3.57 seconds. This performance remained stable across various phrases, showing little variation irrespective of sentence length.

Hardware and Firmware

This semester's focus with hardware was to focus on finding and developing the core components of the project: microphone, display, camera, speaker, and microcontroller. Currently we have all of our hardware working off a Raspberry Pi 3 but ready to be moved to our Jetson Nano. We were able to successfully implement firmware files to create classes for the OLED display, microphone, and speaker. With these classes implemented we are able to have full and complete control over each hardware component. This functionality provides a great basis for continuing and testing development next semester from the Jetson Nano. One of our main requirements from a hardware standpoint was the functional distance of the microphone. Relevant data about the functional distance of the microphone can be found in Appendix 3 in Tables 10 and 11. These tables demonstrate that we met the requirement that language can be understood from a 1m range with our current microphone.

5.0 Technical Plan

Authored: Cristian Palencia

For the upcoming semester we have the following main developments and considerations in mind: hardware advancements, gesture model creation, Web App improvements, encasing, and integration.

5.1 Hardware Advancements

Next semester we want to ensure that we are able to implement and use a battery, allow our device to have some method of connectivity whether it be via bluetooth or internet, and finally ensure that we finalize all microcontroller decisions early on in the semester.

Battery

For our design we want to find a small battery with reasonable power capabilities. Additionally, we want to ensure that our battery can be recharged such that our device can continue being used even when the battery has been completely depleted. Thus, we will research appropriate batteries and implement the additional hardware required to make it rechargeable.

In order to implement this, we understand that it's not as simple as ordering a rechargeable battery, thus we will make sure to research the relevant hardware needed to ensure rechargeability. Furthermore, we have a requirement of a battery life of 2-8 hours per full charge depending on the current functional mode of the device. To ensure this requirement is met, we will evaluate the power consumption of our finalized parts at the beginning of the semester in order to ensure that we can find a battery that fulfills this requirement.

Finally, we will make sure that we research batteries with size and weight as a heavily weighed factor to ensure that our device remains portable.

With these considerations in mind, we hope to find our battery early on into the semester since finding a good battery is also dependent on having all our hardware finalized.

Since this is a hardware component Cristian Palencia will be in charge of handling this in the first weeks of the semester.

Device Connectivity

Our design is expected to be a fully stand alone, portable device that has no external wire connections. Thus we plan to implement a connectivity module into our microcontroller.

We plan on either implementing internet or bluetooth connectivity. Internet connectivity would allow the device to be fully independent while bluetooth would require dependence on a phone to connect to. We will evaluate the positives and negatives for both design choices. Additionally, we will find an appropriate USB module as it will be a simple integration since it will not need to integrate with GPIO pins.

Since this is a hardware component Cristian Palencia will be in charge of handling this in the first weeks of the semester.

Finalize Microcontroller

We moved over to a Jetson Nano at the end of the semester due to its computational strength. Very early on in the semester we plan on fully making the transition to the Jetson Nano in order to optimize our design speeds. In order to do this we will face the following challenges: outdated OS and size.

From initial attempts with the OS we found that its quite outdated (Ubuntu 18) and caused some library conflicts. We will determine a workaround for the outdated OS to allow our software to be able to run on the Jetson Nano.

Despite the computational strength of the Jetson Nano it contradicts our portability requirement for the device. Thus, we will work closely with Professor Osama in order to make plans to reduce the size of the Jetson Nano and include only hardware components that are strictly required for functionality of our product.

Since this is a hardware component Cristian Palencia will be in charge of handling this in the first weeks of the semester but will work closely with Yohan Kim and Andrew Nguyen to ensure library compatibility is properly met.

5.2 Gesture Model Creation

We will develop a gesture model that reads in gestures and translates them to English. We want to focus on implementing this with 5-10 gestures. For all Gesture Model related tasks it will be a collective effort between Andrew Nguyen, Ryan Liao, and Yohan Kim since we understand this is one of the most difficult tasks of the project.

Research

We want to allocate the beginning of the semester to researching the best way to take video input as data and the best way to structure our model to handle that data.

Collect Training Data

We will then start collecting video data of different gestures with our USB camera. This will include extensive recordings of the gestures repeated in a variety of settings if a premade dataset is not readily available.

Develop and Train Model

Once we have all of our training data we will begin to train a model on gesture recognition for use in our ASL detection software using a neural network with tools from Tensorflow.js. The video will be compressed into an image that represents the entire gesture and be trained off of that.

5.3 Web Application

We want to make minor yet relevant improvements to our web application such that it adds more functionality but also is more user friendly.

GCP collection

We will add the full google cloud platform catalog of voices such that users are able to have full choice over the voice they would like to hear output from the device.

Andrew Nguyen developed the prototype version of the Web Application and so will be in charge of all modifications to it. He will work on it in the first week of the semester such that he can then move his focus towards aiding in the development of the Gesture Model along with Yohan Kim and Ryan Liao.

Update UI

We will make the UI more visually pleasing and user friendly. This will be done by modifying relevant HTML and CSS files.

As previously mentioned, Andrew Nguyen developed the prototype version of the Web Application and so will be in charge of all modifications to it. He will work on it in the first week of the semester such that he can then move his focus towards aiding in the development of the Gesture Model along with Yohan Kim and Ryan Liao.

5.4 Encasing

We will design an encasing that permits the user to clip the device comfortably and easily with all parts mounted appropriately.

Collect Measurements

We will collect relevant measurements of the different hardware components of the project before beginning 3D modelling. This will serve as a good starting point to begin designing.

Yohan Kim and Cristian Palencia will work on this together once all the hardware has been ordered and delivered.

3D Model and Print

Once measurements have been completed we will begin to consider the physical design architecture. This will be done first in drafts and then it will be moved over to a 3D modeling software of our preference like Fusion 360 or Onshape.

Yohan Kim will be primarily in charge of the design with the assistance of Cristian Palencia to ensure proper communication about hardware components and their physical mounting.

Mount Components

Once we have finished a preliminary 3D print we will start mounting all hardware components to ensure that everything fits as we need. We will make design revisions and return to modelling as we deem necessary. Revision consideration will depend on portability behavior. Thus, we shall test if the design allows the user to clip on the device easily and comfortability with all relevant hardware in place.

Yohan Kim and Cristian Palencia will work closely with this to ensure hardware integrity and functionality is kept during mounting.

5.5 Integration

Software Integration onto Microcontroller

Once we have finalized all software and the microcontroller we will fully integrate the two. This is to ensure that the microcontroller can properly run all our code off of the microcontroller OS and we can make modifications as needed. This is with the primary focus on the gesture model and the expectation is that other relevant completed software components will have already been tested on the microcontroller earlier in the semester.

This will be led primarily by Cristian Palencia and he will communicate with other members of the team to understand library dependencies and requirements. We will dedicate a large portion of March to ensure that this is done thoroughly and with time to fix any issues that may arise.

Software and Hardware Integration

With code fully running on the microcontroller, we will then ensure that the software is able to fully interface with the hardware in a coherent manner that is reflective of the full device behavior.

This will be a team effort to ensure full system integration. We will dedicate a large portion of March to ensure that this is done thoroughly and with time to fix any issues that may arise.

6.0 Budget Estimate

Authored: Cristian Palencia

Item	Description	Cost
1	Adafruit Microphone	$7
2	Jetson Nano	$215
3	USB Speaker	$15
4	USB Camera	$12
5	OLED Display	$13
6	Wireless Module	$10
7	Battery	$40
8	Bluetooth Module	$10
	Total Cost	$322

The following includes all relevant parts we envision ourselves using for a final design. We currently include the prices for both the Wireless and Bluetooth module since we aren't completely sure which we will go with yet. Another relevant note is on the price of the Jetson Nano. It is our most expensive part as it helps significantly with performing operations locally rather than through the cloud. Additionally, we may consider implementing a more recent Jetson Nano, the Jetson Nano Oran which instead costs $500. In this case our final budget cost would be $607 instead of $322. This could be a relevant consideration just to ensure maximum performance with the most recent OS version to support all our needed libraries.

7.0 Attachments

7.1 Appendix 1 – Engineering Requirements

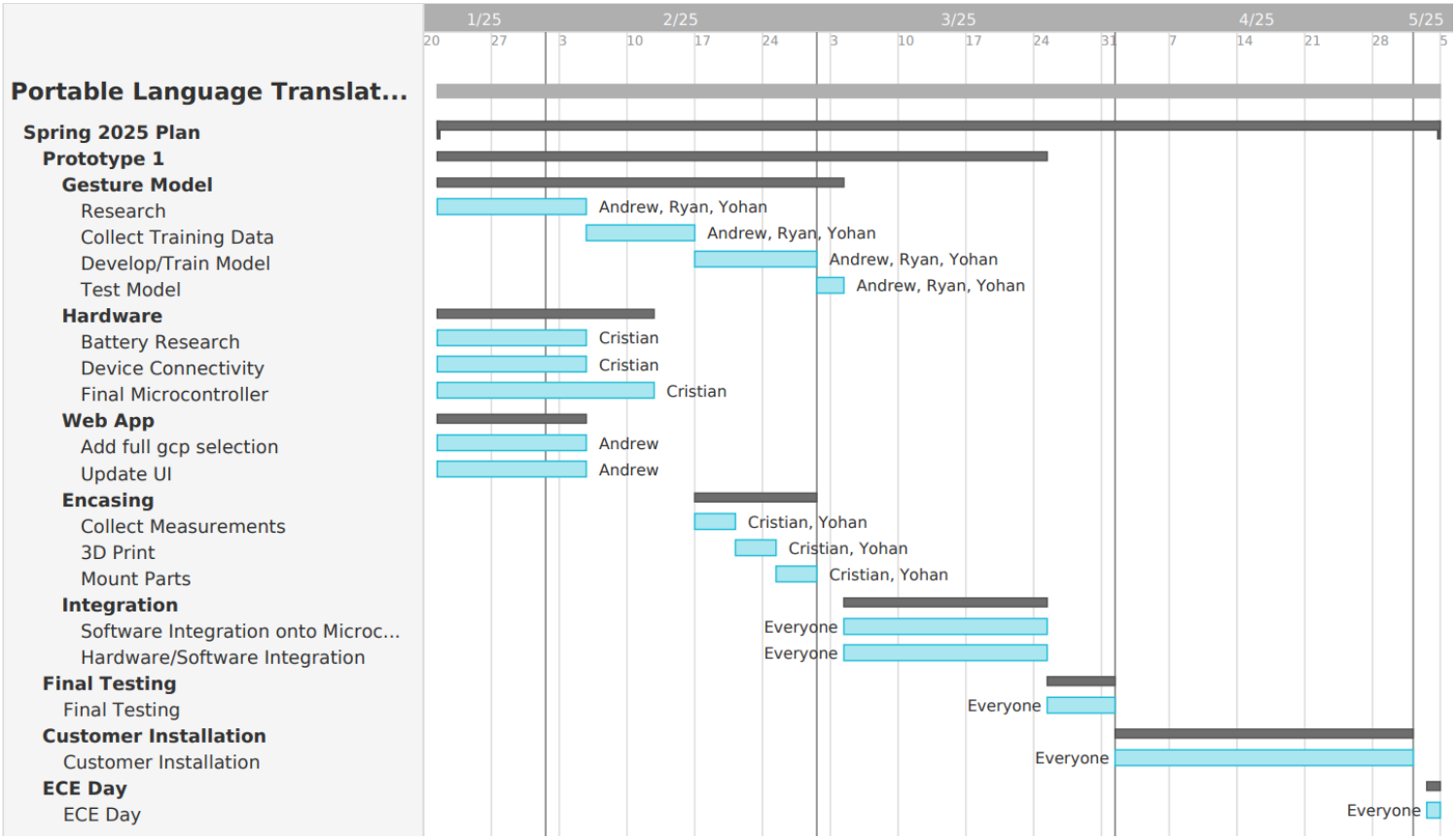
Authored: Ryan Liao

Team #23 Team Name: PLT

Project Name: Portable Language Translator

Requirement	**Value, Range, Tolerance, Units**
Case Dimensions	140mm x 100mm x 80mm
Power	5V battery
Battery Life (Speech)	< 3 sec
Battery Life (ASL)	< 4 sec
Speech Detection Range	50 - 70 dB within 1 m
ASL Detection Range	1 m
Weight	< 200 g

7.2 Appendix 2 – Gantt Chart



7.3 Appendix 3 – Other Appendices

Language Accuracy Tests:

Phrase Said	Phrase Recorded	Translated Phrase	Errors
Hello, how are you?	hello how are you	안녕하세요. 어떻게 지내세요	Transcription: no errors Translation: no errors
Can I get a large coffee with two sugars and no milk, please?	can I get a large coffee with Two Sugars and no milk	Two Sugars와 우유 없이 큰 커피 한 잔을 마실 수 있나요?	Transcription: no errors Translation: no errors
Excuse me, how do I get to the nearest train station?	excuse me how do I get to the nearest train station	실례합니다. 가장 가까운 기차역까지 어떻게 가나요?	Transcription: no errors Translation: no errors

Table 1: English - Korean Translation

Phrase Said	Phrase Recorded	Translated Phrase	Errors
Do you have fresh strawberries or are they out of season?	you have fresh strawberries or are they out of season	¿Tienes fresas frescas o están fuera de temporada?	Transcription: missing word Translation: no errors
Can I try this jacket on in a medium size?	can I try this jacket on in a medium sized	¿Puedo probarme esta chaqueta en talla mediana?	Transcription: minor error Translation: no errors
What time are we meeting tomorrow for lunch at the new restaurant?	what time are we meeting tomorrow for lunch at the new	¿A qué hora nos reuniremos mañana para almorzar en el nuevo?	Transcription: missing word Translation: no errors

Table 2: English - Spanish Translation

Phrase Said	Phrase Recorded	Translated Phrase	Errors
¿Hola, tienes fresas frescas o están fuera de temporada?	Hola tienes fresas frescas o están fuera de temporada	안녕하세요. 신선한 딸기가 있나요? 아니면 제철이 아니나요?	Transcription: no errors Translation: no errors
¿Puedo probarme esta chaqueta en talla mediana?	puedo probarme esta chaqueta en talla mediana	이 재킷을 미디엄 사이즈로 입어봐도 될까요?	Transcription: no errors Translation: no errors
¿A qué hora nos reuniremos mañana para almorzar en el nuevo restaurante?	a qué hora nos reunimos mañana para almorzar en el nuevo restaurante	우리 내일 몇 시에 새 식당에서 점심 먹으러 만나요?	Transcription: no errors Translation: no errors

Table 3: Spanish - Korean Translation

Phrase Said	Phrase Recorded	Translated Phrase	Errors
안녕하세요. 어떻게 지내세요	안녕하세요 어떻게 지내세요	Hola, cómo estás	Transcription: no errors Translation: no errors
우유 없이 설탕 그게 큰 커피 주세요	우유 없이 설탕 그게 큰 커피 주세요	Me gustaría un café grande con dos azúcares y sin leche.	Transcription: no errors Translation: no errors
실례합니다. 가장 가까운 기차역까지 어떻게 가나요?	실례합니다 가장 가까운 기차역까지 어떻게 가나요	Disculpe, ¿cómo llego a la estación de tren más cercana?	Transcription: no errors Translation: no errors

Table 4: Korean - Spanish Translation

Phrase Said	Phrase Recorded	Translated Phrase	Errors
신천한 딸기 있나요 아니면 제철이 아니나요	신천한 딸기 있나요 아니면 제철이 아니나요	Are there any fresh strawberries or are they out of season?	Transcription: no errors Translation: no errors
이 재킷을 M 사이즈로 입어봐도 될까요	이 재킷을 M 사이즈로 입어봐도 될까요	Can I try that jacket in size M?	Transcription: no errors Translation: no errors
우리 내일 몇 시에 새 식당에서 점심 먹으러 만날 거야	우리 내일 몇 시에 새 식당에서 점심 먹으러 만날 거야	What time are we meeting for lunch at the new restaurant tomorrow	Transcription: no errors Translation: no errors

Table 5: Korean - English Translation

Phrase Said	Phrase Recorded	Translated Phrase	Errors
Hola, cómo estás	Hola cómo estás	Hello how are you	Transcription: no errors Translation: no errors
¿Puedo tener un café grande con dos azúcares y sin leche por favor?	puedo tener un café grande con dos azúcares y sin leche por favor	can I have a large coffee with two sugars and no milk please?	Transcription: no errors Translation: no errors
Disculpe, come alcanzo llegar al estacion de tren mas cercano?	Disculpe cómo alcanzo a llegar a la estación de tren más cercano	excuse me, how do I get to the nearest train station?	Transcription: minor errors Translation: no errors

Table 6: Spanish - English Translation

Language Latency Tests:

Phrase	Word count	Latency (seconds)
Hello	1	0.36
Hello, how are you?	4	0.63
I would like to have a water please	8	0.67
Where is the nearest hospital? I am having a serious allergic reaction	12	0.87
I always enjoy quiet evenings at home, reading a good book with a cup of tea	16	1.21
Let’s plan a weekend getaway soon. It’s been too long since we all had some fun together outside at the beach	21	1.22
When we moved into the new neighborhood, we didn’t know anyone, but our neighbors were so welcoming. They even brought over cookies and invited us to their backyard barbecue. It’s been a great place to live ever since	38	1.92

Table 7: Latency Measurements - Data collected for latency for different sized phrases

Phrase	Distance from Camera	Before Autocorrect	After Autocorrect	Errors?
Where is that?	.33 meters	Where is that	Where is that?	No
Where is that?	.67 meters	Where is that	Where is that?	No
Where is that?	1 meter	Whexe i ss thaa t	Where is that?	No
The car is red.	.33 meters	The car is re d	The car is red.	No
The car is red.	.67 meters	The car is r e d	The car is red.	No
The car is red.	1 meter	The cab is r e d	The cab is red.	Yes, B was recorded instead of R
That sounds so exciting!	.33 meters	That sounds so exciting	That sounds so exciting!	No
That sounds so exciting!	.67 meters	That sounds sso exciting	That sounds so exciting!	No
That sounds so exciting!	1 meter	That soumdss so exciting	That sounds so exciting.	Yes, it did not add the exclamation mark
Hi my name is Yohan.	.33 meters	Hi my name is yohan	Hi, my name is Yohan.	No
Hi my name is Yohan.	.67 meters	Hi my name iss yohan	Hi, my name is Yohan.	No
Hi my name is Yohan.	1 meter	Hi my mame is yoham	Hi, my name is Yohan.	No
I went to school today and learned how to read	.33 meters	I w e n t to school and lea r ned how to read	I went to school and learned how to read.	No
I went to school today and learned how to read	.67 meters	I we n t to scool a nd lear n ed how o read	I went to school and learned how to read.	No
I went to school today and learned how to read	1 meter	I we nt sto sc hool and learn e d how to read	I went to school and learned how to read.	No

Table 8: ASL Testing Results

Phrase	Latency (seconds)
Where is that?	3.62
The car is red.	3.5
Hi my name is Yohan.	3.92
That sounds so exciting!	3.55
I went to school today and learned how to read	3.75

Table 9: ASL Latency Results

Quality Score	Meaning
0	Inaudible
1	Muffled and distant
2	Language can be understood
3	Clear audio
4	Perfect audio recording

Table 10: Audio Recording Quality Table

Distance (m)	Whisper	Conversational Speaking	Yelling / Loud Speaking
0.25	3	3	4
0.50	3	3	3
0.75	2	3	3
1	2	2	3
1.25	2	2	3
1.50	1	1	3
1.75	1	1	3
2	1	1	3

Table 11: Functional Microphone Distance Table

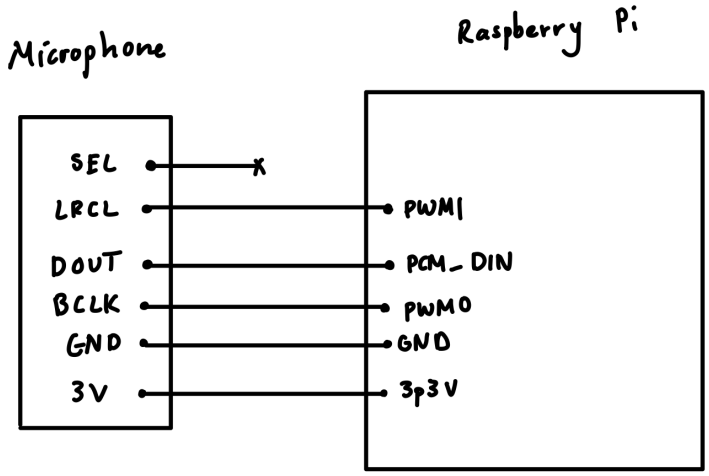


Figure 2: Microphone Wiring Diagram

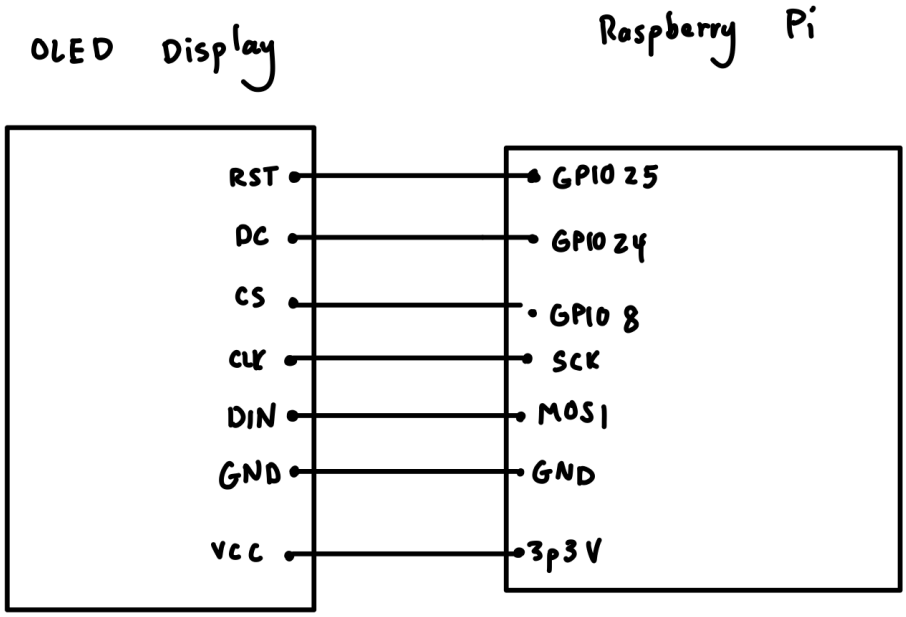


Figure 3: OLED Display Wiring Diagram

7.4 Team Information Sheet

Biographical 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