# Real-Time Sign Language Translator

# ECE 445 Design Document - Fall 2021

---


Project #27

Gene Lee(genel2)

Kaelan To(kto3)

Professor: Wei Hu

TA: Bonhyun Ku

# Table of Contents

# Introduction

## Problem and Solution Overview

Technology is improving rapidly, and with that, it serves the purpose of making our lives easier. We want to leverage the technology available to us and further integrate those with disabilities more into our society; specifically deaf individuals in an academic setting. When we imagine students with hearing impairments working with others, we think of the other students having to wait for the student to type out their thoughts. However, brainstorming/bouncing off ideas requires rapid discussion in order to spark good ideas. Sometimes typing may not be as fast (which might hinder the group) or even may not be accessible to students (especially in K-12) in classroom settings.

We propose a portable real-time sign language translator to solve this problem. We would utilize computer vision to differentiate the hand signs and feed the visual input to a microcontroller and give audio feedback (sound translation of the hand sign). Given enough time, we do also want to add a feature of speech to text in order to help the deaf student understand their peers too. All in all, this portable system would assist them in communicating with teachers, but most definitely help deaf students work in a team with other students efficiently.

## Visual Aid


Figure 1: Application of device in classroom setting

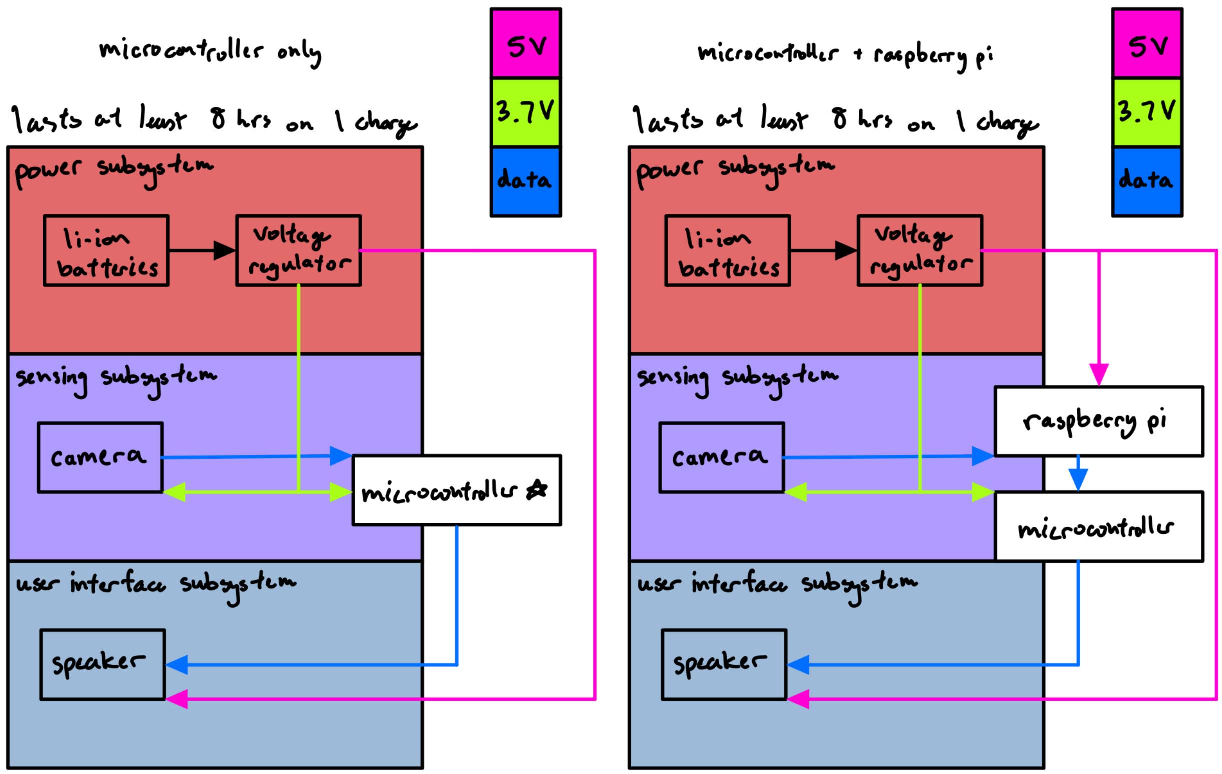
## High-level Requirements

For the device to accomplish the minimum goal for helping deaf students in K-12 classrooms communicate with their peers, the system must reach the following requirements:

- Ability to identify ASL and translate into English in real-time (threshold set to be within 1.5 seconds)
- System is lightweight/portable (desired size set to be a volume of 15x15x15cm)
- Battery lifetime of at least one school day (8 hours - also subsystem requirement)

# Design

## Block Diagram




## Physical Design

This entire device will be just one portable box. This device is meant to be used in a classroom setting for school children so it should be easily stored within a classroom or backpack. As such, it would just be a 15x15x15cm box with a camera on the front, a simple on/off switch, and a built-in speaker. It would be placed upon classroom tables in front of the deaf school children such that our visual sensor, the camera, would be able to take in the signed words and send it to our system for processing and translation into English.

# Subsystems

## Visual Input

Overview: The camera will be the main sensor used for our device. The camera will be used to read sign language inputs from the user and send the information to the central processor.

Requirements: Due to the speed that users will be signing at, and the nature of the pose estimation model, the frames per second and even the resolution needed do not need to be that high.

Parts: Sony Spresense 5MP Camera Board/Arducam Mini Module Camera Shield

| Requirement | Verification |
| --- | --- |
| Must be able to capture video at a frame rate when signing at a moderate “conversational” speed | |

## Central Processor

Overview: In charge of putting together all the subsystems and processing the inputs to generate outputs. It will take in the visual input and process it using a pose estimation model to generate the necessary audio output to the speakers.

Requirements: Needs enough computing power to run a pose estimation model. Needs to be power efficient enough so that the battery is able to provide enough power to reach our minimum requirement.

Parts: ARM Cortex M4/Raspberry Pi, USB Shield

Remarks: We will attempt to use the Cortex M4 microcontroller first and try to build our own custom board that will be able to handle all subsystems centrally AND run the pose estimation model. If this fails, we plan on falling back to using a Raspberry Pi to deal with SOLELY the pose estimation functionality of the device. The PCB we design will still be in charge of integrating all the other subsystems together.

| Requirement | Verification |
| --- | --- |

| | |
|---|---|
| | |

## Audio

Overview: Responsible for producing audio that it receives from the central processor and outputs it through the speakers.

Requirements: It must be loud enough to be heard by multiple people nearby the device.

Parts: Adafruit MAX98357, DB Unlimited SW401404-1

| Requirement | Verification |
|---|---|
| | |

## Power

Overview: Responsible for powering all the components in the device.

Requirements: Must be able to power the device for the duration of a typical school day, which would be 8 hours.

Parts: Li-ion Battery, Voltage Regulators

| Requirement | Verification |
|---|---|
| | |

## Encasing

Overview: Responsible for holding all components together while providing the user with a simple way to interact with the device and achieve our goal.

Requirements: Needs to be small and lightweight (15x15x15 cm) in order to be easily carried around. Needs to be able to fit all of the components and have openings for the camera, speaker, and battery. Needs to be able to let the user charge/change the battery easily.

Parts: 3D Printed

| Requirement | Verification |
| --- | --- |
| | |

## Tolerance Analysis

The main issue that we may run into is that the microcontroller may not be powerful enough to take the camera input and process the data. To remedy this issue, we have considered a design that involves a Raspberry Pi as well, which has shown to have camera and audio capabilities. Another potential issue may be the camera frame rate, as it may not be high enough to actually interface well with the microcontroller. The solution to this problem would be to find a stronger, more powerful camera, but that may affect our power subsystem and likely will require the use of a Raspberry Pi. Another issue that may come into play is the pose estimation library we will be using. The pose estimation library may not be sufficient enough to differentiate between different signs, so we might have to train our model ourselves, which will impact our use cases.

# Cost and Schedule

## Cost Analysis

| **Part** | **Part Number** | **Unit Price** | **Quantity** | **Total** |
| --- | --- | --- | --- | --- |
| Camera Module | ArduCAM-2MP-Plus | 25.99 | | |
| Microcontroller | EFM32PG1B100F256GM32-C0 | 4.06 | | |
| Microcontroller | Raspberry Pi Model B+ | 29.95 | | |

| | | | | |
|---|---|---|---|---|
| USB Shield | 1528-2873-ND | 2.95 | | |
| Audio | MAX98357A | 2.44 | | |
| Li-Ion Battery | BL2200F6034501S2PPMK | 19.16 | | |
| Voltage Regulator | S-1313D33H-A4T2U3 | 1.19 | | |
| LOOK MORE INTO PARTS AFTER DESIGNING SCHEMATIC | | | | |
| PCB | N/A | 30.00 | 1 | |

# Schedule

| **Week** | **Gene Lee** | **Kaelan To** |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# Ethics and Safety

The main safety concern when working with electronics is with the batteries. Batteries can be potentially hazardous as they can leak and even explode in the wrong conditions. To comply with IEEE(I.1), we will do our best to not push the batteries past the point of danger. Another possible safety concern is with the microcontroller, as if pushed too far, it may overheat and become a fire hazard, similarly to the batteries. To avoid this issue, we will stay within spec of the devices that we use.

# Citations


- https://learn.adafruit.com/how-to-choose-a-microcontroller/the-microcontrollers-in-adafruit-products
- https://www.mouser.com/Semiconductors/Embedded-Processors-Controllers/Microcontrollers-MCU/ARM-Microcontrollers-MCU/ARM-Cortex-M4-Core/_/N-a85pc?P=1z0wa45Z1yztkji&pop=48561&gclid=Cj0KCQjws4aKBhDPARIsAIWH0JUJ3LbzQF8fnkfyH-dGgkeWBP_PC5kJ1A5PXVQgIBveFrGWRbh3HUsaAiNnEALw_wcB


Design Doc To-Do:

Draft out schematics based on M4 dev boards and accommodate our functionalities

- Is power sufficient via the Li-Ion battery?
- Voltage regulator regulates to 3.3 V; is this enough? (also check mAH)
- Main schematic (with M4 as processor for pose estimation)
- Back-up schematic (with raspberry pi as processor for pose estimation)

Refine block diagram

Refine visual aid

Scope out parts

- Lower cost
- Back-up circuit in case m4 does not work out with pose estimation

Remarks:

Simulations not really needed since our main task is to piece together our modules (not really circuit based)

- At most see if we need capacitors to prevent voltage spikes and/or pullup resistors