Wearable Communication Device For Deaf And Mute

ECE 445 Design Document

Team 19
Minho Lee and Andrew Ko and Yihan Ruan
ECE 445 Project Proposal - Fall 2021
TA: Stasiu Chyczewski
9/30/2021

Table of Contents

1 Introduction
1.1 Problem

Technology is evolving rapidly and provides benefits in almost every aspect of our lives. However, most solutions aim to make the majority feel convenient where they can make more profits. Consequently, we thought it was a good idea to design our project to focus on the people with disabilities and bringing comfort in their everyday lives. Most of the time, random encounters and socializing proves to be difficult for Deaf and Mute people as sign languages are not one of the popular options in the verbally dominant world. In order to help with the cause and prevent them from being discouraged, our wearable communication device will provide a new way. The user will also be able to see what he types in, essentially on what is like your regular electronic chatroom. This way, deaf and mute people will be able to have conversations with people other than just those around them who know how sign languages work.

1.2 Solution

One could argue that there are many alternatives in which the user can communicate with people without disabilities, but most of the time, they either require the internet or that you have your phone with you and have to have certain apps downloaded at the time such as google translate. However, this isn't the likely scenario for many deaf/mute people. Also, the typing on your phone can never get as fast as to allow for real time communication. With our wearable device, It won't be a hassle and help the mute/deaf communicate vocally real time. The device is attached with a steno keyboard that allows the user to type at the talking speed which is then converted immediately to speech through TTS and played through the speaker on the belt. The interactor will be able to respond via talking to the mic built in with the display module that will put his words on screen.

1.3 Visual aid

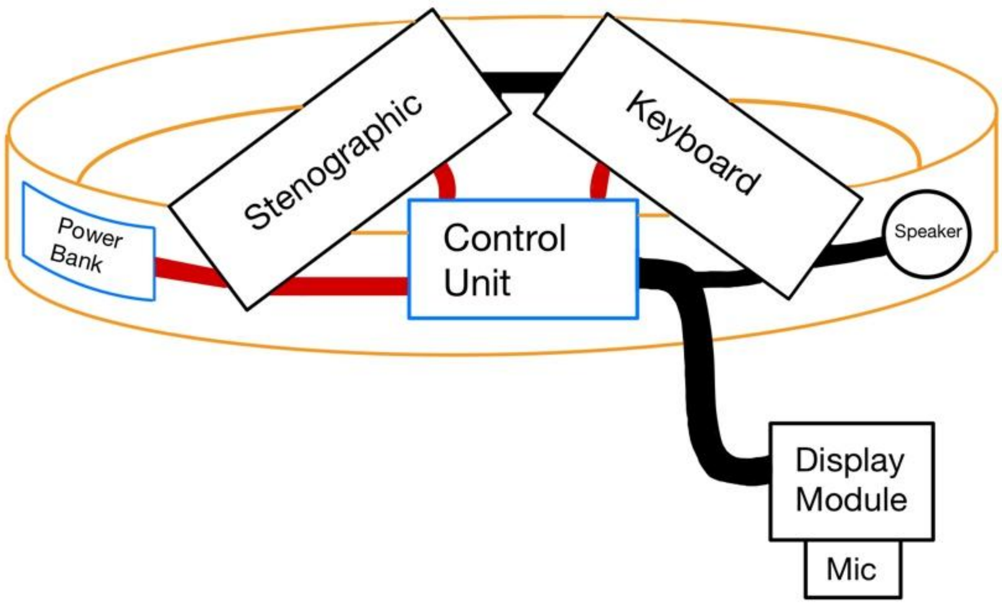


Figure 1. Visual aid

1.4 High-Level Requirements

- Users can have real time communication without using sign languages. The average talking speed is at 150 words per minute and a fast stenographer can type up to 300 words per minute. Users should be able to type with the keyboard as fast as 150 words/min.
- The speed at which the users can see the interactors' words on the display should be delayfree to support real time, almost instantaneously, less than a second.
- The speaker should be able to produce clear and audible sound at a talking distance away. Following the covid guideline, it should be audible at 6 ft away.

2 Design

2.1 Block Diagram

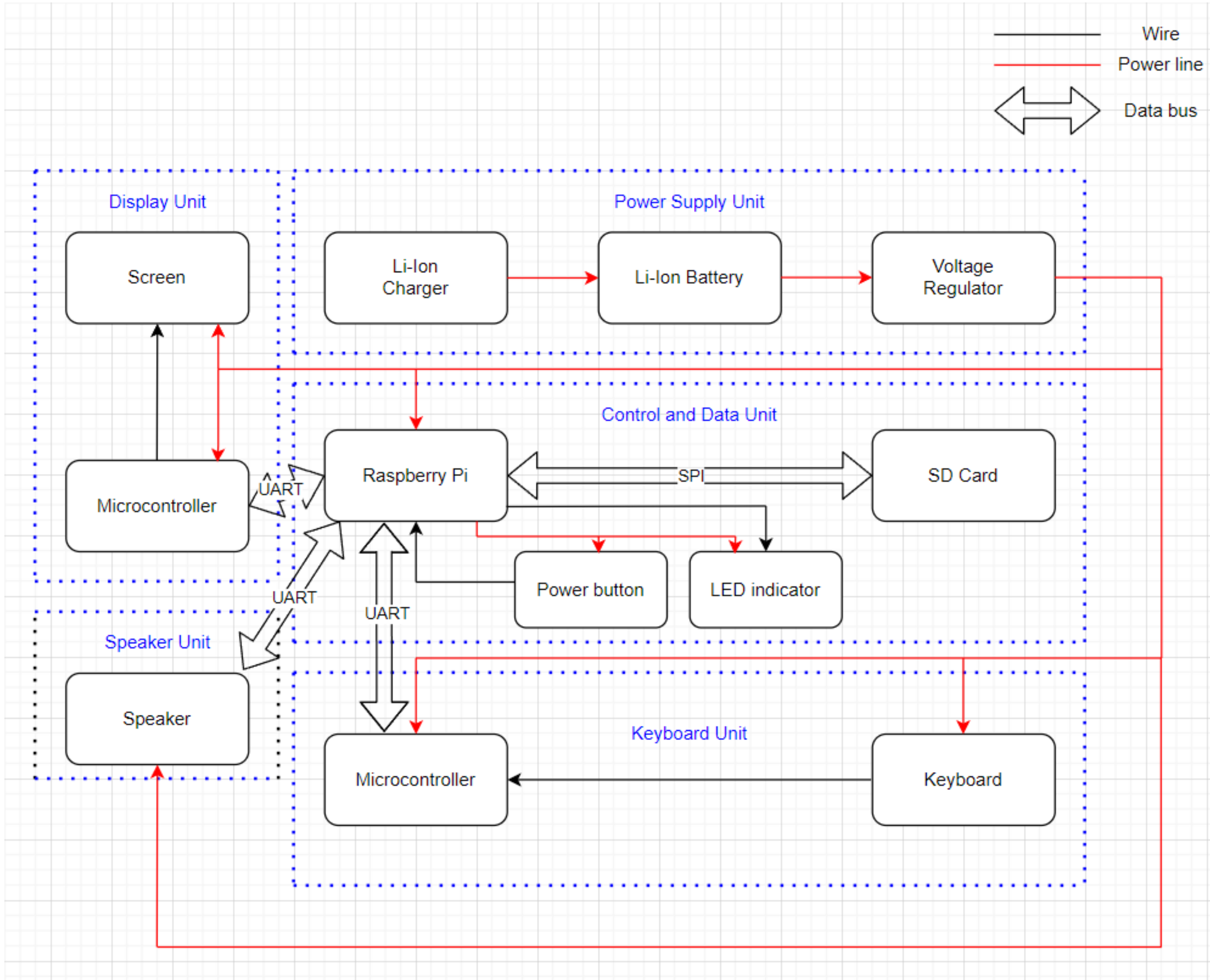


Figure 2. Block diagram for the system

2.2 Power Supply

Input: Rechargeable 3.7V Li-ion Battery
5V from micro-USB type B
Output: 5V, 2A for Raspberry Pi 4.

The power supply provides Raspberry Pi with 5V with a switching current limit of 2A. The battery capacity will be around 10400 mAh for approximately 5 hours of use at full load.

According to Raspberry Pi power requirements [5], it is recommended to use 5V 3A but we decided to provide 2A since it is not likely that the Pi model 4 will require more than 2A.

Devices	Voltage/Current requirement
Raspberry Pi 4B	5V±5%, 1.25A (Max), 1.2A (Avg.)
Keyboard	5V±5%, (mcu max 27mA)
Display	5V±5%, 380mA
Mic	1.8-3.3V, 0.8mA
Speaker	5V±5%, 100mA

At maximum load, the power required will be little less than 2000 mA including Raspberry Pi, display, and speaker and mic. Since all the devices other than Raspberry Pi will be powered by the Pi, the output connector is one USB-C type at 5V 2A for the Pi.

2.2.1 Li-ion Battery

The battery will consist of four cells of li-ion battery connected in parallel to increase the overall capacity. It will be charged through the charging IC MCP73833T. The battery is protected from overcharging, and over discharging with protection IC AP9101C.

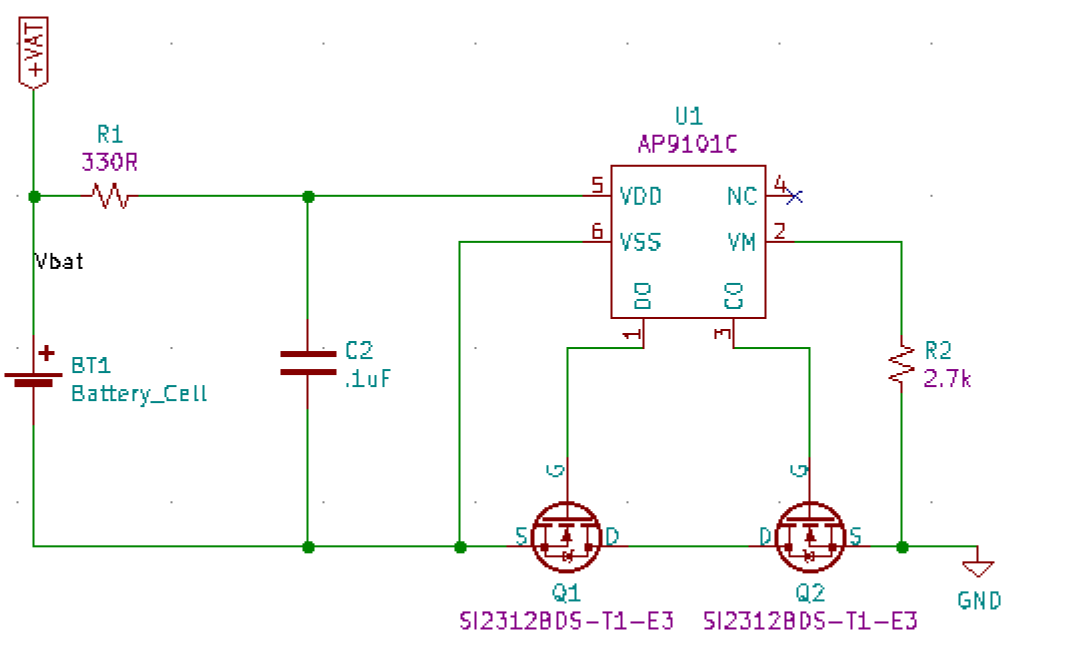


Fig3. Battery Protection Schematic

The pin VM is connected to a high resistor that is used to detect voltage difference between VM and VSS to control battery charging and discharging by DO and CO pins.
The pins DO and CO are for control of the battery in situations of voltage/current overdischarge and overcharge by changing the gate voltage of the mosfets.

2.2.2 Li-Ion Charger
The battery will be charged through a charging IC, MCP73833T. The power supply will come through micro-usb-B input. With maximum 1A output, our 10400 mAh battery will fully charge in around 10 hours. The charging cycle is in three phases, conditioning, constant current and constant voltage

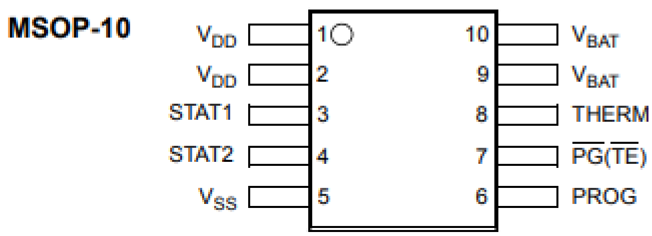


Fig. Pinouts for MCP73833T

The 3.7V Li-Ion battery can be charged through 4.2V which is the maximum voltage of the battery at full charge. For 1A charge current, the PROG pin will be connected to a 1k Ohm resistor.

$$I_{REG} = \frac{1V}{R_{PROG}}$$

The three pins PG(TE), STAT1,2 will be connected to 3 LEDs connected to the charge that will tell (1) input power is good, (2) Charge is in progress, and (3) Charging complete. The meaning of a good input power is that the input voltage is adobe the threshold voltage of the IC.

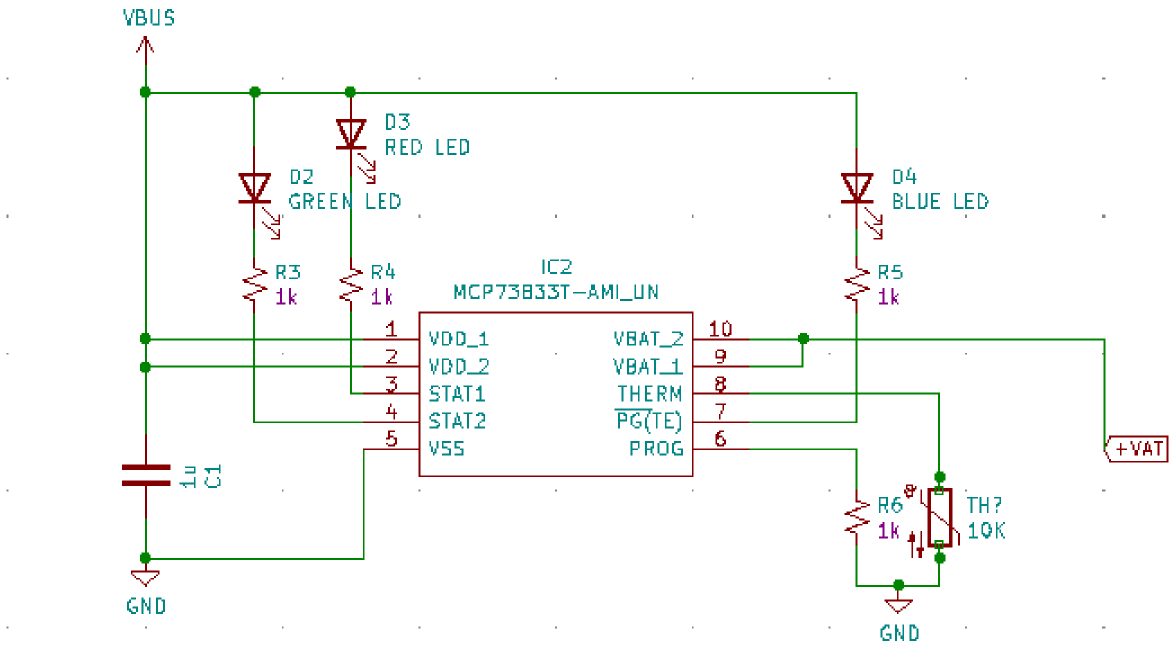


Fig4. Charger Schematic

For the temperature monitoring pin, THERM, is programmed using a 10k Ohm thermistor, which will prevent overheating of the IC.

2.2.3 Voltage Regulator (Boost Converter)

The voltage booster will supply our Raspberry power board with 5V at 3A. Using LTC1700, a constant current DC-DC step-up inverter, the input voltage of 3.7V ~ 4.2V from the battery cells will be converted to 5V 2A output.

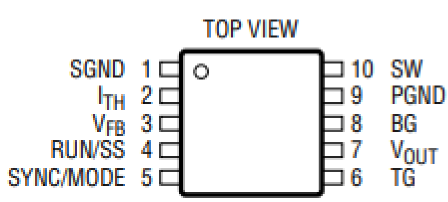


Fig. Pinouts of LTC1700

By connecting sync/mode Pin to Vout, LTC 1700 will go into a Burst Mode at low load current. This allows us to achieve 2A output.

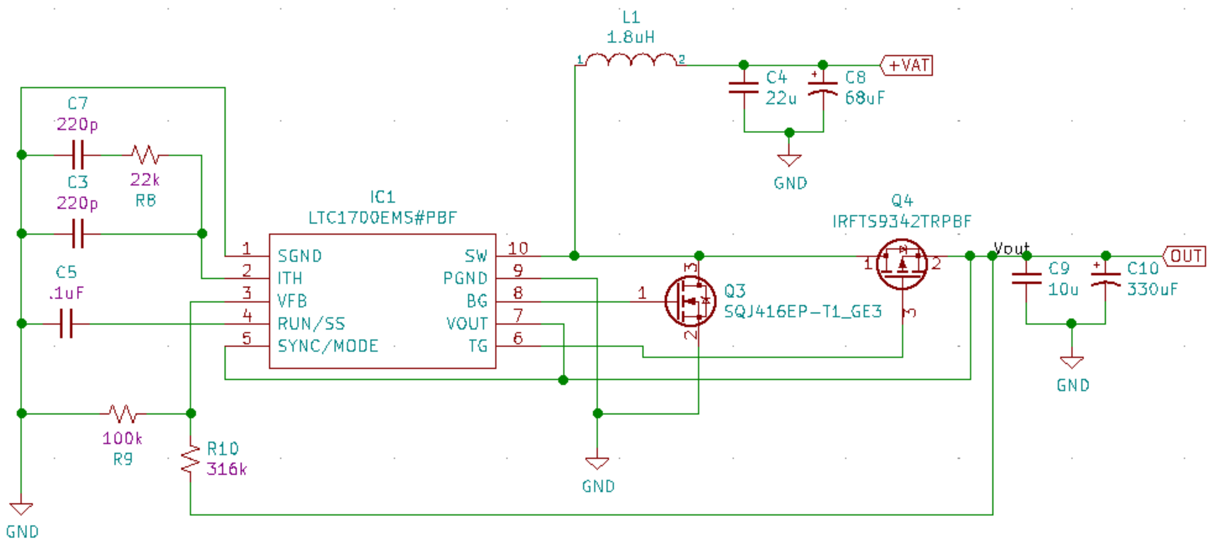


Figure 5. Boost Converter Schematic

There are a N-channel and a P-channel MOSFET. The N-channel is turned on while the P-channel is off during normal operations, and P-channel turns on when N-channel is off. The P-channel is turned on until either inductor current is reversed or the next duty cycle begins. The mosfets are selected based on their R_{DSON}. The following equations are used to calculate the required on-resistance of the mosfet.

First, the duty cycle of mosfet is calculated by $1 - V_{IN}/V_{OUT}$= .16 ~ .26 % for 3.7 and 4.2 V input.

Then, we use the equation for $L_{MINBURST}$ as our duty cycle falls below 36%.

$$L_{MINBURST} = \frac{V_{IN(MAX)}(DC)}{(f)(0.66)\left(\frac{I_{OMAX}}{1-DC}\right)}$$

Where the frequency = 530k Hz

Then, $L_{MINBURST}$ = 0.807 uH.

This is the minimum value for our inductor

For Burst Mode operation, we need low ripple current of around .4*I_{0MAX} = .8A

Though our minimum inductor size is .81 uH, we will be using 1.8uH to further reduce ripple current.

With the L = 1.8uH, we can use the equation below for actual ripple current.

$$\Delta I_L = V_{IN}\left(\frac{DC}{fL}\right)$$ Then, ΔI_L = 1A (3.7V), 0.7A (4.2V)

Now, to get the $R_{DS(ON)}$ we use the following equation.

$$R_{DS(ON)(MAX)} \cong \frac{\Delta V_{SENSE}}{\left(\frac{I_{O(MAX)}}{1-DC}+\frac{1}{2}\Delta I_L\right)(\rho_T)}$$ Where ΔVSENSE = 63mV and $\rho \cong 1.2$

For N-channel, $$R_{DS(ON)(N-CHANNEL)} = \frac{63mV}{\frac{I_{O(MAX)}}{1-D}+0.5(\Delta I_L)}$$ = 3.2 mOhm

For P-Channel, $R_{DS(ON)(P-CHANNEL)}$ = (3.2)(.9) = 2.88mOhm

Where the .9 is from fig. X, provided from the LTC1700 datasheet with duty cycle calculated above.

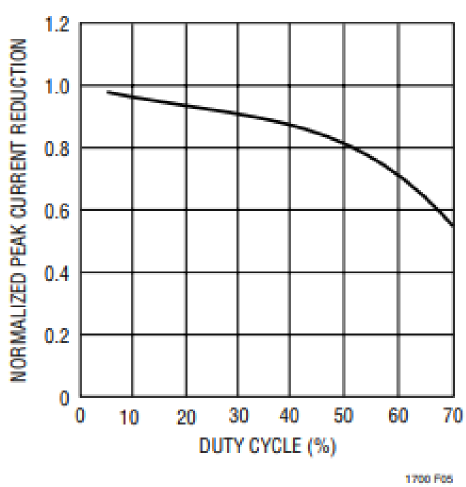


Fig x. Max output current vs. Duty Cycle

The BG pin which is connected to the gate of the N-channel, has voltage swing between 0 to 5V, so out Vgs = 5V. The average current through the MOSFET is 1.3A.
Also, the TG pin which is connected to the gate of the P-channel has the same voltage swing, so our Vgs = 5V. The current through the P-channel is 2A.

To obtain the peak current of the inductor, we use the following equation.

$$I_{O(MAX)} = (I_{PK} - 0.5\Delta I)(1 - DC)$$

Where: I_{PK} = Peak Inductor Current | ΔI = Inductor Ripple Current | DC = Duty Cycle
then I_{PK} = 2.73A (4.2V), 3.2A (3.7V)
Thus, we will need an inductor that does not saturate at 3.2A.

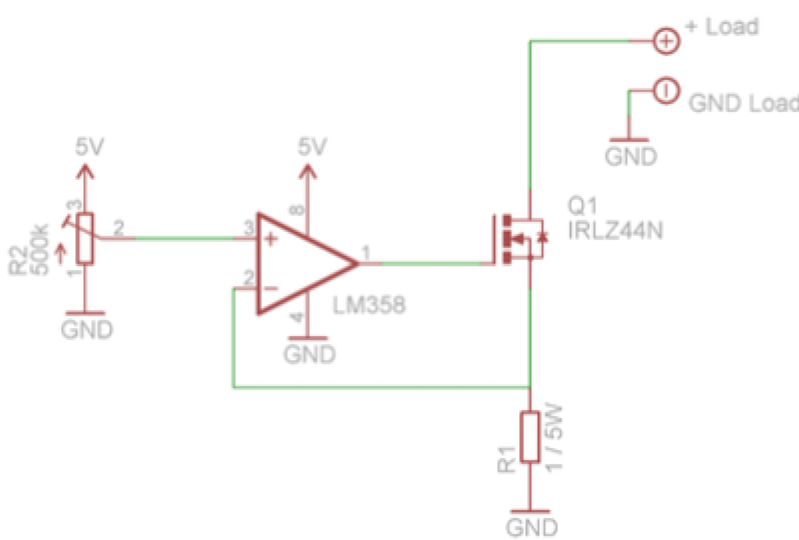


Figure 6. Constant Current Load Circuit

Requirement	Verification
1. Li-Ion battery cells charges to 3.7 V 2. Check if the temperature of the charger can be kept below 125℃ during the charging cycle. 3. The total capacity of connected battery cells store 2.6 A * # of battery cells (4). 4. Provided output voltage is 5V +/- 10%. 5. Output current output ranges from 0 to 2A. 6. Check the temperature of voltage booster is kept below 125℃	1. A. Discharge the battery B. Charge the battery at the output of the MCP73833T. C. As the current termination point is reached, check if the battery is fully charged. 2. A. Throughout the charging cycle, check the temperature with the thermometer. 3. A. Fully charge the battery and connect to the constant-current load circuit (Figure. 6). B. Discharge at 2000mA for 6.5 hours. C. Measure voltage of the battery to check it remains above 3.0 V 4,5. A. Connect output to the load resistance that will draw 2000 mA B. Measure using an oscilloscope, check the output voltage is 5V and current is 3A. C. Repeat steps A and B using different loads. 6. A. During the verification of requirement 4 and 5, check the IC stays below 125℃ using a thermometer.

2.3 Control Module

2.3.1 Raspberry Pi

The Raspberry Pi is required as a connection between different units: power unit, keyboard unit, display unit and speaker unit.

2.3.2 SD card

The SD card will help us to store the API or code for the text-to-speech (TTS) module and speech-to-text (STT) module, which will finally be performed by Raspberry Pi.

2.3.3 Power button

The Power button is required to turn on or off the whole device at any time, which can help us save battery life.

2.3.4 LED indicator

The LED indicator should alert the user of the battery status whether it is charging or not, fully charged or not, and the device is powered on or off.

Requirement	Verification
1. The Raspberry Pi is required to successfully connect and transfer signals between different units 2. The Raspberry Pi can successfully perform speech-to-text module and text-to-speech module 3. The button is inputted correctly to the device without bouncing 4. The LED indicator needs to be able to clearly show three different statuses for the battery	1. A. The display unit can display what we have typed on the stenographic keyboard. B. The speaker can play MP3 files stored in Raspberry Pi within 2 seconds after we open the file. C. The Raspberry Pi can store sounds recorded by the microphone with no more than 2 seconds delay. 2. A. When we run the code stored in Raspberry Pi on a computer, texts will be played with no more than 2 seconds delay. B. By connecting Raspberry Pi and a computer, the computer can output sounds after we type a sentence in the terminal with no more than 2 seconds delay. 3. The device can successfully turn on or off with one press of the button 4. With three different colors of LEDs: red, green and blue implemented in different circuits, each LED can correspond to a certain status: red

	(power lower than 10%), green (power greater than 10%) and blue (device is recharging)

2.4 Display Unit

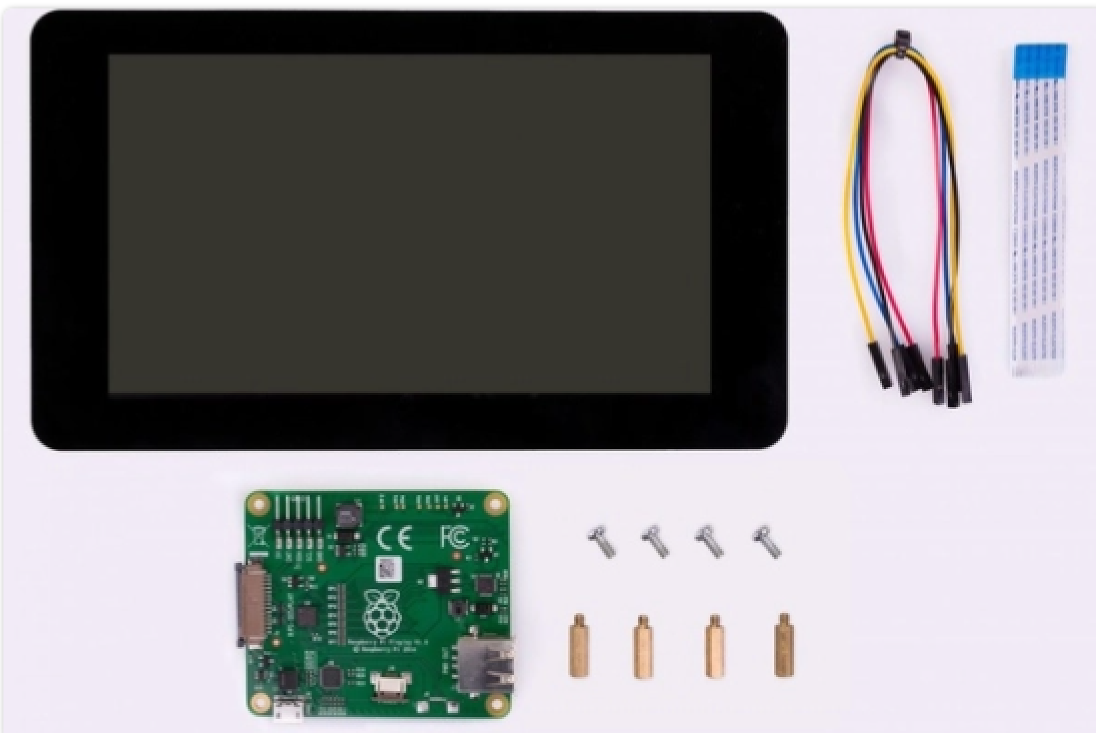


Figure 7: Display Unit Configuration

2.4.1 LCD Panel

The LCD Panel will help us display messages typed by users using keyboard and texts transferred from speech recorded by our microphone . Considering voltage and current requirements, we decided to use a 5" graphic LCD display.

2.4.2 Driver Board

The Driver Board can not only play the role of a connection between the raspberry pi and the IPS panel, but also help us to recharge the display unit.

Requirements	Verifications
1. The LCD Panel is required to clearly show texts typed by the keyboard 2. The LCD Panel is also required to clearly show texts transferred from speech. 3. The display can successfully be charged by the battery through the Raspberry Pi	1. Each word will be displayed on the display unit within 2 seconds after you type the word using the stenographic keyboard. 2. Sentences spoken by an user will be displayed word by word on the display unit within 2 seconds. 3. After the battery runs out, the display unit will reboot within 5 minutes after we recharge the battery.

2.5 Speaker Unit

A simple USB or 3.5mm jack input speaker will be connected to the Raspberry Pi. The power will be provided via USB. The speaker needs to clearly play the mp3 file created on Raspberry pi via TTS library. The volume of the speaker should be loud enough to be heard from 6ft away following the covid social distancing guidelines.

Requirements	Verifications
1. The speaker unit should be able to play an MP3 file created and saved in the SD card via connection to the raspberry pi board. 2. The physical dimension of the speaker should not be too large to fit on the belt but it should be able to produce audio loud and clear enough to be heard at talking distances.	1.For both STT and TTS modules, we don't need to store all the speech. We just need to convert the current sentence of speech or text, which should be no more than 30 seconds. A minute of the MP3 file populates the storage at about 1.5MB. Thus, the SD card should have enough storage. We will verify this by constantly typing and creating mp3 files however long the average conversation lasts which is around 30 mins. 2. The SNR for this speaker is 60db, which is a normal voice level at distances ranging 1 to 4 meters. We will verify that the speaker is audible by having the interactor talk and listen at 6ft away from the user wearing the belt.

2.6 Microphone Unit

A USB microphone will be used for our device, which should provide inputs for our speech to text module. In order to have a high accuracy rate for our module, we select a microphone that has a relatively high SNR value representing signal-to-noise rate, which can be calculated by the following equation.

$$SNR = \frac{Power_{signal}}{Power_{noise}}$$

Requirements	Verifications
1. The range of the microphone is long enough to pick up the interactor's sound. 2. The sound signal quality will be good enough to be used in the STT module.	1. A common distance during conversation is around 1 to 3 meters. Therefore, our microphone is required to record sounds within 3 meters. 2. After integrating the microphone with the STT module, there should be no output while nobody is talking.

2.7 Keyboard Unit

2.7.1 Stenographic keyboard

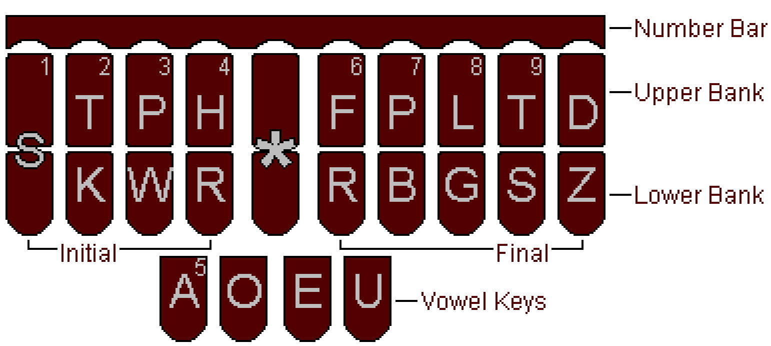


Figure 6. Stenographic keyboard configuration example

This is the typical configuration for a stenographic keyboard.

2.7.2 Microcontroller

Takes in the switch inputs and outputs it in a usable USB signal.

Requirements	Verifications
1. This keyboard is required to be able to type any word based on the steno alphabet 2. The stenographic keyboard can be successfully typed with a speed of normal communication	1. A. The display can show every word user types within 2 seconds. B. When users press more than 1 key at the same time, the keyboard should be able display all of them before we integrate the keyboard with the steno api. According to the steno alphabet, this function should work up to 7 keys. 2. Users are able to type more than 30 words per minute when they are familiar with how to use a stenographic machine.

2.8 Software Unit

For our device, the software unit can be divided into three parts: STT module, TTS module and stenographic api. Those three modules will all be implemented using python code. Besides, In order to have a wearable device, we will use an SD card to store all python codes and corresponding libraries.

2.8.1 Speech-to-Text Module

For the offline speech to text task, we will use a speech recognition toolkit, Vosk, which is able to work offline on the Raspberry Pi.This module is able to convert speech to text for both recorded and real-time conversation.

2.8.2 Text-to-Speech Module

In order to achieve the text to speech task for our device, we decided to implement pyttsx3, which is a text-to-speech conversion library in Python. Different from other libraries in Python, this library is able to work offline and compatible with both Python 2 and 3.By installing this library, we can easily change the voice, rate and volume by changing certain variables.

2.8.3 Plover

We have chosen Plover as our stenographic api, which can work offline. After installing Plover in our Raspberry Pi, our keyboard will be able to have the same functionality as a stenographic machine.

<table>
<tr><th>Requirements</th><th>Verifications</th></tr>
<tr><td>Both modules are required to work without the internet, and they should be easily turned on and off.
Those two modules will not influence each other’s performance.</td><td><ol type="A">
Those two modules will start working after we press the power button with no more than 2 seconds delay.
Both modules should be able to work continuously for more than one hour
One module is required to be turned off when the other module is currently used, and the TTS module should have a higher priority, because the TTS module displays the thoughts of our users.</td></tr>
</table>

3 Circuit Schematics

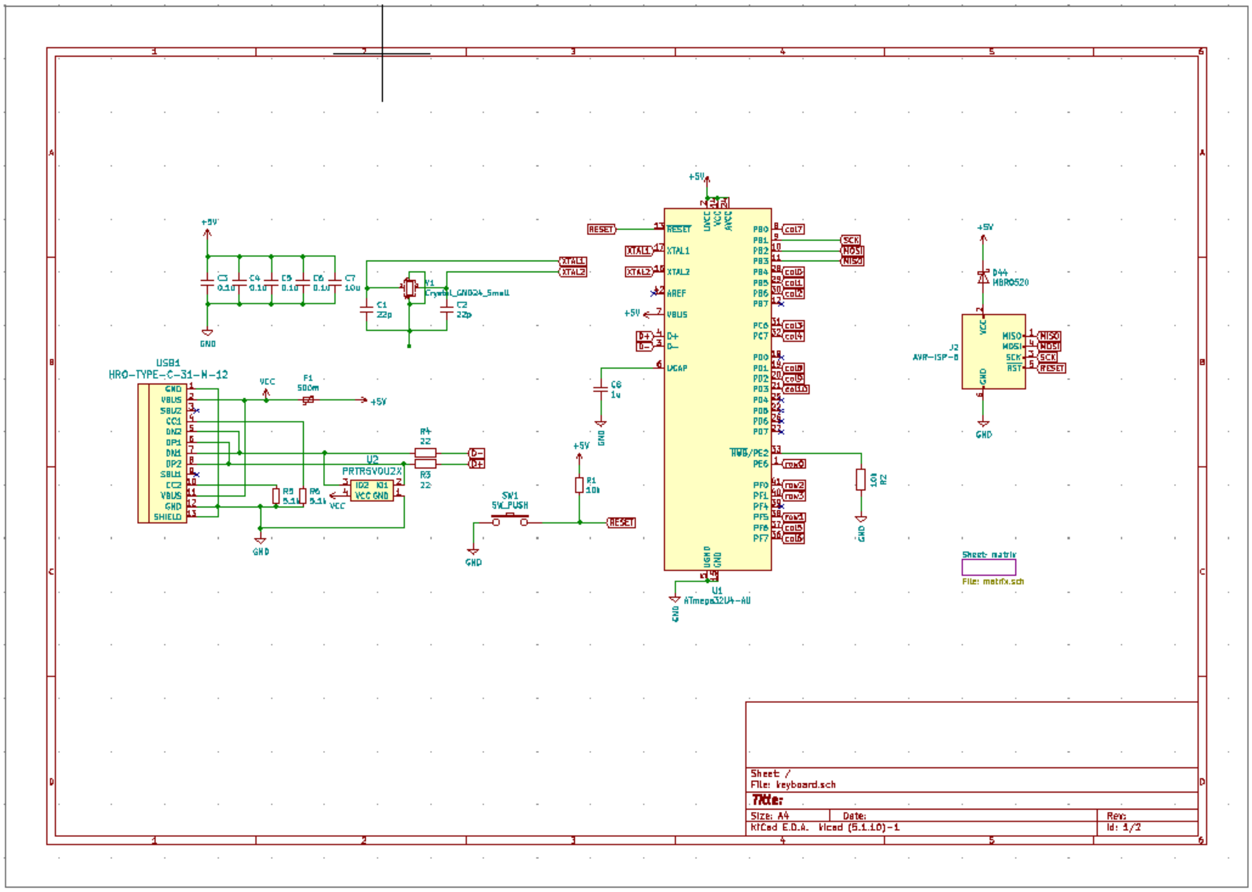
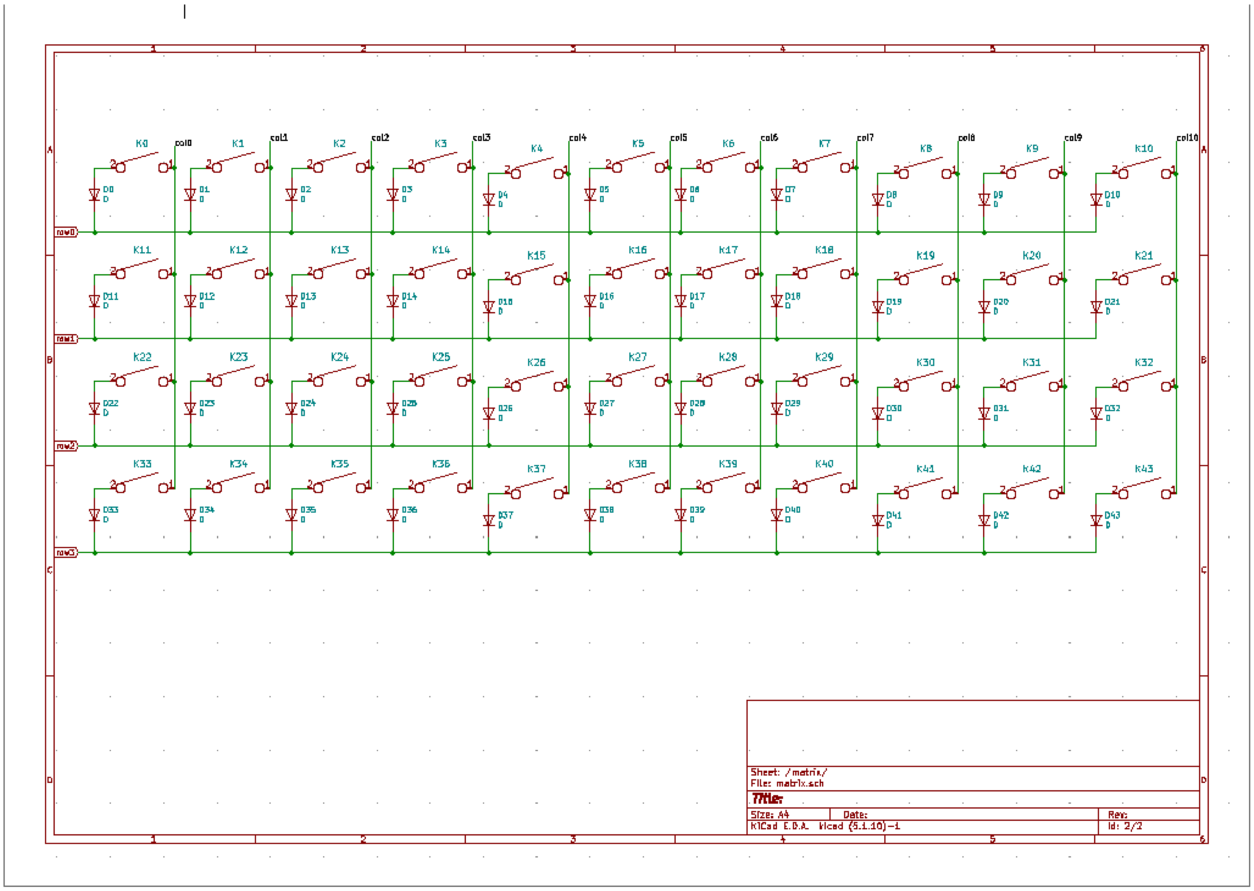


Figure 8. Keyboard schematic part 1



Having the switches connected to diodes prevents the keyboard from having ghost keypresses.

Figure 9. Keyboard schematic part 2

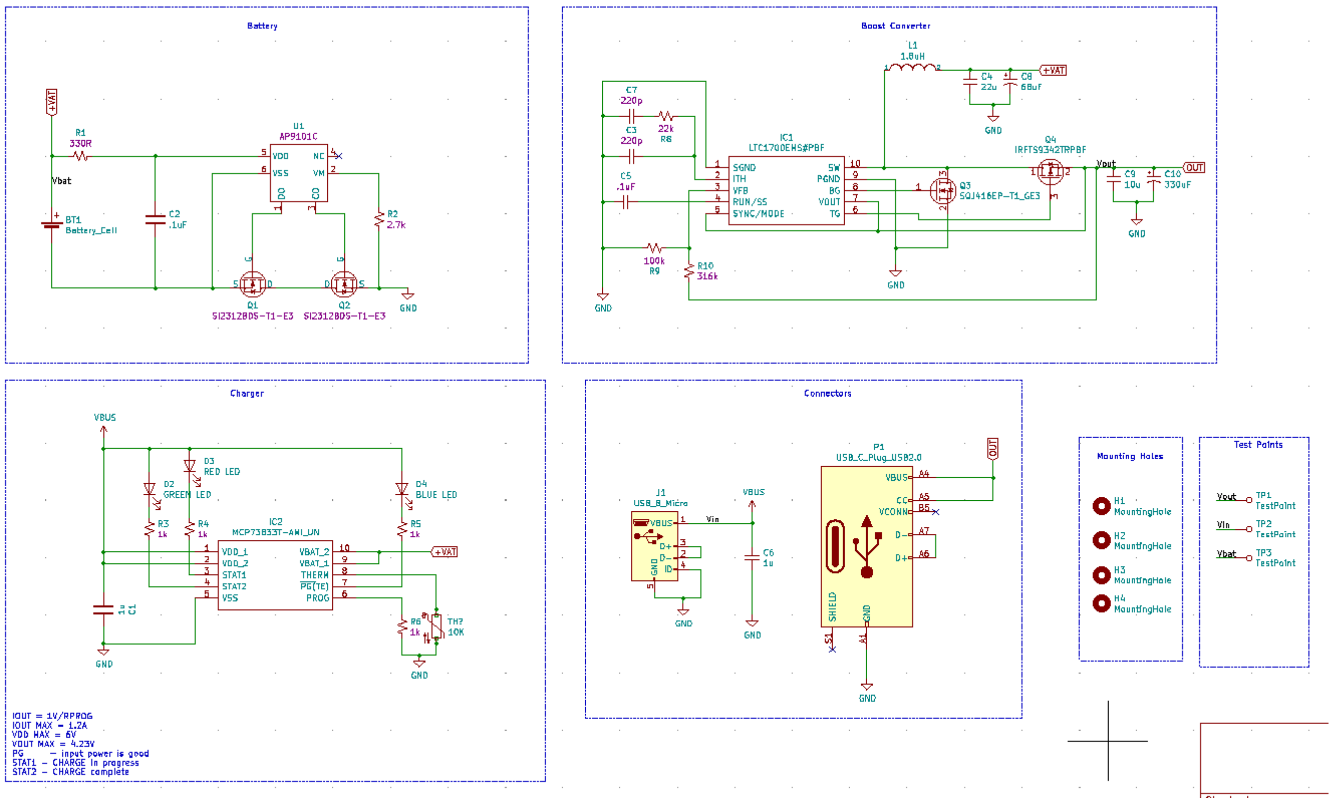


Figure 10. Power bank schematic

4 Tolerance Analysis

The most critical risk associated with our project comes from the accuracy of the microphone, whether it is able to receive the interactors' speech from the background noise especially when our design is for use outdoors.

Consequently, we have to effectively calculate the ratio of background noise to the speech signal that we want our system to convert into displayable text. In order to have our software filter out the noise and only "listen" to the words in need of conversion, the following mathematical formula for determining the ratio between the two may be important.

The SNR (Signal to Noise Ratio) expressed in decibels is obtained by:

$$SNR_{dB} = 20log_{10}(\frac{S_{rms}}{N_{rms}})$$

Where S_rms represents the root mean square of the speech signal without any noise present,

$$S_{rms} = \sqrt{\frac{1}{N}(s_1^2 + s_2^2 + s_3^3 + ... + s_N^2)}$$

and N_rms is the root mean square level of the noise without speech.

$$N_{rms} = \sqrt{\frac{1}{N}(n_1^2 + n_2^2 + n_3^3 + ... + n_N^2)}$$

The SNR formula is equivalent to

$$SNR_{dB} = 10log(\frac{Signal\,Energy}{Noise\,Energy})$$

The challenging part of this calculation is to derive the signal and noise energies without any prior knowledge about the data in the audio file created. The typical speech signal, such as the one shown below.

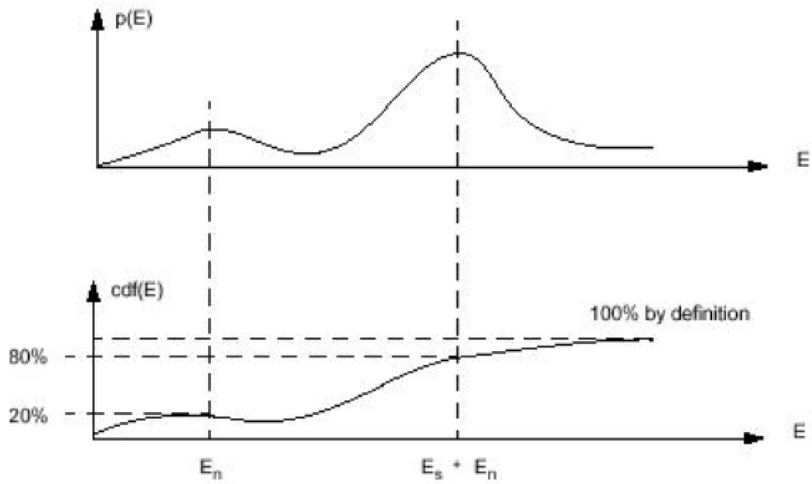


Figure 10. An energy probability density function and the corresponding cumulative distribution function

These two graphs relate to the nominal noise energy and nominal signal plus noise energy respectively. From the cdf above, we can design the thresholds which select the percentage of data points which we would expect to correspond to the signal + noise energy and the noise energy. Typically the thresholds of 80% signal and 20% noise or (80/20 and 95/15) are popular choices and these numbers have been derived by experienced speech researchers based on much research and analyses of various types of data.

With the method above, we can then define the estimated SNR based on the energy levels that meets our thresholds, i.e

$$SNR_{dB} = 10log_{10}(\frac{(E_s + E_n) - E_n}{E_n})$$

5 Costs

5.1.1 Parts

Description	Quantity	Manufacturer	Vendor	Cost/unit	Total cost (Bulk)
Keyboard microcontroller ATMEGA32U4	1	Microchip Technology	Digikey	$5.16	$5.16
1N4148W-TP Diodes (keyboard)	50	MCC	LCSC	$0.0135	$0.68
PTC FUSE nSMD025-24V (keyboard)	1	TECHFUSE	LCSC	$0.047	$0.047
LFXTAL082071Reel Crystal (keyboard)	1	IQD Frequency Products	Digikey	$0.74	$0.74
22pF Capacitors (Keyboard)	2	Samsung Electro-Mechanic s	Digikey	$0.10	$0.20
1uF Capacitors (Keyboard)	1	Samsung Electro-Mechanic s	Digikey	$0.10	$0.20
0.1uF Capacitors (Keyboard)	4	Samsung Electro-Mechanic s	Digikey	$0.10	$0.40
10 uF Capacitors (Keyboard)	1	Samsung Electro-Mechanic s	Digikey	$0.29	$0.29
22 Ohm resistors (Keyboard)	2	Vishay Dale	Digikey	$0.10	$0.20
10k Ohm resistors (Keyboard)	2	YAGEO	Digikey	$0.10	$0.20
5.1k Ohm resistors	2	YAGEO	Digikey	$0.10	$0.20

PRTR5V0U2X	1	Nexperia	LCSC	$0.26	$0.26
RESET Button (Keyboard)	1	C&K	Digikey	$0.56	$0.56
USB C connector (Keyboard)	1	Korean Hroparts Elec	LCSC	$0.3156	$0.3156
Akko CS Switches (45pcs) (keyboard)	1	Akko	Amazon	$13.99	$13.99
Raspberry Pi 4B	1	Raspberry Pi	VILROS	$55	$55
Raspberry Pi Foundation 7" Touchscreen LCD Display	1	Raspberry Pi	CanaKit	$60	$60
USB Mini Stereo Speaker	1	CanaKit	CanaKit	$14.95	$14.95
Dreokee USB Conference Microphone	1	Dreokee	Dreokee	$10.99	$10.99
Silicon Power 32GB 3D NAND High Speed MicroSD Card with Adapter	1	Silicon Power	Silicon Power	$7.99	$7.99
UL1865-26-1P 3.7V 2.6 Ah Li-Ion	4	Ultralast	Digikey	$4.99	$19.96
BK-18650-PC8 4-Cell Battery Holder	1	MPD (Memory Protection Devices)	Digikey	$7.85	$7.85
AP9101CK6-ADTRG1 Battery Protection IC	1	Diodes Incorporated	Digikey	$0.48	$0.48
824500500 Zener Diode	1	Würth Elektronik	Digikey	$0.28	$0.28
SI2312BDS-T1-E3 NMOS	2	Vishay Siliconix	Digikey	$0.54	$1.08

0ZCG0200FF2C PTC Fuse	1	Bel Fuse Inc.	Digikey	$0.19	$0.19
0402YC104KAT2A .1uF Capacitors	2	AVX Corporation	Digikey	$0.10	$0.20
AC1206FR-07330RL 330 Ohm Resistors	1	YAGEO	Digikey	$0.10	$0.10
ERJ-8GEYJ272V 2.7k Ohm Resistors	1	Panasonic Electronic Components	Digikey	$0.10	$0.10
MCP73833T-AMI/UN Charger IC	1	Microchip Technology	Digikey	$1.04	$1.04
SML-512UWT86 Red LED	1	Rohm Semiconductor	Digikey	$0.61	$0.61
SMLD12BN1WT86 Blue LED	1	Rohm Semiconductor	Digikey	$0.73	$0.73
SML-D12P8WT86 Green LED	1	Rohm Semiconductor	Digikey	$0.33	$0.33
RE0603FRE071KL 1k Ohm Resistors	3	YAGEO	Digikey	$0.10	$0.10
RMCF0201FT10K0 10k Ohm Resistors	2	Stackpole Electronics Inc	Digikey	$0.10	$0.10
LMK105C6105MV-F 1uF Capacitors	2	Taiyo Yuden	Digikey	$0.10	$0.20
LTC1700EMS#PBF Voltage Booster IC	1	Analog Devices Inc.	Digikey	$7.16	$7.16
AIML-0603-1R8K-T 1.8uH Inductor	1	Abracon LLC	Digikey	$0.10	$0.10
865080142006 68uF Polarized Capacitor	1	Würth Elektronik	Digikey	$0.20	$0.20
865080145010 330uF Polarized Capacitor	1	Würth Elektronik	Digikey	$0.34	$0.34
CL05C221JB5NNNC	2	Samsung	Digikey	$0.10	$0.20

220pF Capacitor		Electro-Mechanics			
CL21A226MQQNNNE 22uF Capacitor	1	Samsung Electro-Mechanics	Digikey	$0.15	$0.15
CL21A106KOQNNNG 10uF Capacitor	1	Samsung Electro-Mechanics	Digikey	$0.11	$0.11
CRCW060322K0JNEAC 22k Ohm Resistor	1	Vishay Dale	Digikey	$0.10	$0.10
CRCW0402100KJNEDC 100k Ohm Resistor	1	Vishay Dale	Digikey	$0.10	$0.10
CRCW0603316KFKEB 316k Ohm Resistor	1	Vishay Dale	Digikey	$0.10	$0.10
SQJ416EP-T1_GE3 NMOS	1	Vishay Siliconix	Digikey	$0.93	$0.93
IRFTS9342TRPBF PMOS	1	Infineon Technologies	Digikey	$0.58	$0.58
PCBs	2	pcbway	pcbway	$5	$10
Total				$204.07	$229.82

Table 1: Costs for physical parts

5.1.2 Labor

Name	**Hourly Rate**	**Hours**	**Total/Person**	**Total x 2.50**
Andrew Ko	$30	160	$4800	$12000
Minho Lee	$30	160	$4800	$12000
Yihan Ruan	$30	160	$4800	$12000
Total Cost for the Team				**$36000**

Table 2: Costs for labor

5.1.3 Grand Total

Section	**Cost**

Parts	$229.82
Labor	$14,400
Grand Total	$14,629.82

Table 3: Grand total cost

6 Schedule

Week	Minho	Andrew	Yihan
9/20/21	Work on Schematics for a power bank.	Finish keyboard pcb design	Doing research on TTS/STT Api
9/27/21	Finish the schematic and Order parts (Pi and IC)	Get it reviewed and make changes accordingly	Doing research on stenographic keyboard Api
10/4/21	Design enclosure for the power bank. Consult on version 2 PCB	Design enclosure for the display module. Consult on version 2 PCB	Make sure STT module can successfully work on the Raspberry Pi
10/11/21	Test and verify the PCB	Test and verify the PCB	Make sure TTS module can successfully work on the Raspberry Pi
10/18/21	Version 2 PCB design	Version 2 PCB design	Writing code for eliminating noises during STT module, which should increase the accuracy rate
10/25/21	Finish and Order version 2 of PCBs	Finish and Order version 2 of PCBs	Writing code for eliminating noises during STT module, which should increase the accuracy rate
11/1/21	Collect data for power consumption and safety of the battery.	Work on integrating the keyboard and the Pi	Apply stenographer keyboard api on the Raspberry Pi
11/8/21	Test PCBs and verify they pass all requirements.	Scripts and code writing to run the main loop of the system	Work on integrating keyboard, display and Pi
11/15/21	Design enclosure and assemble the modules onto the belt	Testing to ensure the functionality of keyboard to play words as mp3 and mic to to display words on screen	Ensure STT and TTS modules can successfully work with display and keyboard
11/22/21	Final assembly of the	Work on the enclosures	Write final report

	project.	and have the entire system be wearable for the users	
11/29/21	Write final report	Write final report	Write final report
12/06/21	Prepare final presentation	Prepare final presentation	Prepare final presentation

Table 4: Schedule table

7 Ethics and Safety

There can potentially be some safety hazards with our project as it contains batteries. The batteries should not be overcharged and handled with care as they might explode with rough use [2]. The battery may get damaged if the temperature is too high or too low. Therefore, we will add a thermistor and a protection IC to the charging loop to disconnect the battery from the charger when abnormality in temperature is detected. Also, the charger for the lithium battery will include an IC as suggested in ECE445 Battery Safety [3]. Physical abuse of the battery may also damage the battery, so the belt device might have its limits when the user plays sports or participates in vigorous physical activities [4].

Designed as an outdoor use device, the belt might be prone to some disastrous weather conditions, if there is a heavy pour of rain, for example. Since we are building the display module as an extendable unit, we should be careful in enclosures and circuit designs such that the users will not get electrocuted or get physically wounded by any of the device components.

Since we will be using open library sources for text to speech and speech to text in the translation process, we should make sure to cite the sources as to not infringe on one of IEEE code of Ethics - to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others.

8 Citations/References

[1] J. Meyer, L. Dentel and F. Mecunier, "Speech Recognition in Natural Background Noise," *PLOS ONE*, November 19, 2013. [online], Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0079279 [Accessed Sep. 28, 2021]

[2] *Preventing fire and/or explosion injury from small and ...* (n.d.). Retrieved September 29, 2021, from https://www.osha.gov/sites/default/files/publications/shib011819.pdf

[3] *General Battery Safety*. Project :: ECE 445 - Senior Design Laboratory. Retrieved September 29, 2021, from https://courses.physics.illinois.edu/ece445/project.asp?id=6058

[4] *Battery regulation*. U.S. Consumer Product Safety Commission. (n.d.). Retrieved September 29, 2021, from https://www.cpsc.gov/Regulations-Laws--Standards/Voluntary-Standards/Topics/Batteries.

https://www.isip.piconepress.com/projects/speech/software/legacy/signal_to_noise/

[5] *Raspberry pi typical power requirements*. Raspberry Pi Hardware. (n.d.). Retrieved September 30, 2021, from https://www.raspberrypi.org/documentation/computers/raspberry-pi.html#typical-power-requirements