**Deliverable H: Video and User Manual**

**GNG 2101– Introduction to project management and development**

Group Z25

**Authors:**  
José Barragán | 300103915 | [jbarr025@uottawa.ca](mailto:jbarr025@uottawa.ca)

Zarvaan Bacha | 300128068 | zbach073@uottawa.ca

Gavin Gordon | 300299729 | [ggord034@uottawa.ca](mailto:ggord034@uottawa.ca)

Zeyad Abu-El-Quran | 300167089| [zabue094@uottawa.ca](mailto:zabue094@uottawa.ca)

Prof. Emmanuel Bouendeu

Deliverable due July 23th, 2023

[1 Introduction 3](#_Toc141044775)

[2 Overview 4](#_Toc141044776)

[2.1 Cautions & Warnings 4](#_Toc141044777)

[3 Getting started 5](#_Toc141044778)

[3.1 Configuration Considerations 6](#_Toc141044779)

[3.2 User Access Considerations 6](#_Toc141044780)

[3.3 Exiting the System 6](#_Toc141044781)

[4 Using the System 7](#_Toc141044782)

[4.1 Voice Controls 7](#_Toc141044783)

[4.1.1 Basic functions: 7](#_Toc141044784)

[4.1.2 Presets 7](#_Toc141044785)

[4.2 Touch Controls 7](#_Toc141044786)

[5 Troubleshooting & Support 8](#_Toc141044787)

[5.1 Troubleshooting: 8](#_Toc141044788)

[5.1.1 System not booted correctly: 8](#_Toc141044789)

[5.1.2 Faulty electricity: 8](#_Toc141044790)

[5.1.3 Faulty Microphone 8](#_Toc141044791)

[5.1.4 Faulty 6 pin connector 9](#_Toc141044792)

[5.1.5 Faulty buttons 9](#_Toc141044793)

[5.1.6 Voice Recognition issue 9](#_Toc141044794)

[5.2 Special Considerations 9](#_Toc141044795)

[5.3 Maintenance 9](#_Toc141044796)

[5.4 Support 9](#_Toc141044797)

[6 Product Documentation 10](#_Toc141044798)

[6.1.1 BOM (Bill of Materials) 10](#_Toc141044799)

[6.1.2 Equipment list 10](#_Toc141044800)

[6.1.3 Instructions 11](#_Toc141044801)

[6.2 Testing & Validation 16](#_Toc141044802)

[6.2.1 Physical Testing 16](#_Toc141044803)

[6.2.2 Performance Testing 16](#_Toc141044804)

[6.2.3 Results 16](#_Toc141044805)

[7 Conclusions and Recommendations for Future Work 17](#_Toc141044808)

[APPENDICES 18](#_Toc141044809)

[8 APPENDIX I: Design Files 18](#_Toc141044810)

# Introduction

This used and product manual provides all the necessary information to create a voice activated controller for a reclining chair utilising a 6 pin controller. It was built by using an mini computer, software packages, and a 3D printed structure in order to be able to interpret voices and transform them into chair movements.

This document will first explain how to use the product, any warnings or precautions that should be taken when using it, any troubleshooting that may be necessary, how and what is necessary to build it, as well as any other additional information necessary.

# Overview

Over half a million Canadians are classified as homebound. Over 80% of those suffer from some form of motor skill impairment. In our particular case, a patient suffering from ALS wanted to be able to control his reclining chair by using his voice instead of his hands.

ALS affects motor skills as well as speech, so an adaptable solution is key to be able to understand his voice or any sound regardless of how much deterioration it suffers.

Our product requires no internet or any other additional resources other than being plugged into the wall. It’s a standalone, plug and play solution developed from the ground up by us. It is cheaper to buld and sell than any other prduct on the market, and it performs better as well.

A white box with buttons on it

Description automatically generated

The system consists of a microphone, a mini computer, and a plastic case in order to interpret voices and operate reclining chairs.



## Cautions & Warnings

This work may not be used for commercial purposes. Creators are not responsible for any bodily harm that occurs from the use, build or test of the product.

# Getting started

Usage of the product is very simple thanks to the code it uses, to get it started:

1. Connect power box to the wall, and usb-c to the power box
2. Connect usb-c cable to remote control (blue circle)
3. Connect microphone to remote (red circle)
4. Connect remote to chair (6 pin cable)
5. Wait 3 minutes for the system to boot up.
6. Once the device is ready, a loud click will be heard.
7. Use voice commands to move chair: See more details in “Using the system” section

A white rectangular object with buttons and a cord

Description automatically generated

The remote was built with simplicity in mind. It uses a script in order to launch the voice recognition program in the computer. This results in the device being completely plug and play. The most important part is making sure that it’s all connected correctly to the electricity, to the wall, and to the chair; and waiting for activation. .

## Configuration Considerations

There are no configurations to quickly adjust anything, the voice model is robust enough to not require any adjustments.

## User Access Considerations

Any user with a voice will be able to use the system. The only problem that may arise is that if the user is an ALS patient and their voice deteriorates, they might need to contact the developer in order to be able to adjust the system to their worsening voice.

## Exiting the System

To exit the system, you need to disconnect every component from the remote: That is electricity, microphone and the chair. No damage will occur on disconnection. Its important to note that it should not be turned off during an execution of a command.

# Using the System

The following sub-sections provide detailed, step-by-step instructions on how to use the various functions or features of remote. The remote controls 2 different axis’ of movement for the chair, the headrest and the inclination.

## Voice Controls

The whole point of the voice controlled remote is to be able to use voice to control it. In addition to accomplishing the main objective of the project, using the voice allows for additional functionalities that are not accessible with the normal buttons, and help improve the quality of life of the user.

### Basic functions:

To control the headrest and the position of the chair, one can use each of the following commands, once one of them is called, the software will imitate the click of a button for a duration of 5 seconds.

* Chair up: Will move the inclination of the chair backwards.
* Chair down: Will move the inclination of the chair forward.
* Headrest up: Will lift the headrest.
* Headrest down: Will lower the headrest.

### Presets

The main advantage of using the voice to control the chair, in addition to not being dependent on limbs that may suffer from fine motor control issues, is the fact that presets can be accessed to go to a pre-determined position, the following can be used but more can be trained depending on the needs and wants of each client.

* TV: Headrest fully up, Inclination all the way back
* Sit: Default position
* Stand: Inclination all the way forward, headrest in the retracted position
* Sleep: Inclination all the way back, headrest in the retracted position

## Touch Controls

Additionally, 4 buttons exist on the remote to be able to access the functions if for some reason the voice module becomes unavailable or breaks, they perform the “Basic Functions”.

A white box with buttons on it

Description automatically generated

# Troubleshooting & Support

Given that the remote is a very complex device, which has very simple function, it means that a lot can go wrong, however it will all show as the same behavior: the chair won’t move. Read below how to correctly diagnose and correct any possible issues, and who to ask for help in case the problem can’t be fixed.

## Troubleshooting:

The different points in this section will show whaqt could have gone wrong, and what can be done to test and fix:

### System not booted correctly:

* When: After connecting and waiting for 3 minutes, if pushing buttons and saying commands wont make the chair move
* What to do: Wait 2 more minutes, a click should come from the remote signaling its ready to use. Then try pushing on the buttons. The buttons should emit a loud click when actioned and the chair should move. At this point its ready to work
* If problem persists: Disconnect from electricity, microphone, and chair. Reconnect and wait
* If problem still persists: Contact support (see next section)

### Faulty electricity:

* When: After connecting and waiting for 3 minutes, if pushing buttons and saying commands won’t make the chair move
* What to do: Look at the side of the box next to the wire connections. A light red glow should be seen through the plastic casing. If no light or sounds is being emitted from the remote, try using a different usb-c cable or usb power box.
* If problem still persists: Contact support (see next section)

### Faulty Microphone

* When: After connecting and waiting for 3 minutes, if saying commands won’t make the chair move but pushing buttons will
* What to do: Disconnect and reconnect the usb wire from the remote and the microphone, then reconnect.
* If problem persists: Order a new microphone or test with a computer.
* If problem still persists: Contact support (see next section)

### Faulty 6 pin connector

* When: After connecting and waiting for 3 minutes, if pushing buttons and saying commands wont make the chair move, but a clicking noise if heard after saying the command or pushing the buttons
* What to do: Disconnect and reconnect 6 pin wire to chair, verify integrity of wire
* If problem persists: Use factory remote with chair to see if that will work, verify electrical connection of chair to the wall.
* If problem still persists: Contact support to see about a replacement in 6 pin wire

### Faulty buttons

* When: After connecting and waiting for 3 minutes, if saying commands and pushing some buttons result in a clicking sound from the remote, but other buttons result in no clicking sound or if chair movement is erratic
* What to do: Disconnect remote from the chair. It may be unsafe to use chair if button activates continuously. Contact support for button replacement.

### Voice Recognition issue

* When: After connecting and waiting for 3 minutes, if saying commands and pushing buttons will make the chair move, but some commands won’t work.
* What to do: Try to pronounce word differently, move the microphone to a quitter place or closer to the mouth
* If problem persists: Contact support to update the voice model to match worsening or different voice.

## Special Considerations

* If the remote at any point emits a burning smell of smoke, use fire extinguisher, and disconnect from electricity.
* If the remote starts commanding erratic movements from the chair, disconnect and contact support immediately. Cease use since this may result in bodily harm.

## Maintenance

No maintenance is necessary, but support may be contacted if new pre-sets or customizations are desired.

## Support

Please contact any member of team Z25, information available via the maker repo link in the appendix, for help with the product, troubleshooting, or general questions. **Unless it’s an emergency, in which case contact the emergency services in your area.**

# Product Documentation

### BOM (Bill of Materials)

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Quantity / Notes** | **Cost (CAD)** | **Link** |
| Wire | 6 | 0.4 | [Ali Express](https://www.aliexpress.com/item/32668248691.html?spm=a2g0n.productlist.0.0.692632f9DciSmc&browser_id=c24db530d3ff49f58d4a5bbf0a181a8a&aff_platform=msite&m_page_id=kbbtgwvamcasgmth18877ee7e761c79510e01d333a&gclid=&pdp_npi=3%40dis%21CAD%210.49%210.4%21%21%21%21%21%402102160416856392986862530d0745%2159996283032%21sea%21CA%210&algo_pvid=1ea7d232-a5e3-4123-a4d5-e6b3f1e3ff12) |
| Soldering Equipment | From maker space | 0 |  |
| Solder | From maker space | 0 |  |
| Buttons | 4 | 4 | Maker Space |
| PLA Filament | From maker space | 1 |  |
| 3D Printer | From maker space | 0 |  |
| Raspberry Pi 4B  2 GB | From maker space | 55 | Makerspace |
| SD Card | 32 GB | 7 | [Amazon](https://www.amazon.ca/ALERTSEAL-MicroSD-Memory-microSD-Camera/dp/B07F81VC69/ref=sr_1_7?crid=1F7S6NL0UDC0I&keywords=microsd+card&qid=1687817201&sprefix=microsd+card%2Caps%2C104&sr=8-7) |
| Mini USB Cable | Included with Pi | 0 |  |
| Microphone | From Amazon | 26 | [Amazon](https://www.amazon.ca/dp/B07K427F1S?psc=1&ref=ppx_yo2ov_dt_b_product_details) |
| Screwdrivers | From maker space | 0 |  |
| Fasteners (Velcro straps) | From maker space | 1 |  |
| 3D scanner | From maker space | 0 |  |
| Python | Version 3.7 | 0 |  |
| TensorFlow | Version 1.18 | 0 |  |
| Precise | Version 0.3 | 0 | [Github](https://github.com/MycroftAI/mycroft-precise) |
| Generic computer | For training model | 0 |  |
| SolidWorks | For case design | 0 |  |
| 4-Channel Relay | For remote interphase | 12 | [Amazon](https://www.amazon.ca/SainSmart-101-70-101-4-Channel-Relay-Module/dp/B0057OC5O8?th=1) |
| Standoffs | Various sizes | 16 | [Amazon](https://www.amazon.ca/Raspberry-Standoff-Standoffs-Building-Quadcopter/dp/B0788FR59P?th=1) |
| Heat Shrink |  | 10 | [Amazon](https://www.amazon.ca/Shrink-Tubing-Electric-Insulation-Industrial/dp/B0BTKSQ8NS/ref=sr_1_15?keywords=Heat+Shrink&qid=1687817505&sr=8-15) |
| Total |  | 132 |  |

### Equipment list

#### General Equipment

* 3D Printer for case
* Screwdrivers and pliers

#### Electronics

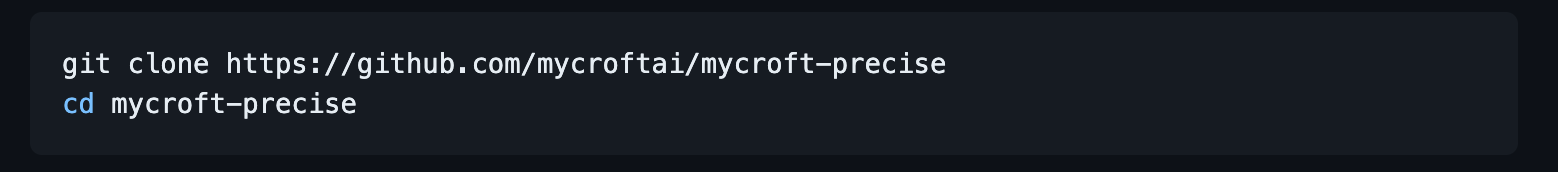
The following are some pre-requisite requirements needed for the electronics:

* Access to soldering tools, wires and a multimeter.
* Understanding of basic electronics and circuits
* Standard wiring from existing controller with plug.

### Instructions

1. Software Environment Setup

The following are some pre-requisite requirements needed for the product to function.

* A modern computer operating on Linux
* Linux Bash Command Line knowledge
* Understanding on how to code
* Knowledge and understanding of Python
* Microphone
* USB drive
  1. Downloading the Precise wake-word factory
  + Setup a folder on your computer and clone [this](https://github.com/MycroftAI/mycroft-precise) GitHub repository.
  + Run the setup script
  + Activate Virtual Environment
  + Precise now has an environment setup.

1. Creating you first wake word (Hey Computer)

2.1 Folder Setup

* Create a folder and a subset of folder with the following structure.
  + hey-computer
    - * test
        + wake-word
        + not-wake-word
      * wake-word
      * not-wake-word

2.2 Collecting Sample Data

* A computer screen with white text

  Description automatically generatedPrecise has a built-in sample collection tool which can be used to record your audio, enter the script in the terminal to start recording.
* For a successful wake-word, collect a minimum of 15 samples
* Now, place most of these files under hey-computer/wake-word/ and the rest under hey-computer/test/wake-word

2.3 Training your model

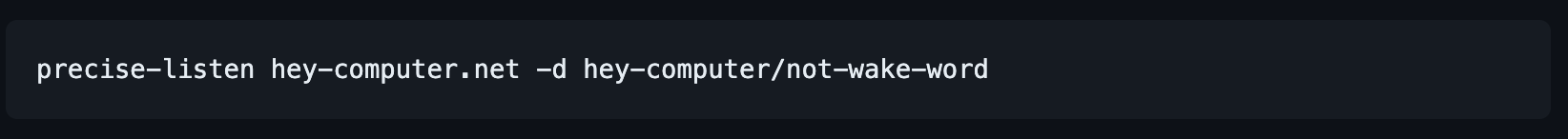
* A computer code with white text

  Description automatically generatedUse the precise-train script to start training your model on the data you have provided.
* A screenshot of a computer

  Description automatically generatedYou can run this model against live microphone input using precise-listen. It will listen to the microphone and output confidence bars. Each line represents one measurement: the more Xs there are, the more confident that the model believes that the wake word was uttered. Any Xs over the threshold are denoted with a lowercase x.

2.4 Reducing False Activations

* The quickest way to reduce false activations is by telling precise to collect any audio from the mic that activates the model. Launch precise-listen in save mode.



* + Now you can say words similar to your wake word and every time the model activates, it will save that recording into the hey-computer/not-wake-word folder. **Make sure never to say the actual wake word while in save mode.**
  + Once you've gathered a few samples of new false activations, retrain your model with the same precise-train command.
  + A black background with white text

    Description automatically generatedTest it again through the microphone with

2.5 Converting your first model

* A black and white text

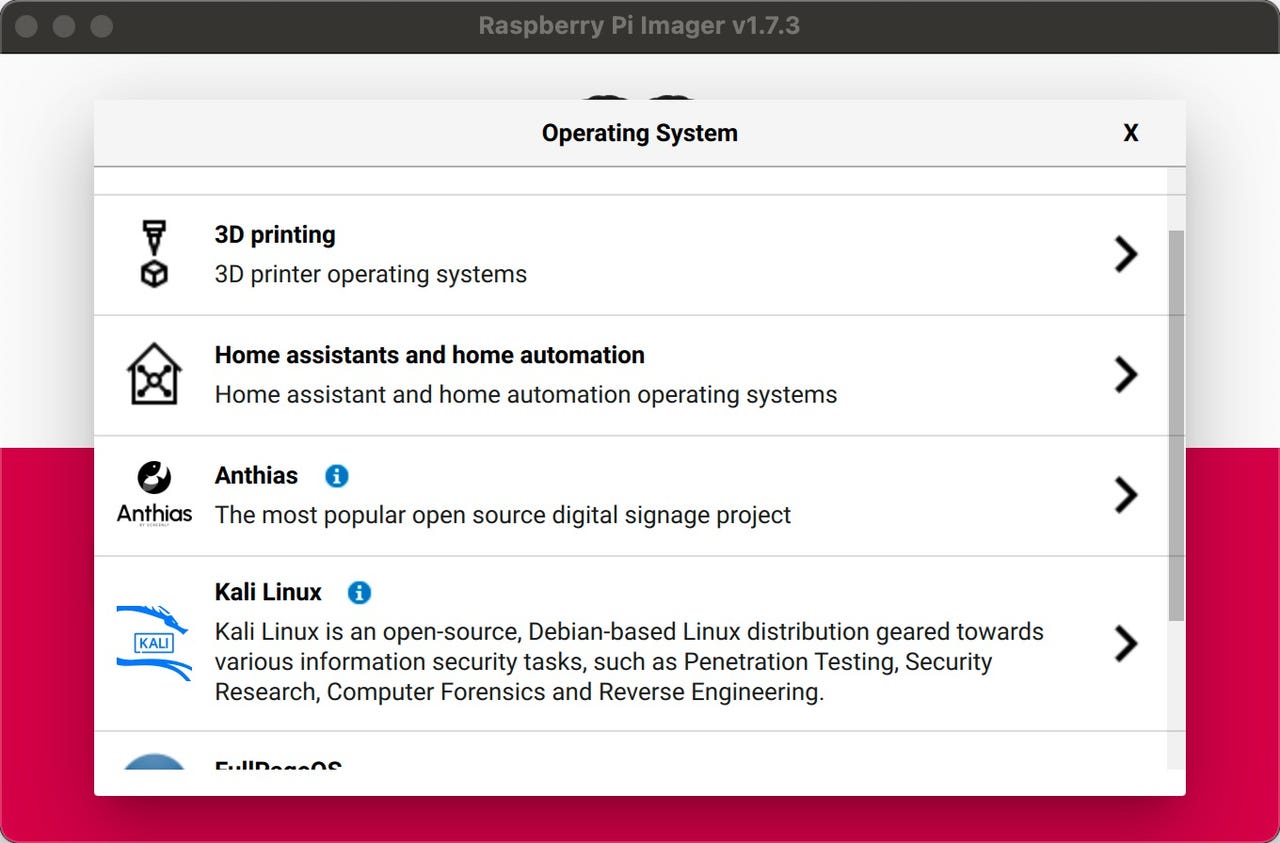
  Description automatically generatedOnce you are satisfied with the performance of your model, convert it so it can be run on other devices
* Once the process completes, you should see two more files in the working directory
  + hey-computer.pb
  + hey-computer.pb.params
* These two files are required later for your device to recognize sounds. Copy these files over to the USB

1. Raspberry Pi Setup

The following are some pre-requisite requirements needed to setup and operate the Raspberry Pi

* 16 GB micro SD Card
* Micro SD to USB adapter (or anything else that can connect SD to computer)
* Raspberry Pi Model 3 or greater
* Keyboard, Mouse and monitor with HDMI input
* USB drive

3.1 Downloading the Raspberry Pi Imager

* Download and install Raspberry Pi Imager from [here](https://www.raspberrypi.com/software/)
* Choose Raspberry Pi OS 64 Bit
* Once installed on your SD Card, place the SD card into the Raspberry Pi

3.2 Setting up the Raspberry Pi

* Connect the keyboard, mouse, and monitor to the Pi.
* Connect the power cable for the Pi
* Follow the instructions on the screen to complete the setup for the Pi
* Connect it to the internet either using WiFi or plug in an ethernet cable
* Create a new folder on the Pi for the project
* Inside the folder right click and open the terminal
* Type in the following code in the terminal line by line
* The following downloads and sets up the Binary required by the Pi to run Precise
* A black background with white text

  Description automatically generatedNext, in the same terminal install precise-runner

3.3 Writing your first program

* In the same folder where you previously installed the binary, create a new Python file, name it Main.py
* Copy over the files from the USB to the same folder where Main.py is
* A computer screen shot of a program code

  Description automatically generatedEdit your Main.py file with the following code
* On line three modify *‘my\_model\_file.pb’*with the location ***path of the hey-computer.pb***
  1. Running your program
* Run the code by running **‘*python Main.py*’** in the same terminal window you opened earlier
* Now that the program is running say your wake word you trained earlier and the program should print *hello*in the terminal

4.1 Creating your wiring harness

* To successfully control a recliner, chair a custom wiring harness needs to be created based on this schematic.

Red Wire (voltage) – 4 Connections

Black Wire (ground) – 4 Connections

Pink Wire (M2-) – 1 Connection

Purple Wire (M1-) – 1 Connection

Blue Wire (M1+) – Connection

Green Wire (M2+) – 1 Connection

A diagram of a circuit board

Description automatically generated

4.2 Wiring your harness

* Wire the harness as indicated by the schematic

4.3 Connecting your Pi to the relay

* Use jumper cables to connect the GPIO pins from the Raspberry Pi to the relays as indicated by your module
* You can send digital signals to the relay using the GPIO pins from the Pi to trigger the actions

## Testing & Validation

### Physical Testing

The final prototype underwent extensive physical testing in order to validate its robustness and reliability:

* Drop test: Remote was dropped from 1 meter height 10 times in order to validate physical robustness of the design
* Shake test: Remote was shaken for 10 seconds 10 times in order to validate robustness of physical design and interior electronic connections
* Button endurance: Each button was clicked 50 times by each member fo the team to validate physical robustness

### Performance Testing

The final prototype underwent extensive testing in order to validate its functionality and real-life performance:

* Voice test: Each team member gave tested each voice command 25 times to make sure that the computer could understand the voice being spoken and the command being asked.
* Movement test: Once a command was registered, the movement of the chair was compared to what was expected to happen. This was made by giving each command 10 times and measuring the action taken and the final position / time of activation for each command.

### Results

Ideal performance was obtained in the following aspects:

### 

Marginal performance was obtained in the following aspects:

### 

# Conclusions and Recommendations for Future Work

Overall, this project was a great success since we were able to create a fully functioning voice-controlled solution for anyone to use, at a very low cost. It allows for great flexibility in a large variety of applications, both in what it can interpret as a command and what it can execute as an action

However, there is still a lot to improve in this project, as can be see in the list below.

1. **User friendliness:** Whilst the final prototype works extremely well for what it was intended to do, we were able to develop ways to make the model learn new voices. However this is very challenging and requires a high degree of knowledge in computer science. It would be ideal to implement a user interphase so they can be able to customize their experience as opposed to needing help to do it.
2. **Physical design:** The physical design of the product is not polished, it requires to be built in different materials such as laser cut acrylic or metal in order to enhance the aesthetics, increase robustness, and prevent damage to internal elements.
3. **Electrical design:** At the moment the electrical harness in the inside of the controller was experimentally built, and needs a lot of improvement in order to meet any type of fire or safety regulations.
4. Performance in noisy environments: It was discovered during testing that the voice recognition was not robust when performing in loud environments. This is due to the large amount of background noise that confuses the computer. This would need to be improved in order to better the performance of the remote in real life scenarios.

During this project we had the opportunity to learn a lot. However, if we had a chance to go back to the beginning and give ourselves some advice, it would be to develop all aspects of the project equally, instead of making specialized prototypes for each thing. This would be because we ran into the issue that our electrical plans for the interphase were always changing due to poor understanding of the remote and other aspects. In other words, we would like to make sure to test every aspect of the prototype in some way before proceeding to develop more specialized versions of it.

If we had a few more months to work on this we would like to figure out the aforementioned points, that is to clean up the design and improve the performance in noisy environments. However, we would also like the opportunity to take the project to market at least in a small scale, to gauge for interest and usefulness in real life.

APPENDICES

# APPENDIX I: Design Files

It is important to consult every available document for this project in order to better understand it. They are divided into 3 main zip files with all the information of different areas. All of the key documents are available in maker repo (link below) and can be divided as follows:

* Deliverables.zip: Include every deliverable with details steps into how each part was made, and the though process behind everything
* 3D Design.zip: Includes al the 3D designs for the case and the rendering of the product.
* Voice acrtivated remote.zip: Includes all the software that was used to make the remote work

<https://makerepo.com/JoseB/1673.voice-controlled-reclining-chair-remote-w-raspberry-pi>

Table 1. Referenced Documents

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Document Location and/or URL** | **Issuance Date** |
| Deliverables A-H | Maker Repo | 20 Jul 2023 |
| 3D Design | Maker Repo | 20 Jul 2023 |
| Voice Activated remote | Maker Repo | 20 Jul 2023 |
|  |  |  |
|  |  |  |