i.

University of Ottawa

Faculty of Engineering

GNG2101 Introduction to product Development and management for Engineers

Project Deliverable D

Submitted by:

Talk Box. C01, Team C13

Tia El Masry, 300160596

Zainab Badawi, 300034146

Kain Mozafarian, 300138481

February 7, 2021

University of Ottawa

Abstract

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
BOM	Bill of Materials

1 Introduction

The team had their second meeting with the client Roy and provided a brief overview of current concepts and potential solutions that arises from the teams brainstorming efforts. The solutions were demonstrated through images of 3D Solidworks models and user interface sketches (can be found in PD C). In this deliverable, the goal is to create small prototypes and test their functionality and practicality. The group's main focus for this prototyping stage is to benchmark the hardware and the inputs the user will likely be using, and by putting our mindset into the user's mindset, reach a conclusive decision for the design depending on its fidelity. In this stage of prototyping, the computing unit with its inputs and the mounting system will be benchmarked.

2 Client Feedback Analysis

The team demonstrates a summary of the client feedback statements and analysis on how it will affect our current design.

Client Feedback Statements

The client stated that the speaker holes are too small and expressed his concern that small speakers tend to have high pitch. Small speakers tend to have high pitch whine and are uncomfortable to the human ears. The client mentioned that the speakers need to be heard over the sound produced by the TV and loud enough to be heard by users losing their hearing.

The client does not see much value in having LED lights as indicators to the button's response. Additionally, the client added that port locations on the enclosures will need to be changed to match the port locations of the Raspberry Pi.

The client thinks that the slider is problematic to the user due to the risk of accidental movements. A suggestion if the slider is a necessity, then moving it away from the user's side to prevent accidental movements.

The client thinks it is a great idea of using a softer material for the top of the device, since it will be placed where the user usually tends to rest their arm, however there is the concern of its durability against alcohol-based cleaning materials, the deterioration over time, and “being abused” from absent-minded users.

The client adds that the caregiver can make edits to a spreadsheet and then saves a CSV copy to a thumb drive. the thumb drive is then plugged into the device and it copies the data over.

- this would be much easier to edit than trying to edit on the device itself
- it would require that the device be taken away from the user to reprogram it

Another approach would be to “pair” the device with a particular google sheet at install time. Then the caregiver could modify the sheet using a smartphone or computer, and the changes could be automatically downloaded to the user's device (or in response to a special action on the user's device).

The client thinks that the device’s ability to make a connection to a TV would be a plus, however it is not the main concern of the talk box, so if it is possible to implement it would be great.

The client had no objections in terms of the user interface, he mentioned that the client is a 55-year-old male, to simply keep in mind aesthetic-wise when creating the final additions on the user interface. When a custom word is generated by a caretaker, it should be stored in the speech (TTS) application.

The client believes that clicking multiple times would be exceedingly difficult for many clients and having only 3 clicks on each side only allows for 6 total functions which is very limiting. Additionally, the size overall needs to be smaller and the bottom edges of the device needs to be deburred.

The client mentioned to try to design a button layout that requires the least amount of effort in terms of moving an arm and causing as little friction as possible (for this case of user criteria, they can use their wrist and it requires the least amount of effort).

Client Feedback Analysis

The big takeaway from this feedback is that the button must be deburred in all edges, able to be mounted in any direction desired by the user, and reduce the steps needed to ask the talk box to perform a command. In addition, create a process that is easily accessible to caregivers from any device such as a smartphone or a tablet.

The client's feedback was taken into consideration when developing the updated design of the talk box. Considerable changes were made to improve the solution to the hardware and software requirements of the talk box. In summary, the button is no longer used to select between commands to choose a command, instead a joystick is used (can also use buttons instead of a joystick) to navigate between commands which are displayed on an 8cm x 3.5 cm LCD screen. Once the user lands on the desired selection they can press the button to perform the command. Finally, a text to speech algorithm will be used to say phrases aloud and communicate with other devices if required. These steps are demonstrated in Figure 1. Regarding adding new phrases/sentences for the user, the caregiver will be provided an interface to input such needs which then will be added to the database either via SQL, Firebase, or other database platforms, which can then be accessed by the user via HTTP request (it is just a get protocol, so the user does not need any understanding of what it actually is). To add new smart home devices, it is needed by the caregiver to set it up with a Google Home or an Echo (Alexa) and the task should not be any different compared to setting it up for themselves. After the setup, the appropriate commands can be added to the "simple phrases" database i.e., after adding a smart light, add "Turn off/on light" to the database.

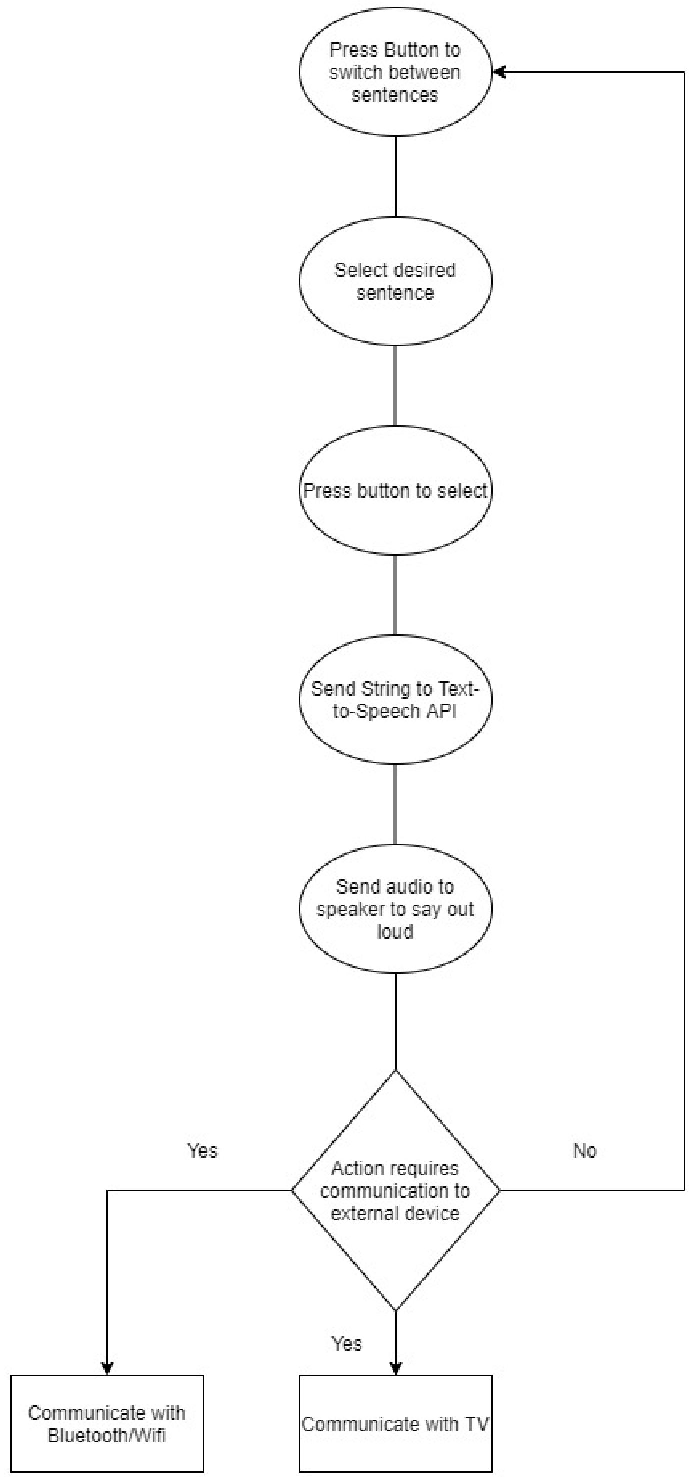


Figure 1: Flow Diagram of the Button Actions

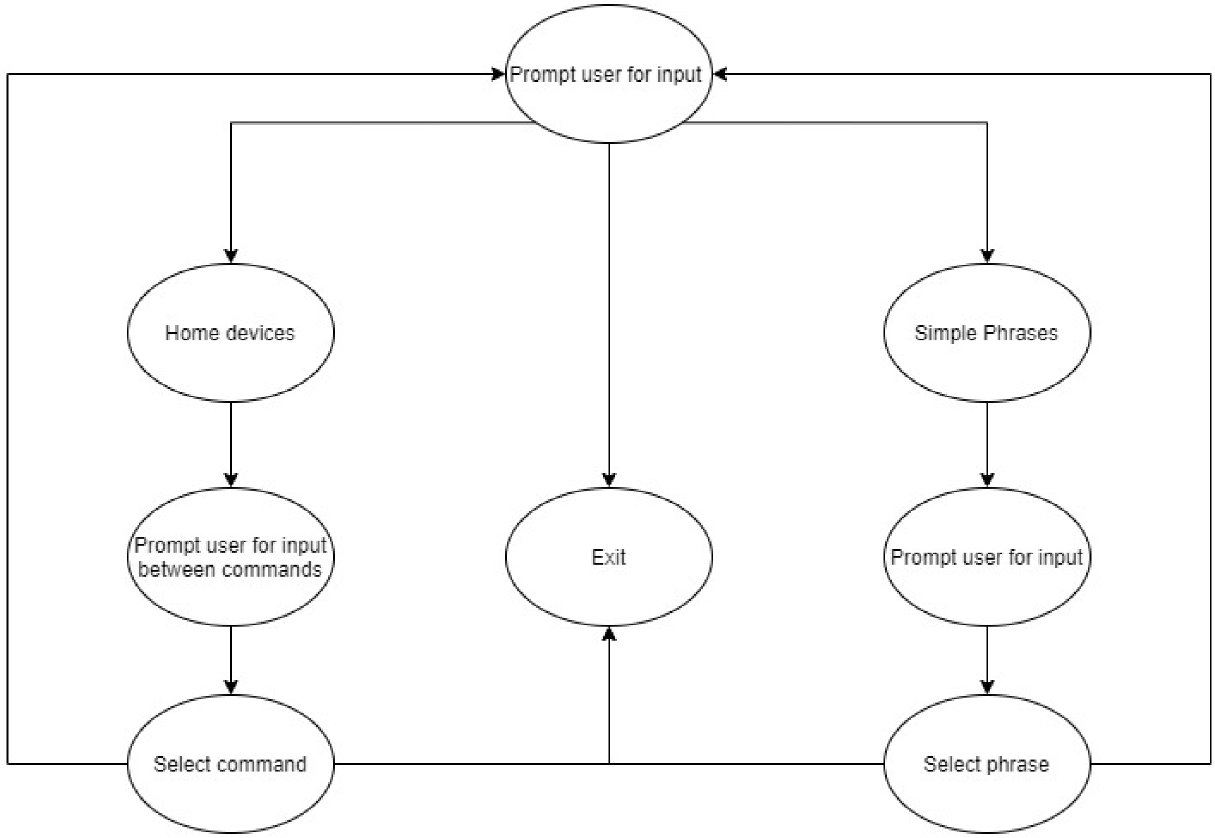


Figure 2: Flow Diagram of the process required to add new command to the talk box

3 Prototype

The prototype objectives are to test the reactiveness and the force required to select among options that the talk box can perform. In addition, to create a simple method to update the commands sent to the talk box and their output.

Hardware

Firstly, the size of the button was reduced to be (5.5cm x 5.5cm) and all edges were deburred. The slider and LED lights were eliminated, and the bigger speakers were chosen for the purpose of having a better-quality sound with lower frequency. In addition, the strap slots were redesigned and placed at a distance that will exert no stress on the button frame. The button is strapped tightly as seen in figure 3 on the wheelchair armrest. The final prototype design and dimensions for the button is shown in figure 3 and figure 4.

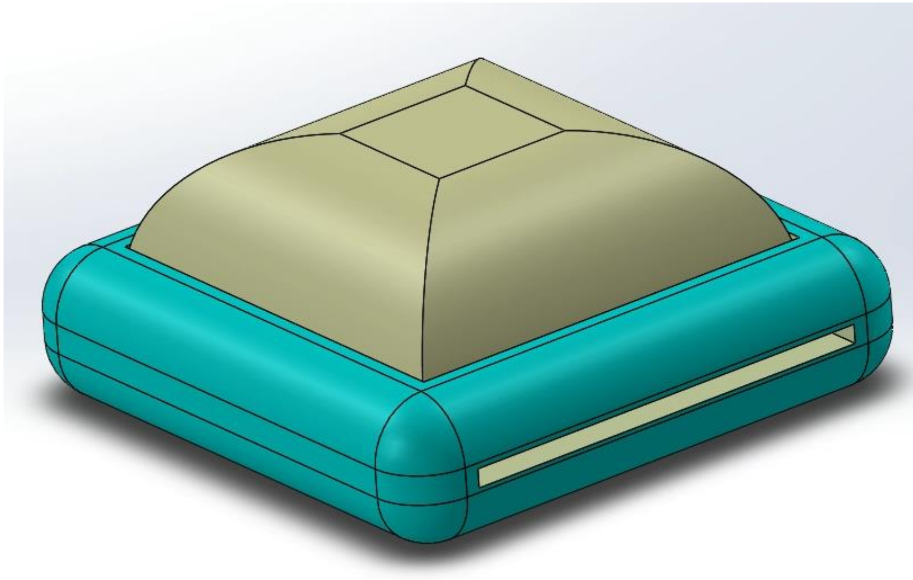


Figure 3: Button 3D CAD

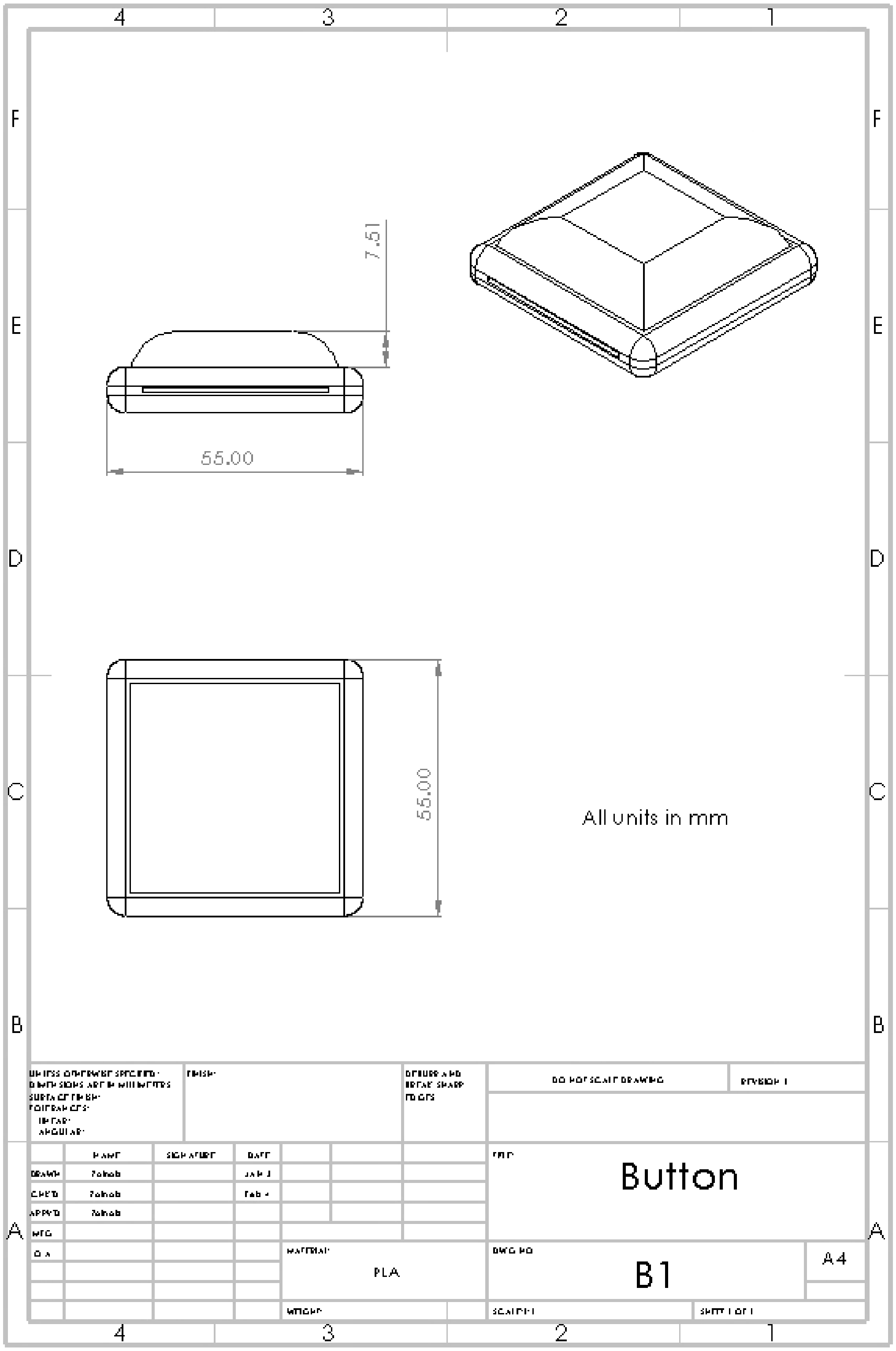


Figure 4: Button drawing

To prototype the wheelchair arm, a broomstick with a diameter of ~3 cm was used and secured with tape to a tabletop. The wheelchair armrest was prototyped using recycled amazon bubble packing and cardboard. Finally, a large black cloth mask was securely tightened around the bubble packing and cardboard.

To mount the prototype components of the talk box on the wheelchair armrest, a white piece of cardboard was tied with a white elastic strap as shown in figure 5 on the far right. The joystick was taped to the armrest and finally the button was secured with a black strap.

Figure 5: Prototype of the wheelchair tubing frame and armrest with talk box joystick and button

To prototype the behaviour of the button, the following electronics were used: an Arduino, a crystal LCD display, a Joystick, 220 Ohm resistor, a potentiometer, cables, and a button. The joystick and the button are used as an input while the display is showing the output. The microcontroller is programmed to only read input when the joystick moves in the horizontal axis. This requires the least amount of wrist movement from the user. Finally, the joystick can be pressed inwards to select a command. Figure 6 and 7 demonstrate a detailed image of the microcontroller setup.

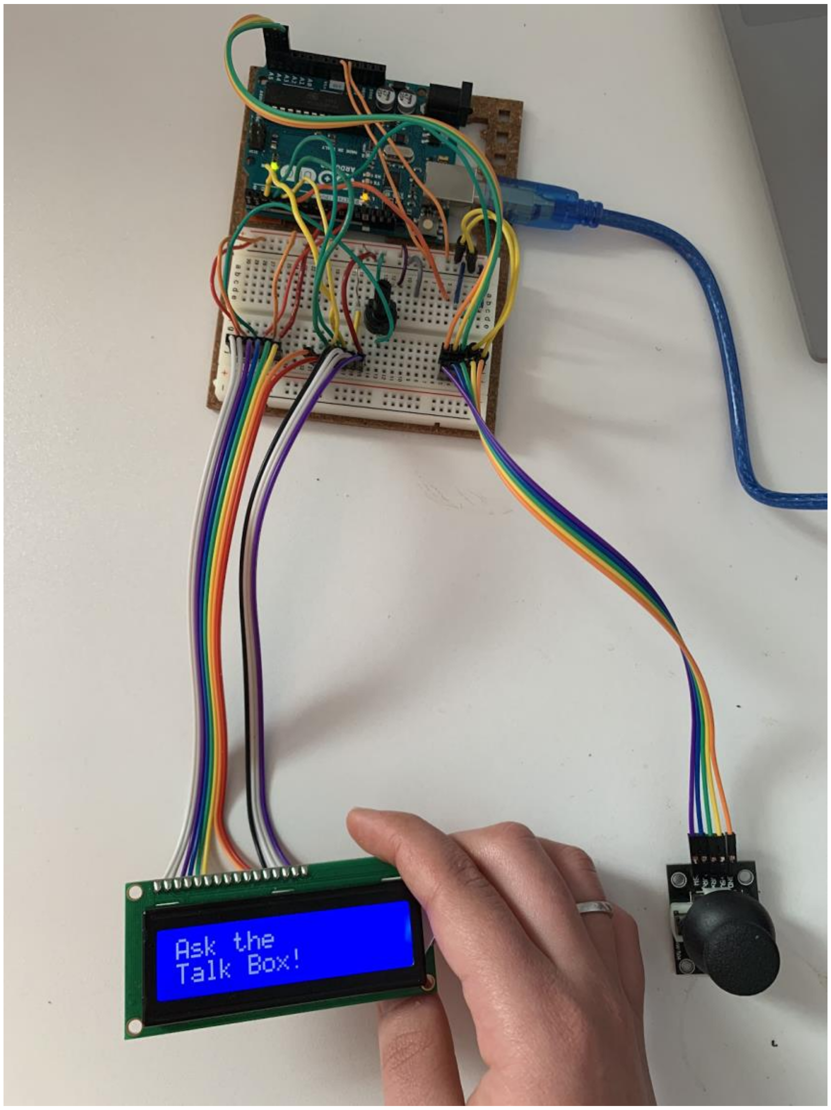


Figure 6: The complete image the microcontroller and other components

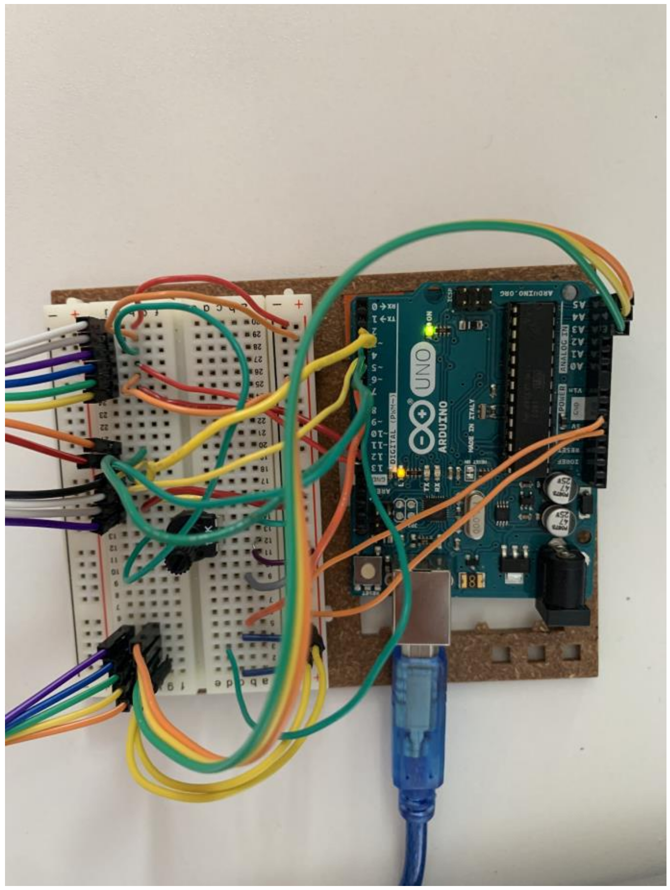


Figure 7: A detailed image of the microcontroller wiring

Software:

The software prototype would have the ability to read and decompose different text documents that contains different categories of phrases which the user can pick from. The first function would turn the text files into dictionaries that contain the number of each sentence as the key and the sentence as its value, and the return value would be the dictionary as a whole. The second function would take an integer and a specific dictionary as input and would return the value of the key that it is given. A simple run down of the program would start by essentially showing the user all the different categories and prompt the user to select one by picking a number. Then the program would show the user all the different options of sentences within that category and prompt the user to select one by picking a number. The number must be in the range of how many sentences there actually are in the dictionary, otherwise the program would show an error message and keep asking the user for an integer. Once a proper integer is selected, the program would return the sentence that it corresponds to and the program would shut off. Although this program won't behave exactly how the final product aims to work, it shows the basic idea of how to read a file that contains all the data of the possible phrases that a user might wish to say and store all of it in the same dictionary. It fulfills the idea of selecting a certain number to obtain a specific sentence.

Prototype 1.py - C:\Users\tiael\OneDrive\Desktop\Prototype1\Prototype 1.py (3.8.5)

File Edit Format Run Options Window Help

```
#This is Prototype 1 for TalkBox


def parsing_excel (file_name):
    '''(str) -> (dict)
    Returns a dictionary of the phrases in the txt file
    '''
    f = open(file_name).readlines()
    my_dict = {}

    for i in range(len(f)):
        t = f[i]
        my_dict[i+1] = t
    return my_dict

def sentence(x, dictionary):
    '''(integer) -> (str)
    Returns the sentence that corresponds to that integer
    '''
    return (dictionary[x])


#main

simple_phrases = "Simple Phrases.txt"
talk = parsing_excel(simple_phrases)

home_devices = "Home Devices.txt"
home = parsing_excel(home_devices)



print("Please select a category")
print("1. Home Devices")
print("2. Simple Phrases")
print("Press space if you wish to exit")
choice = input()


if choice == '1':
```

Ln: 65 Col: 9

Prototype 1.py - C:\Users\tiael\OneDrive\Desktop\Prototype1\Prototype 1.py (3.8.5)

File Edit Format Run Options Window Help

```python
if choice == '1':
    flag = True
    while flag:
        for i in range(1,len(home)+1):
            print (i, home[i])
        number = int(input("Please enter a number between 1-"+ str(len(home))+": "))

        if (number < 1) or (number > len(home)):
            print("Error: Please try again")
        else:
            flag = False
    print('\n')
    command = sentence(number, home)
    print (command)



elif choice == '2':
    flag = True
    while flag:
        for i in range(1,len(talk)+1):
            print (i, talk[i])
        number = int(input("Please enter a number between 1-"+ str(len(talk))+": "))

        if (number < 1) or (number > len(talk)):
            print("Error: Please try again")
        else:
            flag = False

    print('\n')
    say = sentence(number, talk)
    print (say)


else:
    print("Good bye :)")
```

Ln: 65 Col: 9

4 Prototype testing

The team agreed on the following list of assumptions to carry out the testing of the talk box components:

- The User can make a fist and is able to move the joystick
- The User can provide power to the microcontroller using the wheelchair
- The User uses a wheelchair that has an armrest that is 7cm wide
- The caregiver can use smart devices
- The User can read the text on the LCD display
- The User has Wi-Fi connection in their residence

To test out the behavior of the behavior of the button the following Arduino code was put into action. The

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2);

const int switchPin = 6;
int switchState = 0;
int prevSwitchState = 0;
int reply;
int ButtonState =0;
int x,y,z;
int scroll = 0;
int scrollprevious;

void setup() {
  // put your setup code here, to run once:

  pinMode(A5,INPUT);//Z-button
  pinMode(A4, INPUT); //Y-direction
  pinMode(A3, INPUT);//X-direction
  lcd.begin(16, 2);
  pinMode(switchPin, INPUT);
  lcd.print("Ask the");
  lcd.setCursor(0,1);
  lcd.print("Talk Box!");
  pinMode(ButtonState, INPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  //if value is high increment counter, else if the value is low decrement counter
  scrollprevious = scroll;
  //have if else statements each outputting one one selection based on the counter.
  //inner if: button pressed the then "this is selected" and goes to speaker
  z=analogRead(A5);//read-Z
  y=analogRead(A4);//read-Y
  x=analogRead(A3);//read-X
  /*
   * print joystick coordinates
   */

//   Serial.print(x); //right = 1020, left = 0
//   Serial.print(" , ");
//   Serial.print(y); // up = 1020, down = 0
//   Serial.print(" , ");
//   Serial.println(z); //push = 0
//
//   delay(150);

if(x>1000 && scroll < 8){
  scroll++;
}else if(x<100 && scroll >=0){
  scroll--;
}
 // switchState = analogRead (switchPin);
 // ButtonState = digitalRead(7);
  if (scrollprevious != scroll){
   // if (switchState == LOW){
    //  reply = random(8);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("select: ");
      lcd.setCursor(0,1);
      switch(scroll){
        case 0:
        lcd.print("Say 'I Love you'");
        if(ButtonState == HIGH){
           lcd.setCursor(0,0);
           lcd.print("Say 'I Love you'");
           lcd.setCursor(0,1);
           lcd.print("Say 'I Love you'");
        }
        break;
        case 1:
        lcd.print("Say 'Thank you'");
        break;
        case 2:
        lcd.print("Say'Leave me alone'");
        break;
        case 3:
        lcd.print("Say'I'm hungary'");
```

```
        break;
        case 4:
        lcd.print("Lights on/off");
        break;
        case 5:
        lcd.print("TV on/off");
        break;
        case 6:
        lcd.print("Call for help");
        break;
        case 7:
        lcd.print("Play Music");
        break;

      }

    }
    delay(150);
  }
  //prevSwitchState = switchState;
//  ButtonState =0;
```

*Along with this report a video is submitted to the showing the code in Action!

Table 1: Final Target Specs for comparison

Target #	Metric	Unit	Expected Value	Actual Value
1	Button response time	Processing time	Fast	Fast*
2	Capability of simple dialogue commands	Processing time	N/A	Somewhat Fast
5	User friendliness	Usability	3	3
7	Force required to push button	N	>4	>~10

8	Travel Before Activation	cm	N/A	0.5cm
9	Size of button(s)	cm^2	~6.5	30.25
10	Space between button(s)	cm	N/A	Custom
14	Features offered	# of functionalities	3	2
15	Ability to mount on either side of wheelchair	Wire length (cm)	N/A	20 cm
16	Device comfort	Comfort scale	N/A	2
18	Maximum cost	Cad $	>=100	$75
19	Screen Size	cm	~38	8x4cm
20	Weight	gram	<2000	200g

Jumper cables were used to connect the LCD screen and joystick. These cables are 20 cm long which is not an ideal distance to be able to mount the components on either side. however, in the next prototypes longer jumper cables will be purchased. The device comfort was compared to the medical device comfort scale and it scored 3 out of 5. The device is comfortable to use for a long period of time as it does not require a lot of wrist movement to move the joystick. The screen size is less than the expected value but for the second prototype a larger screen will be purchased. The weight of the component is divided into the weight of the LCD screen, joystick and the button. The LCD screen weighs 100g, the joystick weighs 50g, the PLA button weighs about 50g. The number of features offered was based on the different categories that were implemented in the software prototype, which was the home devices section and the simple phrases section.

5 Bill of Materials (BOM)

Table 2: Raspberry Pi Build Cost

Parts: Price	**Images**	**Links**
Touch Sensitive Switches: $7.50		https://www.amazon.ca/dp/B07HRRVZ11?th=1&psc=1
Raspberry Pi 3 - Model A+: $32.00		https://www.buyapi.ca/product/raspberry-pi-3-model-a-plus-512mb-ram/
Device Mount: $0		

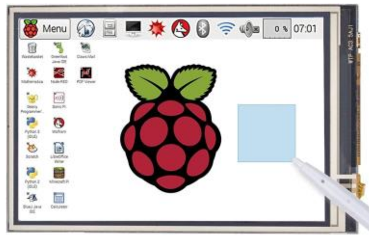
Display: $24.00		https://www.amazon.ca/Raspberry-Touchscreen-Monitor-SunFounder-480x320-Display/dp/B07SKVF392/ref=sr_1_29?dchild=1&keywords=HDMI+Monitor&qid=1612649178&sr=8-29
Speaker: $13.95		https://www.buyapi.ca/product/usb-powered-speakers/

Table 3: Arduino Build Cost

Parts: Price	**Images**	**Links**
Touch Sensitive Switches: $7.50		https://www.amazon.ca/dp/B07HRRVZ11?th=1&psc=1
Arduino: UNO R3: $15.00		https://amzn.to/3oVDqIu
Mount: $0		

Arduino WiFi adapter: $11.00		https://www.amazon.ca/KeeYees-Internet-Development-Wireless-Compatible/dp/B07PR9T5R5/ref=sr_1_5?crid=FN5ATXZZRBA8&dchild=1&keywords=arduino+wifi+module&qid=1612479314&sprefix=arduino+wifi+%2Caps%2C162&sr=8-5
Display: $27.75		https://www.amazon.ca/480x320-Screen-Module-Arduino-Without/dp/B07NWH47PV/ref=sr_1_10?dchild=1&keywords=arduino+display+rgb&qid=1612478634&s=electronics&sr=1-10
Speaker: $10.88		https://amzn.to/3jjUKW7
LM386N: $18.05		https://www.amazon.ca/National-Semiconductor-LM386N-1-Voltage-Amplifier/dp/B01MUHBZ91/ref=sr_1_13?dchild=1&keywords=LM386&qid=1612480265&sr=8-13

By researching and comparing both microcontrollers from Raspberry Pi and Arduino, a few conclusions could be made. Firstly, while microcontrollers from both companies cost different amounts of money, the utilities they offer differ a lot but can be compensated by 3rd party accessories i.e the wifi adapter for the Arduino UNO; with this considered, the addition of the cost for all the parts comes around $70+ dollars for both devices/computing units. Secondly, on a closer inspection, while Arduino UNO would give more experience putting parts together, it's possible to see that the Raspberry Pi requires fewer parts and less assembly to get up and running for our project, thus saving us time for software development purposes that are required to actually make the project plausible. The Raspberry Pi offers many ports that would otherwise need to be considered as a separate issue when it comes to using an Arduino. So for this project, Raspberry Pi has been favored but that is not to say that arduino does not hold any potential, in fact if there was more time for development and perhaps a bigger budget, the UNO microcontroller would suffice as a versatile computing unit that would benefit this project.

6 Conclusion

The group was successful in creating a high-fidelity prototype for the computing unit, by using an Arduino UNO, it was possible to replicate a simplified version of the final design. By using an analog stick, we were able to navigate through some predefined words and actions to be performed. The options were successfully displayed on an 8-bit LCD display. For the wheelchair mount, the experimentation was also successful, however it could be considered as low-medium fidelity since only the armrest was simulated and tested on, not an entire wheelchair to see where all the possible mount locations are. Furthermore, it is possible to either host a webpage that is running on the Raspberry Pi in order to access the smart home system, or just make the Pi act as a server which hosts a custom application (if time permits for development) to execute smart home functionalities. Conclusively, this prototyping session has yielded appropriate and bountiful results.

Bibliography

LCD model specifications

https://www.arduino.cc/documents/datasheets/LCDscreen.PDF