Design Specifications Specifications Document

Sign Language Assistance
ECET 400-102
2/29/2024

Team Members:

Matthew Frazier

SungWon Lee

Oluwadamilare Ajayi

Submitted to:

Professor Blinder

Table of Contents

List of Tables and Figures

1. Product Summary

This project utilizes a Raspberry Pi 4 to take in user speech and translate it into ASL. Along with the Pi the main components will consist of a touch screen, microphone, and 3D printed robotic hand. Combining these components will allow us to take user speech, break each word down into single letters, and have the hand sign each letter spelling it in ASL.

The screen will display a graphical user interface, GUI, that has all the letters of the alphabet as individual buttons. These buttons will also be able to trigger the robotic hand as it will sign each letter when pressed. Another button on the GUI will be the record button that will activate the microphone for a short period of time to allow the Pi to record speech using the connected microphone. The screen will also display the current letter it is signing in a text box.

In the meantime, all the letters will also be sent to the Arduino that is serially connected with Pi. And based on what letter that Arduino received from Pi, it will control the potentiometer that is connected to the robotic hand.

To show accurate hand signals we will utilize a robotic hand. The hand itself will be put together by 3D printed parts. The fingers will be attached by a strong fishing line to servo motors. There will be one motor for each finger to allow them to bend at different angles for the appropriate hand signals. There are also two servo motors in the wrist to control the X and Y directions of the hand. Attached to the hand will be an elbow joint also controlled by a motor, which will allow the arm to turn upside down as some ASL signs are with the hand facing the floor.

2. Requirements

1. GUI displays 26 letter buttons, record button, and text box
2. Microphone records voice and sends to raspberry pi
3. Raspberry pi interprets data from screen and microphone and converts to text.
4. Raspberry pi communicates with arduino through serial connection to send strings
5. Arduino can interpret strings received through serial connection and choose the correct function
6. Arduino can call the correct letter function and send the position to the servos to create sign language corresponding from English to ASL.

3. Requirement Specifications

3.1 Technical General Description

The GUI and the microphone will serve as user inputs. The inputs will be sent to the raspberry pi through DSI connection and USB connection respectively. The pi will communicate information with the arduino uno through serial port connections. The arduino uno will use its digital ports to pulse width modulation signals to the servo motors on the correct positioning.

3.2 Power Requirements

3.2a Sources

Component	Power Source
Raspberry Pi	Adafruit USB Battery Pack (2 usb ports)*
Arduino	Adafruit USB Battery Pack (2 usb ports)*
Servos	Arduino
Screen	Raspberry Pi

Table 1

3.2b Voltage and Current Requirements

Component	Voltage (V)	Current (A)
Raspberry Pi	5.1	3.0
Arduino	4.8 - 6.0	100-250 m
10” Touch Screen	5.0	3.0
SG90 Servo Motors	4.8 - 6	100-250 m
Digital Servo	4.8 - 6.8	1.0
Microphone	3-5	50-500 m

Table 2

3.3 Case Requirements

3.3a. Description

The casing for all of this device will be 3D printed. The main component, raspberry pi and screen, will be housed in a triangular box. The top end of the box will be angled to allow for the screen to face the user as it will be mounted to the box. All other components, pi and wiring, will be housed inside the case. There will be a cutout for wires to leave to connect to the arduino and for the microphone to stick out the back. For the bottom of the case it will be a simple rectangle that interlocks with the top of the casing.

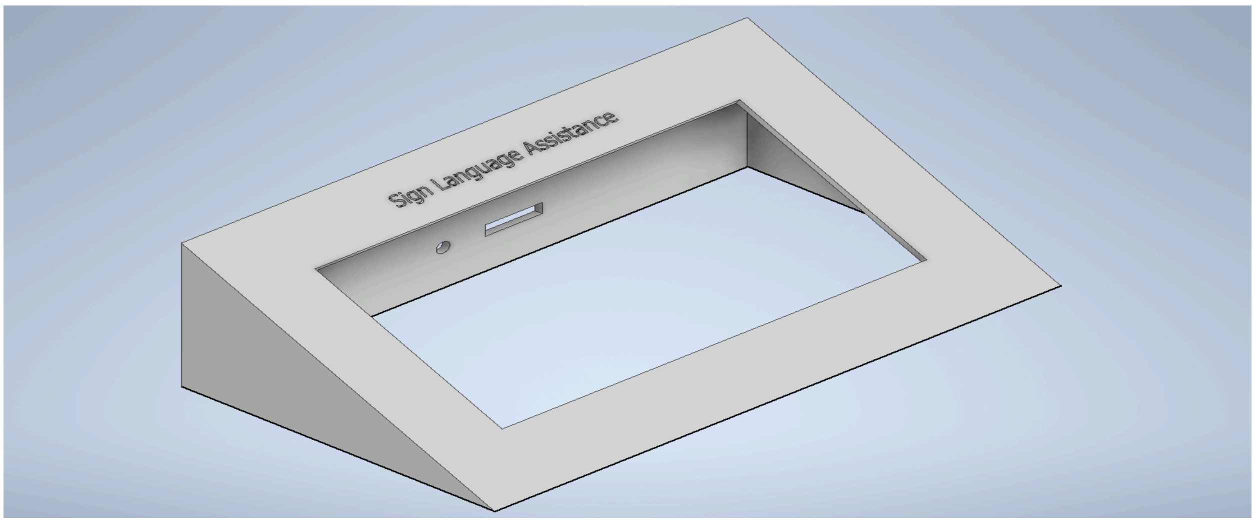


Fig. 1

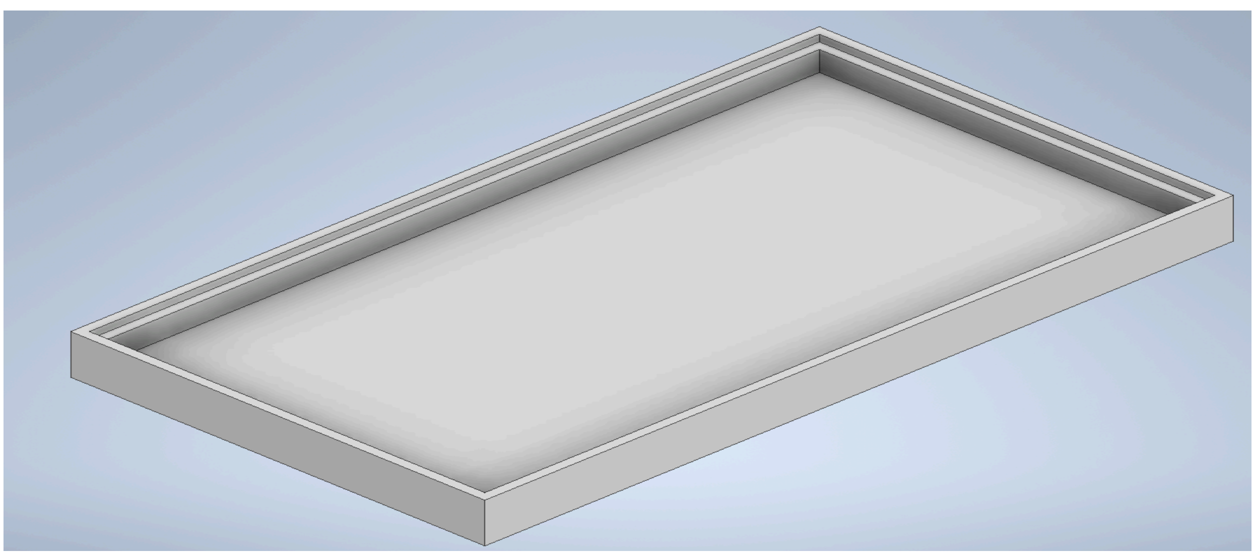


Fig. 2

3.3b Overall Case Size (HxWxD)

6.18”x9.0”x15.5”

3.3c Material

The case will be printed out of PETG (polyethylene terephthalate glycol). This is a stronger 3D printing material than PLA, which is usually used for 3D printing. This will allow the device to be more durable allowing it to last longer.

3.3d Special Features

The case will feature the product name engraved at the top, while at the bottom in braille the name of the product will be embossed.

3.4 Technical Requirements

3.4a Signal Inputs

Signal	Voltage (V)	Frequency (Hz)	Digital/Analog
1. Capacitive Touch (Screen)	5	N/A	Digital
2. Voice	N/A	80 - 300	Analog
3. Power	5	N/A	Analog

Table 3

3.4b Signal Outputs

Signal	Voltage (V)	Frequency (Hz)	Digital/Analog
1. Display	5	N/A	Digital
2. Pulse Width Modulation	0 - 5	50	Digital

Table 4

3.4c Major Internal Signals

Devices	Description
Arduino	UART
Microphone	Processes sound waves to analog data
Raspberry Pi	UART
Screen	DSI to send digital data from screen to pi
Servo Motors	Pulse Width Modulation

3.4d Microcontroller

Microcontroller	Description
Atmel MXT640T	Touch screen microcontroller, used to interface with the raspberry pi 4 through the DSI port.
ATmega 2560	The arduino houses this microcontroller and allows for direct interaction. This microcontroller will be tasked with communicating with the raspberry pi and running the servo motors.

Table 5

3.4e Ports

USB port on Raspberry Pi for plug microphone, Arduino connection, External battery if we need external power supply. Output ports from Arduino so it can control the potentiometer that will move wrist and fingers

3.4f Keyboard and Display

10 inches display to display speech to text file.

3.4g Other

3.5 Operational Requirements

3.5a Data Stored

- The system must store data related to user preferences, including language settings, display configurations, and personalized learning profiles.
- Arduino will store functions of each letter and their corresponding servo motor movements.

3.5b Data Loaded

- Upon system initialization, user preferences and settings are loaded from the stored data to personalize the user experience.
- Any updates or modifications to the system's database, including new gestures or vocabulary additions, are loaded as needed to ensure the system's accuracy and relevance.

3.5c Program Executed

- Speech recognition functionality is triggered upon receiving voice input through the microphone, initiating the process of transcribing spoken words into text.
- The GUI will run at all times on the screen, displaying letters to be pressed. It will also update the text box to show what was pressed or spoken.
- Arduino will have pre programmed functions corresponding to each letter of the alphabet. These functions will be called depending on the letters received from the pi.

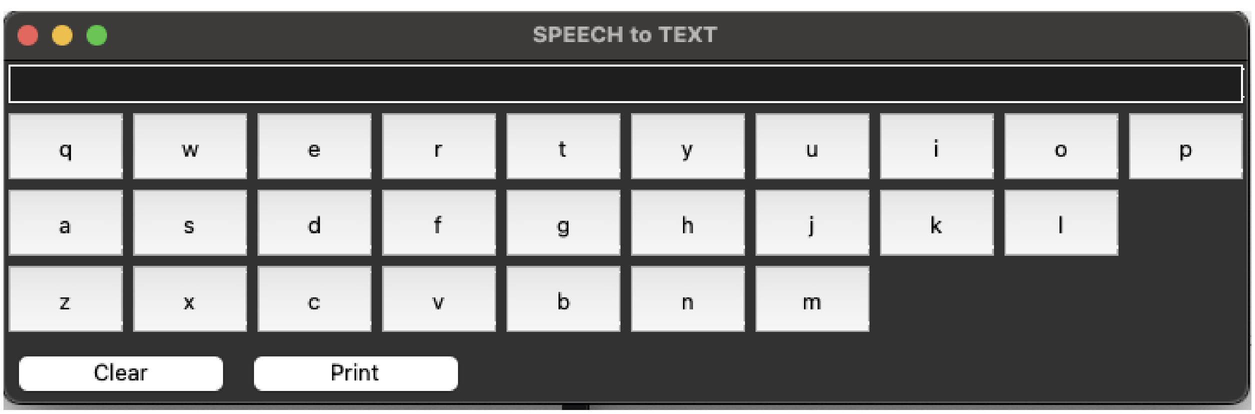


Fig 3

3.5d. Test Procedures

1. Test GUI compiles on the raspberry pi. Run on pi and test to make sure all buttons can be pressed on touch screen and the correct letter displays on the screen.
2. Testing the arduino is capable of communicating with the raspberry pi via serial communication to send strings to be interpreted by the arduino.
3. Arduino communicates with one SG90 servo. Servo is able to move its full range of motion smoothly without delay.
4. Button on GUI is pressed, information goes from GUI and is converted to a string on the raspberry pi. Raspberry pi sends strings to arduino. Arduino successfully receives and recognizes letters.
5. Button on GUI is pressed, information processed by raspberry pi and sent to arduino. Arduino processes string and calls functions to move the servo motors into the correct position.
6. Microphone records 5 seconds of voice. Raspberry pi processes speech and turns it into a string, and displays a string that Raspberry pi recorded on GUI.
7. Microphone records voice. Raspberry pi processes speech and turns it into a string. Displays a string that Raspberry pi recorded on GUI. Send string to arduino, arduino processes string, and selects correct functions. Arduino sends correct functions to servos for positioning.

3.6 Cost and Schedule Requirements

6a. Materials and Cost

Hardware	Description	Quantity	Price	Total Price
20kg Servo Motor	Will rotate and bent the wrist	3	$13.59	$40.77
SG90 Micro Servo Motor	Will bent or release the fingers based on the input text	7	$2.43	$16.99
3D Printer Filament	Will be use to print fingers and arms	2	$14.99	$29.99
Microphone	Will send speech signal to Raspberry Pi, so it can translate speech to text	1	$16.99	$16.99
Touch Screen	Will receive the text data	1	$68.99	$68.99

	from Raspberry Pi and will display the text			
Fishing Wire	Used to connect fingers. This will act like muscle tissue on fingers	1	$5.99	$5.99
Hinges	Use to make a door on arm, where we will put the bread board	2	$2.84	$5.67
Glue gun & Glue Stick	Will use to glue fishing wire with fingers, motor with fishing wire, and wrist with motor	1	$9.99	$9.99
Raspberry Pi	Translate speech to text, sent text data to touch screen to display the text and Arduino to move robotic arm	1	$76.49	$76.49
Arduino Uno	Receive text from Raspberry Pi, send signal to robotic arm based on text it receive	1	$27.60	$27.60
Bread Board	Will hold Arduino and will connection between arduino and servo motors	1	$8.99	$8.99
USB Battery Pack	External power source that produce power to Raspberry Pi & Arduino	1	$39.95	$39.95
				$347.41

Table 06

Estimated cost is $347.41 exclude tax, and it should not exceed $500 include tax

6b. Development and Cost

Python	$59/hr
Circuiting & Wiring	$45/hr
Hardware	$32/hr
Total Price	

Table 07

6c. Project Should

1	Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13
2	Research about what project to do (Sungwon Lee, Matthew Frazier, Oluwadamilare Ajayi)	1	1	1	1									
3	Research about hardware we need for project (Matthew Frazier, Sungwon Lee)		1	1	1	1								
4	Research about software we will use for project (Oluwadamilare Ajayi, Sungwon Lee)		1	1	1	1	1	1	1	1	1	1	1	
5	Buy the hardware we need (Matthew Frazier, Oluwadamilare Ajayi)				1	1								
6														
7	Programming (Oluwadamilare Ajayi, Sungwon Lee)					1	1	1	1	1	1			
8	Hardware building (Matthew Frazier, Sungwon Lee)								1	1	1			
9														
10	Trouble shooting (Sungwon Lee, Matthew Frazier, Oluwadamilare Ajayi)					1	1	1	1	1	1	1	1	
11														
12	Work on presentation (Sungwon Lee, Matthew Frazier, Oluwadamilare Ajayi)									1	1	1	1	
13														
14	Presentation Date (Sungwon Lee, Matthew Frazier, Oluwadamilare Ajayi)													1

Fig. 4

4. Top Level Block Diagram

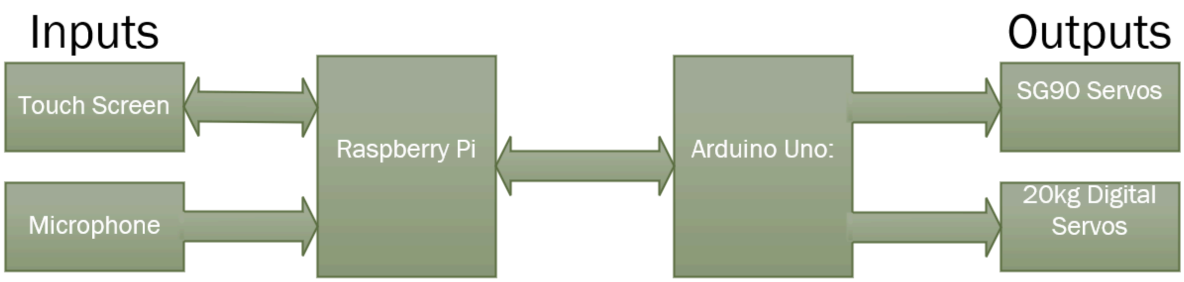


Fig. 5

5. Top Level Diagram Breakdown

The user will interface with the microphone and touch screen. Both components will be connected to the raspberry pi. The pi will also send data back for the screen to display. Once processes are run by the pi the data will be sent over to the arduino uno. The uno will pick the correct function to run, based on the data given by the pi. After that function is selected it will send information to the servos to tell them how much to move.

6. <u>Second Level Block Diagram</u>

6.a Block Diagram

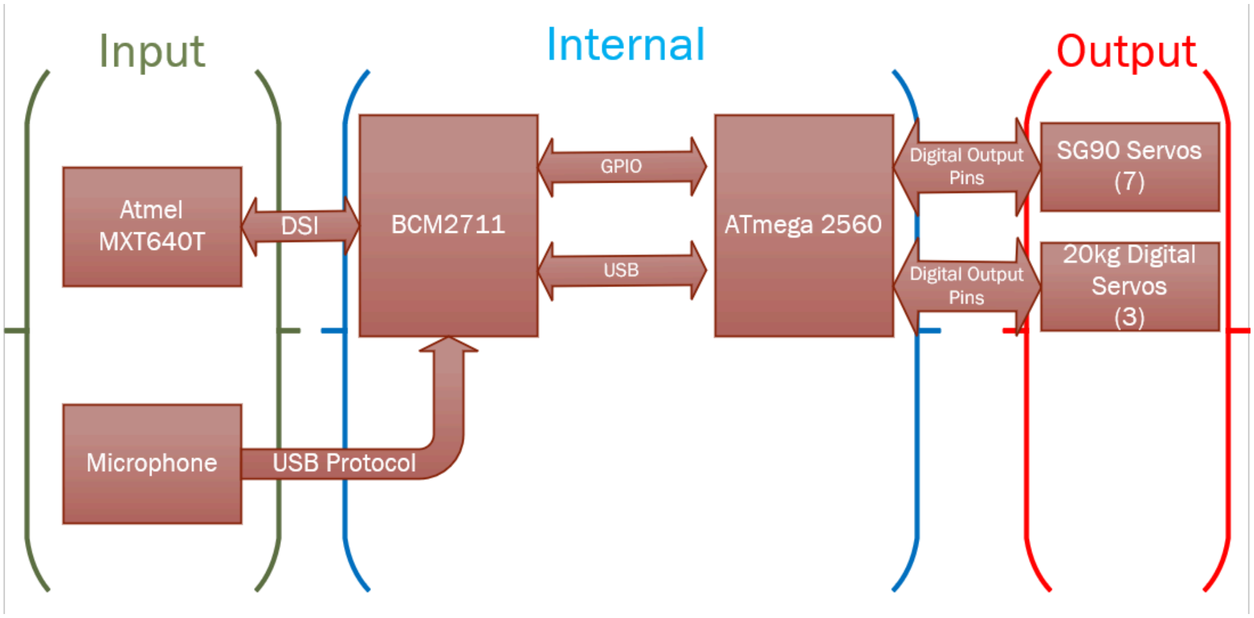


Fig. 6

6.b Block Diagram Explanation

The MXT640T is the microprocessor within the touch screen. This will handle communication of touch and text display through the DSI connection, display serial interface. The microphone will be connected to the raspberry pi through standard usb connection. It will send voice data to the BCM2711 chip on the raspberry pi. The BCM will process data from the screen and the microphone, converting button presses and speech to text. Once processed the text will be sent back along the DSI connection to be displayed on the screen.

The BCM will be connected to the ATmega 2560, housed on the arduino, through their serial ports. This will allow the BCM to send the Atmega information through the built in pins on the board. The ATmega is tasked with receiving the information and sending PWM signals to the servo motors for their correct positioning.

7. Program Flow Chart

7.a Flow Chart

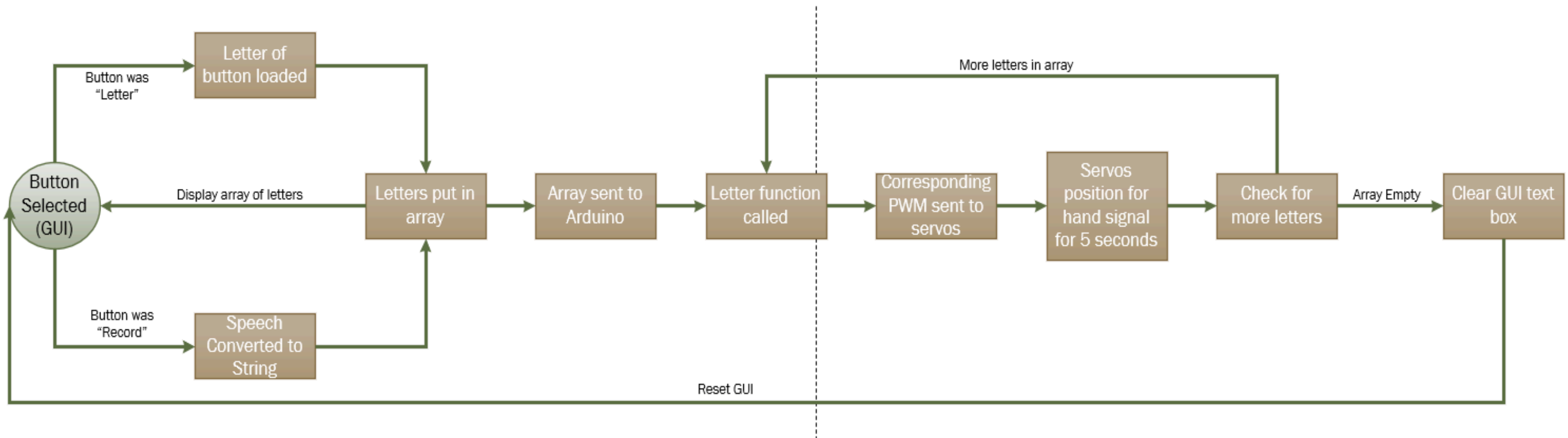


Fig. 7

7.b Flowchart description

Program starts at the graphical user interface, on the raspberry pi in python, where the user is able to select from 27 buttons. Each button corresponds to a letter with the last being recorded. If the record button is pressed the microphone records 5 seconds of user speech. That speech is then converted to text with each letter being put in a different position of an array. Similarly if a button corresponding to a letter is pressed that letter is converted into an array. The array is then sent to the arduino to be processed as a function and the screen to be displayed as text.

The letter is processed and its corresponding function is called upon in the arduino programming, C++. The function has pre programmed positions for each servo that match the finger position for the letter's hand sign. After holding the sign for 5 seconds the loop checks to see if the array is empty. If the array is not empty the loop starts back at the top calling the next letter's function again. If the array is empty the arduino sends a signal back to the pi. The pi clears the textbox of the GUI and starts from the beginning.

8. Major Component Details

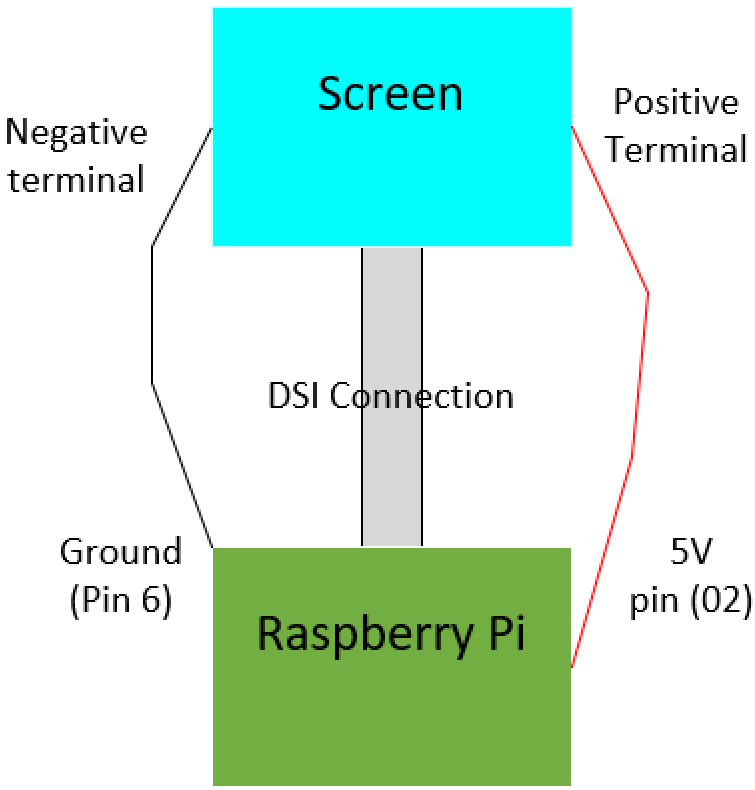
Hardware	Type	Reason	Total Price
Servo 1	Sg90	The sg90 is a cheaper lower power servo that is perfect for pulling the line to move the fingers given that the fingers are not extremely heavy.	$40.77
Servo 2	20kg Digital	To move the entire arm a servo with more torque is needed. The 20kg digital servo is reliable and able to handle heavier weights	$16.99
Microphone	ChanGeek mini USB	The raspberry pi supports microphone usage through USB connection. Since the microphone has to be close to the device a small USB one was chosen	$16.99
Touch Screen	LC0880 LCD Display	The display is the official display supported by the raspberry pi. To avoid compatibility issues this screen was chosen for reliable display.	$68.99
Mini Computer	Raspberry Pi 4	To support the GUI and communication to the microcontroller the Pi was chosen for its computing power. This device contains enough storage and memory for what is needed.	$76.49
Microcontroller	Arduino UNO	Arduino is great for handling multiple servo motors. It is also able to communicate with the raspberry pi making it perfect for this project.	$27.60

Table 08

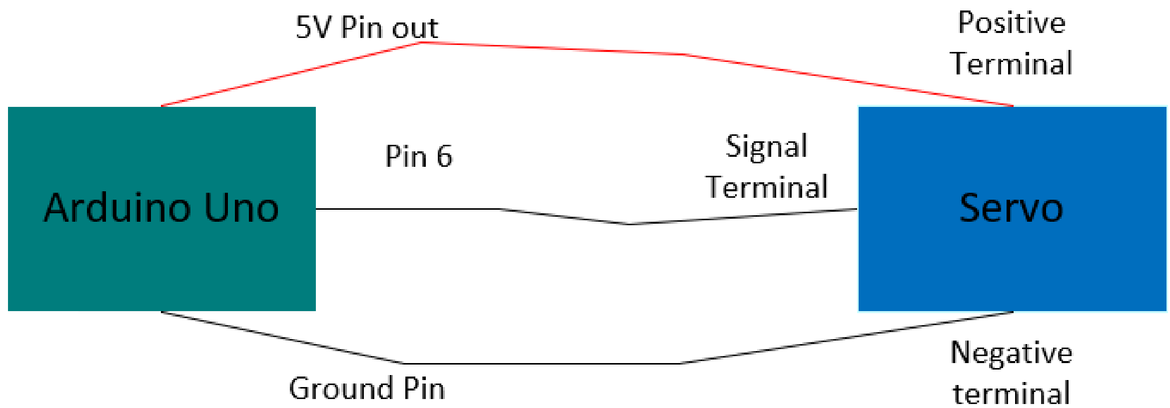
Appendices

Appendix A: Circuit Diagrams

Screen to Pi diagram



Arduino to Servo Diagram



Appendix B: Coding With Comments

- ***Code for GUI***

```python
import tkinter as tk

def add_letter(letter):
    current_text = text_entry.get()
    text_entry.insert(tk.END, letter)

def clear_text():
    text_entry.delete(0, tk.END)

def print_text():
    typed_text = text_entry.get()
    print("Typed text:", typed_text)

root = tk.Tk()
root.title("SPEECH to TEXT")

# Text entry widget to display typed text
text_entry = tk.Entry(root, justify="center")
text_entry.pack(fill=tk.X)

# Create buttons for each letter of the alphabet in a QWERTY layout
alpha_layout = [
    'qwertyuiop',
    'asdfghjkl',
    'zxcvbnm'
]

for line in alpha_layout:
    button_frame = tk.Frame(root)
    button_frame.pack(fill=tk.X)
    for letter in line:
        button = tk.Button(button_frame, text=letter, width=4, height=2,
                           command=lambda l=letter: add_letter(l))
        button.pack(side=tk.LEFT)

# Clear button
clear_button = tk.Button(root, text="Clear", width=10, command=clear_text)
clear_button.pack(side=tk.LEFT, padx=5, pady=5)

# Print button
print_button = tk.Button(root, text="Print", width=10, command=print_text)
print_button.pack(side=tk.LEFT, padx=5, pady=5)

root.mainloop()
```

Fig. 8

- **Code for servo setup and first letter positioning**

Hand-Motion-Codes-V1.ino

```
#include <Servo.h>

//Set fingers as servos
Servo pinky;
Servo ring;
Servo middle;
Servo pointer;
Servo thumb;
Servo wristx;

//set variable for incoming pi letter
String sensorPin = 2;

void setup() {
  //Set pins for servos
  pinky.attach(3);
  ring.attach(5);
  middle.attach(6);
  pointer.attach(9);
  thumb.attach(10);
  wristx.attach(11);

  //Open pin 2 for input through serial com. with pi
  Serial.begin(9600);
}

void move_a(){
pinky.write(45); //Put pinky down
delay(500);      //Delay to allow to move
ring.write(45); //Put ring down
delay(500);      //Delay to allow to move
middle.write(45); //Put middle down
delay(500);        //Delay to allow to move
pointer.write(45);  //Put pointer down
delay(500);          //Delay to allow to move
thumb.write(0);   //Keep thumb straight
delay(500);        //Delay to allow to move
}

```

Fig. 9

Appendix C: Potential Problems and Obstacles

- Raspberry Pi has trouble communicating with Arduino through serial connection
  - Change connection type between the two
- SG90 servos are not strong enough to move finger correctly
  - Switch all servos to 20kg digital servos

Appendix D: Bill of Materials

Hardware	Type	Quantity	Price	Total Price
20kg Servo Motor	Will rotate and bent the wrist	3	$13.59	$40.77
SG90 Micro Servo Motor	Will bent or release the fingers based on the input text	7	$2.43	$16.99
3D Printer Filament	Will be use to print fingers and arms	2	$14.99	$29.99
Microphone	Will send speech signal to Raspberry Pi, so it can translate speech to text	1	$16.99	$16.99
Touch Screen	Will receive the text data from Raspberry Pi and will display the text	1	$68.99	$68.99
Fishing Wire	Used to connect fingers. This will act like muscle tissue on fingers	1	$5.99	$5.99
Hinges	Use to make a door on arm, where we will put the bread board	2	$2.84	$5.67
Glue gun & Glue Stick	Will use to glue fishing wire with fingers, motor with fishing wire, and wrist with motor	1	$9.99	$9.99
Raspberry Pi	Translate speech to text, sent	1	$76.49	$76.49

	text data to touch screen to display the text and Arduino to move robotic arm			
Arduino Uno	Receive text from Raspberry Pi, send signal to robotic arm based on text it receive	1	$27.60	$27.60
Bread Board	Will hold Arduino and will connection between arduino and servo motors	1	$8.99	$8.99
USB Battery Pack	External power source that produce power to Raspberry Pi & Arduino	1	$39.95	$39.95
				$347.41

Table 09

Appendix E: Calculations

PWM Calculations:

Period = 1/ Frequency

Ton = Start time　　Toff = End time　　T = Start Time - End Time

Duty Cycle = (Ton / T) * 100

Appendix F: Important Data Sheets

- **Raspberry Pi data sheet**

4.1 Power Requirements

The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.

5 Peripherals

5.1 GPIO Interface

The Pi4B makes 28 BCM2711 GPIOs available via a standard Raspberry Pi 40-pin header. This header is backwards compatible with all previous Raspberry Pi boards with a 40-way header.

5.1.1 GPIO Pin Assignments

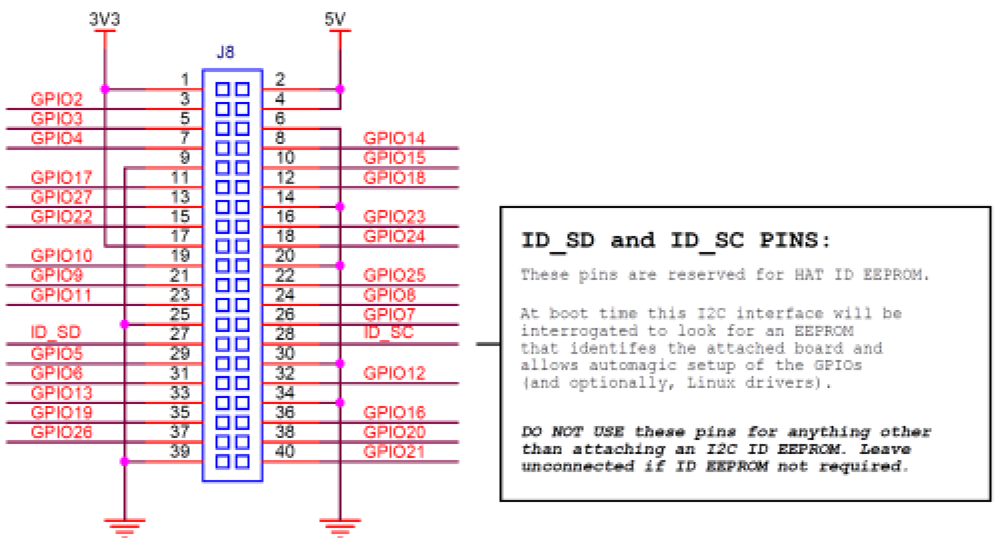


Figure 3: GPIO Connector Pinout

As well as being able to be used as straightforward software controlled input and output (with programmable pulls), GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI.

In addition to the standard peripheral options found on legacy Pis, extra I2C, UART and SPI peripherals have been added to the BCM2711 chip and are available as further mux options on the Pi4. This gives users much more flexibility when attaching add-on hardware as compared to older models.

5.1.2 GPIO Alternate Functions

GPIO	Default Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	SPI3_CE0_N	TXD2	SDA6
1	High	SCL0	SA4	DE	SPI3_MISO	RXD2	SCL6
2	High	SDA1	SA3	LCD_VSYNC	SPI3_MOSI	CTS2	SDA3
3	High	SCL1	SA2	LCD_HSYNC	SPI3_SCLK	RTS2	SCL3
4	High	GPCLK0	SA1	DPI_D0	SPI4_CE0_N	TXD3	SDA3
5	High	GPCLK1	SA0	DPI_D1	SPI4_MISO	RXD3	SCL3
6	High	GPCLK2	SOE_N	DPI_D2	SPI4_MOSI	CTS3	SDA4
7	High	SPI0_CE1_N	SWE_N	DPI_D3	SPI4_SCLK	RTS3	SCL4
8	High	SPI0_CE0_N	SD0	DPI_D4	-	TXD4	SDA4
9	Low	SPI0_MISO	SD1	DPI_D5	-	RXD4	SCL4
10	Low	SPI0_MOSI	SD2	DPI_D6	-	CTS4	SDA5
11	Low	SPI0_SCLK	SD3	DPI_D7	-	RTS4	SCL5
12	Low	PWM0	SD4	DPI_D8	SPI5_CE0_N	TXD5	SDA5
13	Low	PWM1	SD5	DPI_D9	SPI5_MISO	RXD5	SCL5
14	Low	TXD0	SD6	DPI_D10	SPI5_MOSI	CTS5	TXD1
15	Low	RXD0	SD7	DPI_D11	SPI5_SCLK	RTS5	RXD1
16	Low	FL0	SD8	DPI_D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPI_D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPI_D14	SPI6_CE0_N	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPI_D15	SPI6_MISO	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPI_D16	SPI6_MOSI	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPI_D17	SPI6_SCLK	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPI_D18	SD1_CLK	ARM_TRST	SDA6
23	Low	SD0_CMD	SD15	DPI_D19	SD1_CMD	ARM_RTCK	SCL6
24	Low	SD0_DAT0	SD16	DPI_D20	SD1_DAT0	ARM_TDO	SPI3_CE1_N
25	Low	SD0_DAT1	SD17	DPI_D21	SD1_DAT1	ARM_TCK	SPI4_CE1_N
26	Low	SD0_DAT2	TE0	DPI_D22	SD1_DAT2	ARM_TDI	SPI5_CE1_N
27	Low	SD0_DAT3	TE1	DPI_D23	SD1_DAT3	ARM_TMS	SPI6_CE1_N

Table 5: Raspberry Pi 4 GPIO Alternate Functions

Table 5 details the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the BCM2711 Peripherals Specification document which can be downloaded from the hardware documentation section of the website.

5.1.3 Display Parallel Interface (DPI)

A standard parallel RGB (DPI) interface is available the GPIOs. This up-to-24-bit parallel interface can support a secondary display.

5.1.4 SD/SDIO Interface

The Pi4B has a dedicated SD card socket which suports 1.8V, DDR50 mode (at a peak bandwidth of 50 Megabytes / sec). In addition, a legacy SDIO interface is available on the GPIO pins.

5.2 Camera and Display Interfaces

The Pi4B has 1x Raspberry Pi 2-lane MIPI CSI Camera and 1x Raspberry Pi 2-lane MIPI DSI Display connector. These connectors are backwards compatible with legacy Raspberry Pi boards, and support all of the available Raspberry Pi camera and display peripherals.

5.3 USB

The Pi4B has 2x USB2 and 2x USB3 type-A sockets. Downstream USB current is limited to approximately 1.1A in aggregate over the four sockets.

5.4 HDMI

The Pi4B has 2x micro-HDMI ports, both of which support CEC and HDMI 2.0 with resolutions up to 4Kp60.

5.5 Audio and Composite (TV Out)

The Pi4B supports near-CD-quality analogue audio output and composite TV-output via a 4-ring TRS 'A/V' jack.

The analog audio output can drive 32 Ohm headphones directly.

5.6 Temperature Range and Thermals

The recommended ambient operating temperature range is 0 to 50 degrees Celcius.

To reduce thermal output when idling or under light load, the Pi4B reduces the CPU clock speed and voltage. During heavier load the speed and voltage (and hence thermal output) are increased. The internal governor will throttle back both the CPU speed and voltage to make sure the CPU temperature never exceeds 85 degrees C.

The Pi4B will operate perfectly well without any extra cooling and is designed for sprint performance - expecting a light use case on average and ramping up the CPU speed when needed (e.g. when loading a webpage). If a user wishes to load the system continually or operate it at a high termperature at full performance, further cooling may be needed.

- **Arduino UNO data sheet**

2 Ratings

2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C (185°F)

NOTE: In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected.

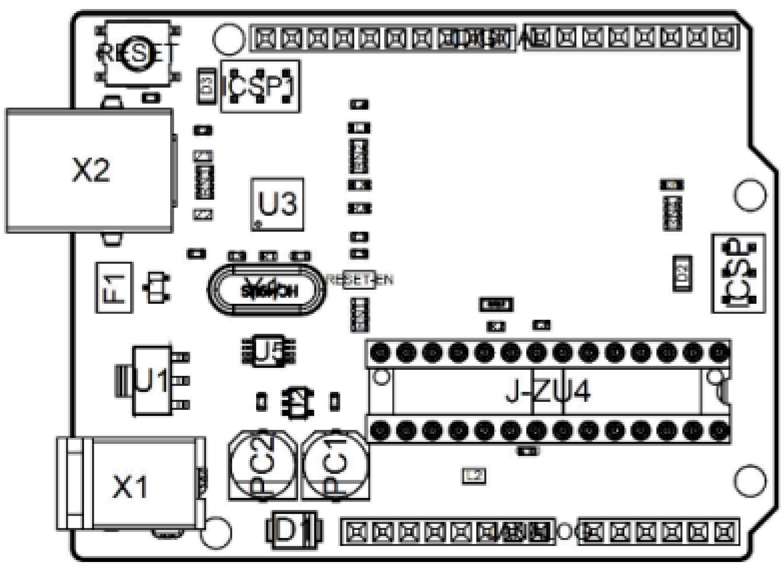
2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

3 Functional Overview

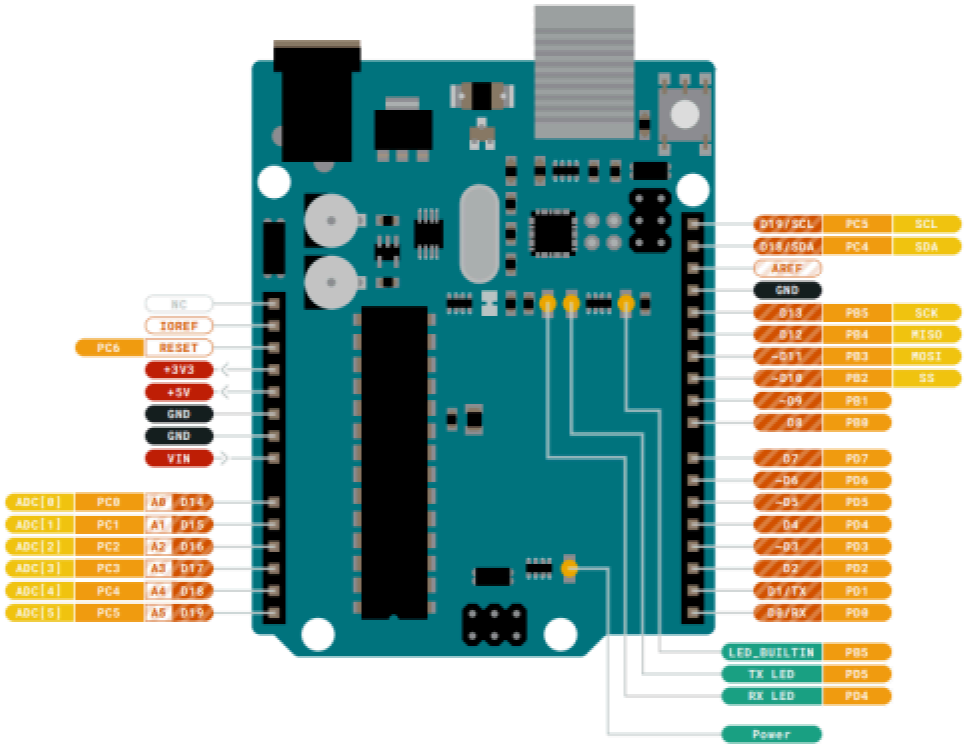
3.1 Board Topology

Top view



Board topology

5 Connector Pinouts



Pinout

5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

Peer Review

ECET 400 Notes A.Blinder

Peer Review Form

A. Document Being Reviewed: Design Specifications Document

B. Project Title: Sign Language Assistance

C. Names on Project: Matthew Frazier, SungWon Lee, Oluwadamilare Ajayi

D. Reviewer / date of Review: Seraphina Thorpe / 02/21/24

1. **Document Format**

a. Is there a Table of Contents? Y / N

b. Are pages numbered? Y / N

c. Are all required sections present? Y / N

2. **Spelling and Grammar**

a. Was spelling checked and correct (poor) 0 1 2 3 4 5 (perfect)

b. Sentences make sense and are readable (poor) 0 1 2 3 4 5 (very clear)

3. **Content**

a. Is the overall project clear in terms of:

What the product is supposed to do? Y / N

How it will be accomplished? Y / N

What the major components are? Y / N

b. Does the document accomplish its task in terms of:

Partition/Test Procedure document, is the project clearly subdivided? (poor) 0 1 2 3 4 5 (very clear)
and the test methodology clear? (poor) 0 1 2 3 4 5 (very clear)
and do the deliverables build on each other (not at all) 0 1 2 3 4 5 (very much)

c. Rate the document and project in terms of level:

Document	(low level) 0 1 2 3 4	5 (very professional)
Project	(low level) 0 1 2 3 4	5 (very professional)

4. **Specific Suggestions**

On the back of this sheet state:

a. **at least one** good and innovative aspect about this project
b. **at least four** aspects of the document needing improvement
c. **at least two** aspect of the project needing improvement

a) Project aims to help hearing impaired in a interactive Way not yet done

b) 1) Missing Sections

2) Table and figure orginization

3) Formatting inconsistencies

4) Better description of processes - Hardware and Software

c)

1) More defined scope

2) Use servo contioller block if possible