McMaster University

PROC TECH 4TR3: Final Report

Mini Desktop Robot

Section: C01

Supervisor: Dr. Zhen Gao

Student Names:

Deep Gupta – 400243809

Zhen Tang – 400199602

Aryan Sharma – 400209706

This final report contains our work for PROCTECH 4TR3: Capstone Design Project II and has not been submitted, in whole or in part, towards fulfilling requirements in any other course, either at McMaster University or elsewhere. I/We have read McMaster's academic integrity policy and understand that plagiarism, whether intentional or unintentional, will be dealt with under university policy and practice.

Abstract

The objective of this project is to create a mini-desktop robot that will be portable, safe, and customizable. The scope of this project involves 3D printing of the parts required for the assembly of the Hardware, developing the firmware (ElectronBot-FW & ServoDrive-fw), and software (SDK & C++). The scope of this project has been reduced compared to last semester as 2 group members were unable to take the course in fall 2023, with one new group member being added as seen in **Future Scope for Further Development.** As a result, the scope of objectives will now mainly be focused on the hardware creation of the robot and firmware integration of the robot with less of a focus on software development and a reduction in the overall features of the robot. In summary, the focus is on the body of the robot while the brain or advanced software upgrades/capabilities will be left to the previously mentioned 2 group members for further development from a software perspective.

This final report will cover the hardware, software, and firmware notes that have cumulated over the last 12 months as well as the final progress of the main deliverables of the project and the reduced scope. In addition, the design concepts and project ideation will be discussed in detail with a primary focus on hardware development and core features. Finally, the completed work for the project as of completion of this term will be displayed along with a discussion on the challenges and future next steps.

Contents

List of Figures

List of Tables

Introduction

The inspiration for this project stems from the EVE robot in WALL-E. The idea of this project is to enhance the user experience for a desktop owner with the accessory item of a Mini-Desktop robot connected to the PC via USB-C connection. The Mini-Desktop robot aims to be portable, safe, and customizable to use with 6 degrees of freedom and an LCD display screen acting the robot's face.

The major modules for this project include hardware, such as 3D printings, the designs of PCB and soldering related components for modifications; software including CAD, SDK design, C++ coding IDE; and firmware, such as the control board and servo drive. They are interconnected by using software to modify the firmware, and then using hardware to make the outer cases and cover the firmware inside. In addition, the final progress and overall final evaluation for the robot will be covered in this report.

Project Background

The challenge that our team decided to pursue was the challenge of creating a Mini-Desktop robot that can connect to the individual PC via USB-C connection to perform several types of tasks such as gesture recognition, voice recognition, pre-set demos, and operational movement for entertainment and potentially educational, purposes.

Objectives

The objectives for this Mini-Desktop robot have been adjusted to accommodate the reduced scope for this project during this semester with the loss of 2 team members who were focused on the software side of the project. Below are the main objectives for this project:

- Creation of a "to-scale" prototype mini-desktop robot with full motion capabilities
- Full hardware creation, integration, and testing of the Mini Desktop Robot from 3D-Printing the material with modifications to the CAD model and soldering hardware components such as the sensor board for modifications and overall integration for the robot.
- Program the body and motion; leave brain to future group for development.
- Basic software integration to control the body of the robot via servo motor control.
- Firmware integration for features such as voice recognition, gesture recognition and larger degree of freedom for movement
- User-based flexibility options to control Robot, i.e. manual or auto control with various methods as well as LCD screen with custom pictures/gifs/videos

Model Modifications

The steps below have been implemented during this semester for the model prototype:

1. Increase interior space to enhance the capacity to install hardware components with greater tolerance.

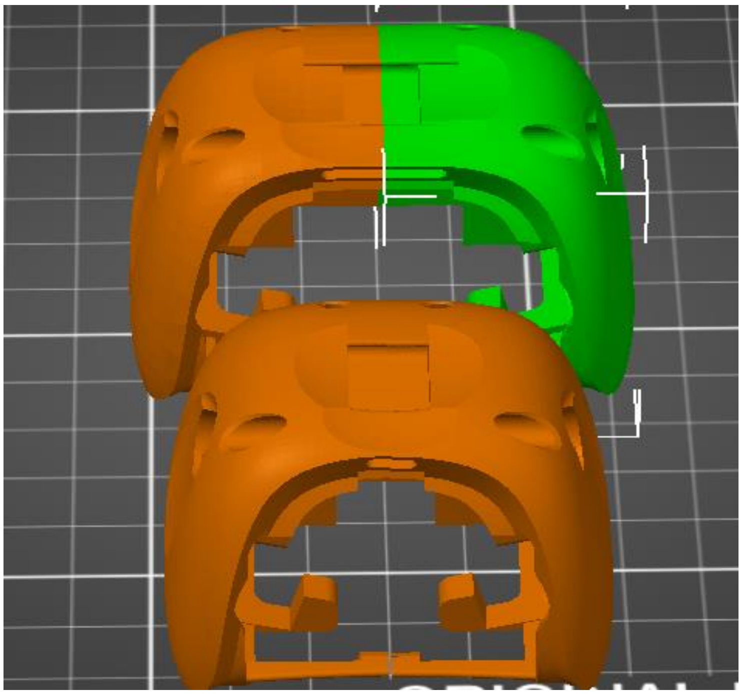


Figure 1 - Model Modifications Part 1

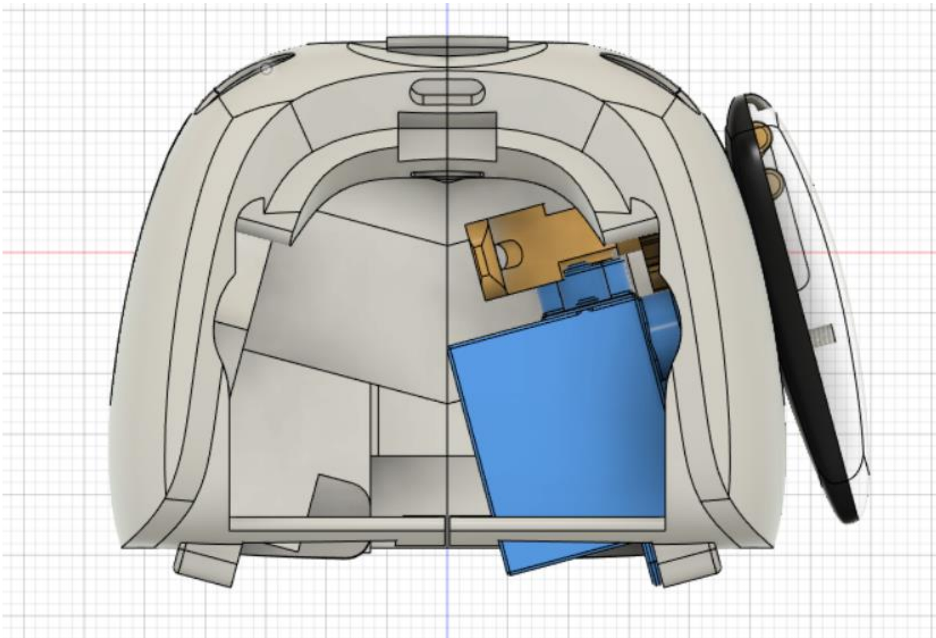


Figure 2 - Model Modifications Part 2

(Peng-Zhihui, *ElectronBot/enREADME.md at main · Peng-Zhihui/Electronbot*)

2. Add installation position for a servo rocker arm.

In the original design, the motor is directly attached to the bottom base cover as seen in **Figure 3 - Original Servo Rocker**

In the modified design, a servo rocker arm is added between the motor and bottom base cover as seen in **Figure 4 - Modified Servo Rocker Arm**

Potential benefits include:

- ✓ Increased precision as a servo rocker arm can provide better control over the movement of the motor.
- ✓ Reduced vibration as some of the vibration can be absorbed by a rocker arm.
- ✓ Increased lifespan by reducing the stress on the motor.

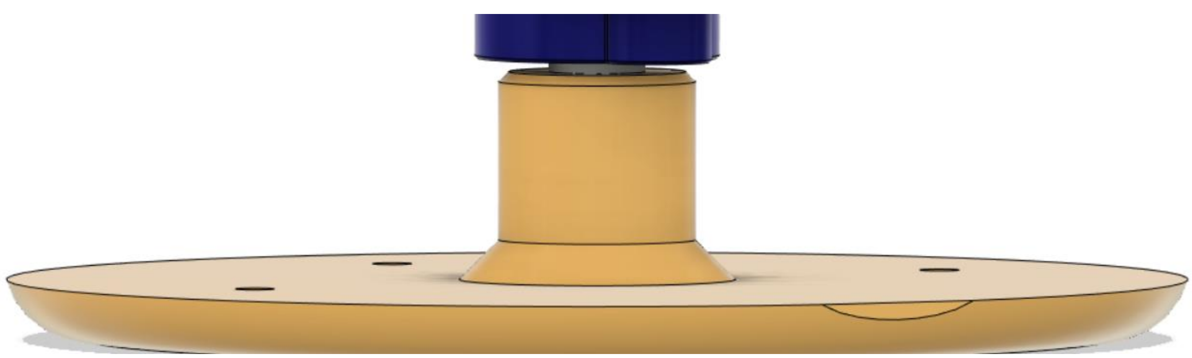


Figure 3 - Original Servo Rocker

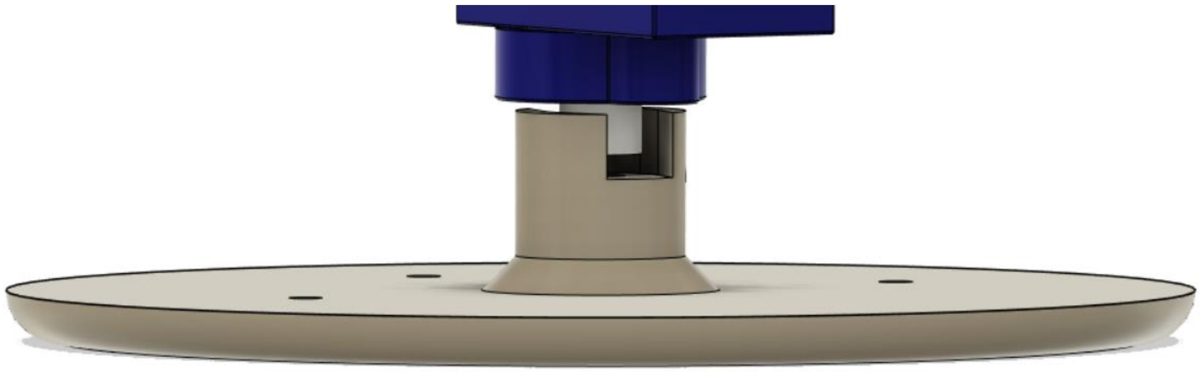


Figure 4 - Modified Servo Rocker Arm

Figure 5 - Servo Rocker Arm Alternatives

Servo Motor to hybrid of a stepper/servo motor which connects to the servo motor for motion control achieves the following:

- ✓ Allows for easier control; uses closed loop control.
- ✓ Servo motor; closed loop control
- ✓ After modification with closed loop control, provides more control, accuracy, and includes a potentiometer inside which provides feedback.
- ✓ Connects to Servo motor.
- ✓ Program the body and motion; leave brain to future group for development.

Hardware

Figure 6 - Base 3D Printed Model

CAD Model Explanation

To create the model sleek and resilient outer body, we experimented with PLA filament, hoping for a cost-effective solution. However, the quality was regrettably subpar. The layers were uneven, and it lacked finesse. This setback led us to explore alternative materials. Eventually, UV resin was utilized for its outer shell. UV resin not only provides a sturdy and durable framework for the robot, but its unique properties allowed for intricate detailing and a flawless finish. The hands possess two-degree freedom, while the body and head each exhibit one degree of freedom. The arm's forward/backward rotation is operated through gear rotation, while the rotation in the opposite direction is driven by a separate servo and the metal T-shaped rod, as shown in **Figure 9 - Internal Part of Model**. The same concept is applied to the body and head, utilizing a single servo motor for their movement. The microcontroller chosen for this application is the STM32, it will be responsible for overseeing LCD operation, servo control, and communication tasks as shown in **Figure 9 - Internal Part of Model**. A sensor board integrated of a camera and infrared gesture sensor is placed at the chest of the robot for capturing the surrounding and to recognize gestures as shown in **Figure 7 - Fusion 360 CAD Model**.

Figure 7 - Fusion 360 CAD Model

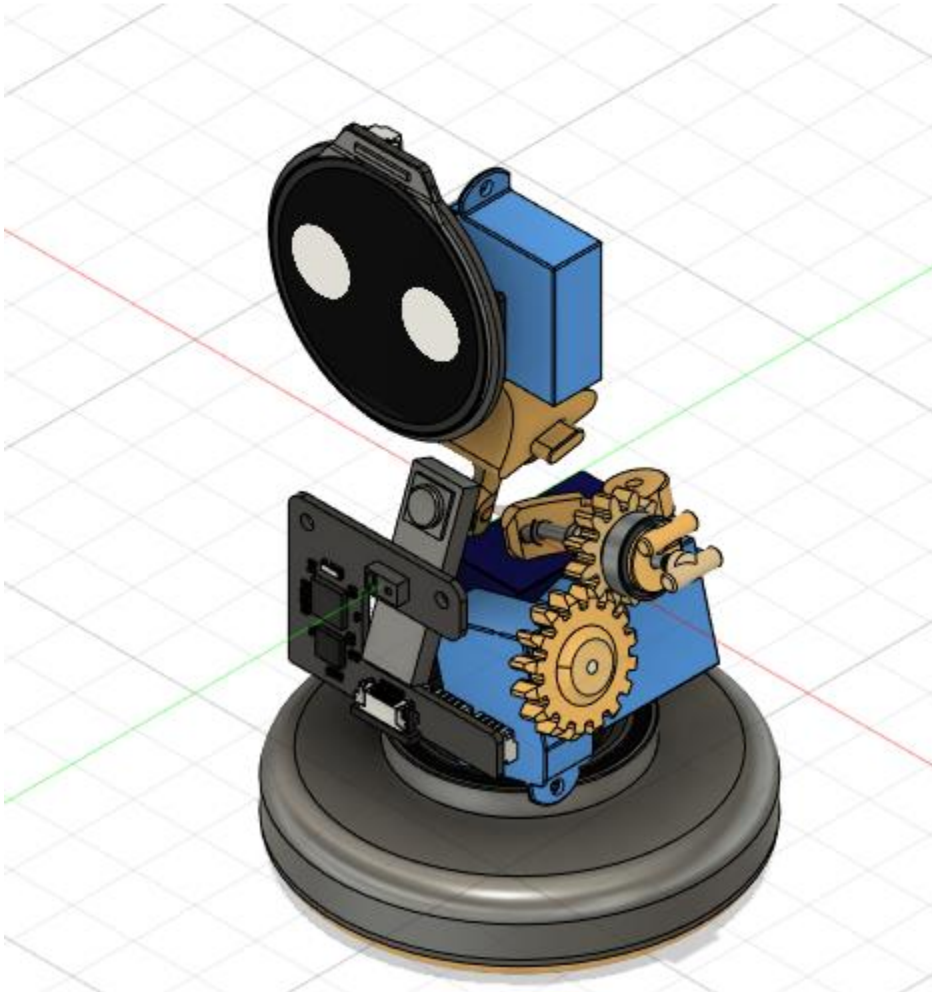


Figure 8 - External Part of Model

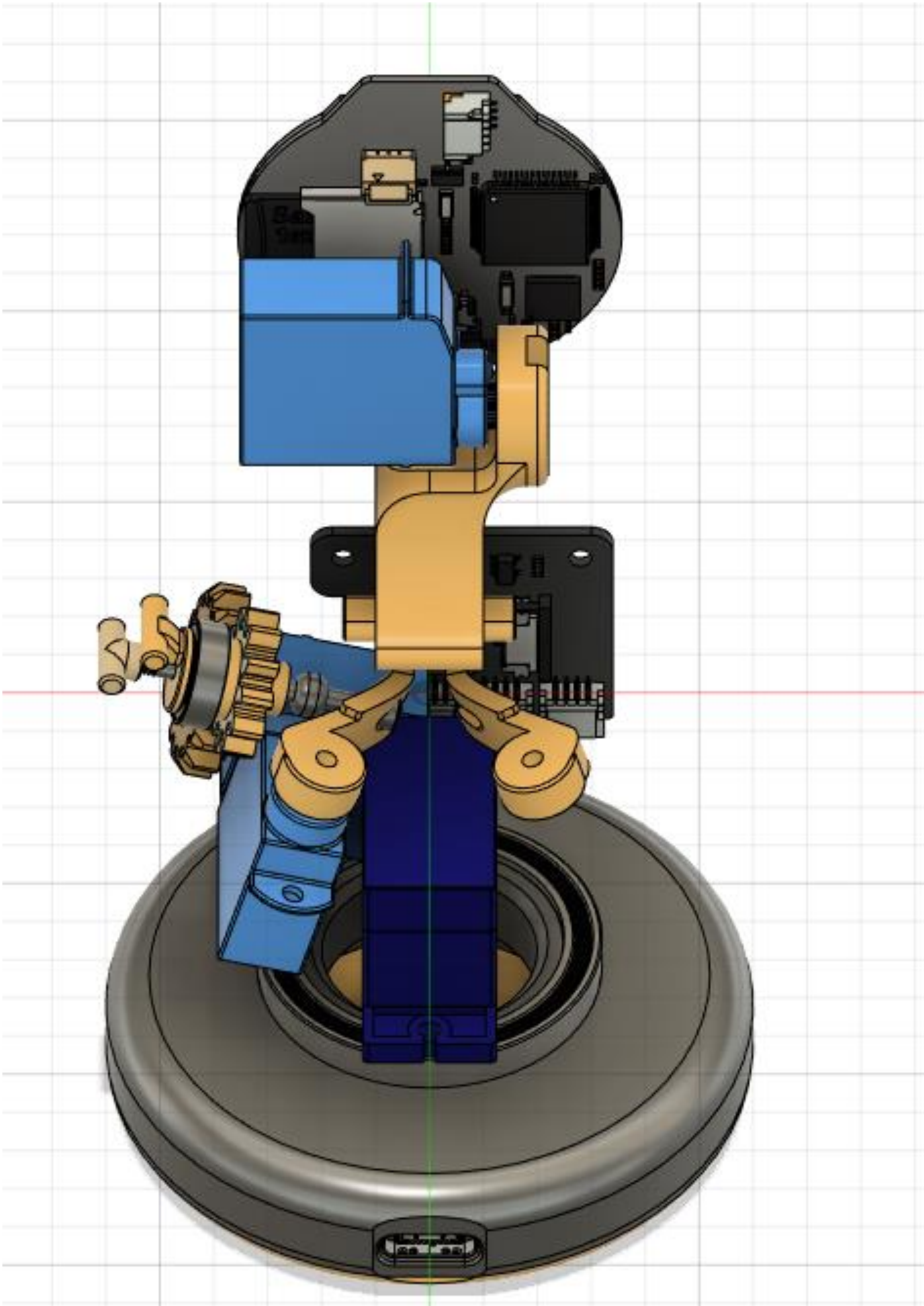


Figure 9 - Internal Part of Model

Hardware Parts Purpose and Capabilities

Gesture Sensor PAJ7620U2 -

Figure 10 - Gesture Sensor

- Recognizes 9 gestures, including: move up, move down, move right, move left with hand movement (*PAJ7620U2 gesture sensor*)
- Features (*PAJ7620U2 gesture sensor*):
  - Recognizes 9 gestures, supports gesture interrupt output (*PAJ7620U2 gesture sensor*)
  - Embedded infrared LED and optical lens
  - Ability to work in different environments, such as dark or low light rooms (*PAJ7620U2 gesture sensor*)
  - 12C Interface, requires two signal pins to control
  - Onboard Voltage Translator – 3.3/5V Logic Level
- Specifications (*PAJ7620U2 gesture sensor*):
  - 5cm - 15 cm distance
  - Up, down, left, right, forward, backward, clockwise, anticlockwise, shake.
  - Recognition rate: 240Hz
  - Recognition angle: 60 degrees, diagonal
  - Ambient Lighting Immunity: <100K Lux
    - Lux – SI unit for illuminance
  - Dimensions: 20 mm (about 0.79 in) x 20mm (about 0.79 in)
  - Mounting hole size:2.0 mm (about 0.08 in)
- How to use (*PAJ7620U2 gesture sensor*):
  - Can utilize Raspberry PI
  - Python

- Relation to Project (*PAJ7620U2 gesture sensor*):
  - Used for gesture recognition feature of robot.

- Hardware Connection (*PAJ7620U2 gesture sensor*)

Hardware connection

The colors of wires may be different, please connect it according to the silk screen printing.

Gesture Sensor	Raspberry Pi (Board)	Raspberry Pi(BCM2835)
VCC	3.3V	3.3V
GND	GND	GND
SDA	3	P2
SCL	5	P3

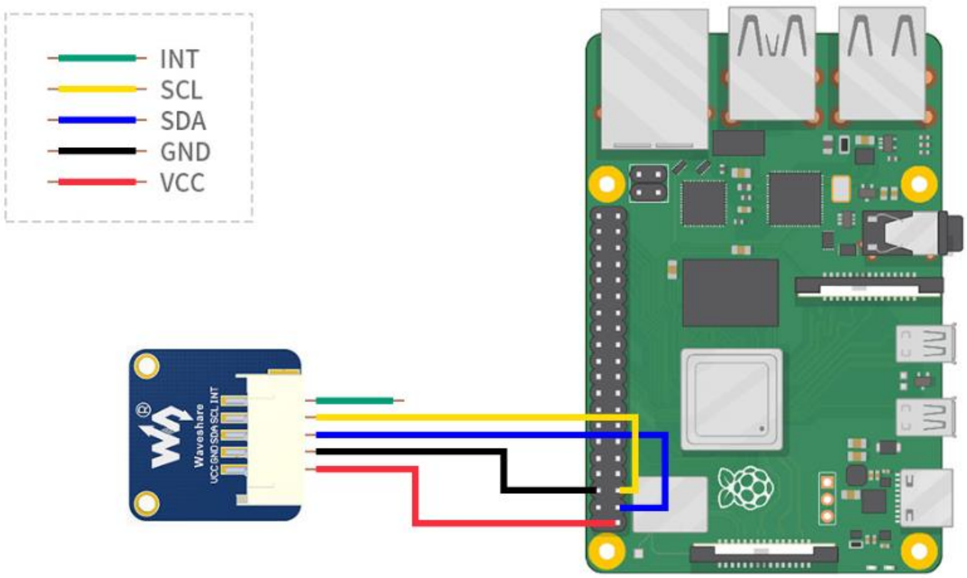


Figure 11 - Gesture Sensor Diagram

Six-Axis (Gyro + Accelerometer) MEMS Motion Tracking Device MPU-6050

- Features (*MPU-6050: TDK*):
  - Designed for Low Power, Low Cost, and High Performance
  - Combines a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die, together with an onboard Digital Motion Processor™ (DMP™), which processes complex 6-axis Motion Fusion algorithms (*MPU-6050: TDK*)
- Definitions (*MPU-6050: TDK*):
  - An accelerometer measures linear acceleration along one or several axes
  - A gyroscope measures angular velocity along one or several axes
- Specifications (*MPU-6050: TDK*):
  - Used for precision tracking of both fast and slow motions, the parts feature a user-programmable gyro full-scale range of ±250, ±500, ±1000, and ±2000 °/sec (dps)

- The device contains a user-programmable accelerometer full-scale range of ±2g, ±4g, ±8g, and ±16g. Additional features include an embedded temperature sensor and an on-chip oscillator with ±1% variation over the operating temperature range

- Relation to Project(*MPU-6050: TDK*):
  - Used for precision tracking as a part of gesture recognition.
  - Secondary Objective: Can be used as a temperature sensor as a part of the Home Assistant aspect of the robot.

- Schematic Diagram (*MPU-6050: TDK*):

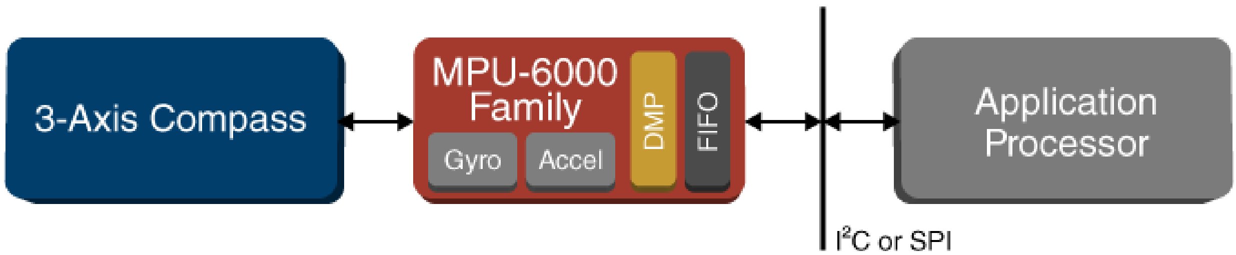


Figure 12 - Six-Axis MEMS Motion Tracking Device

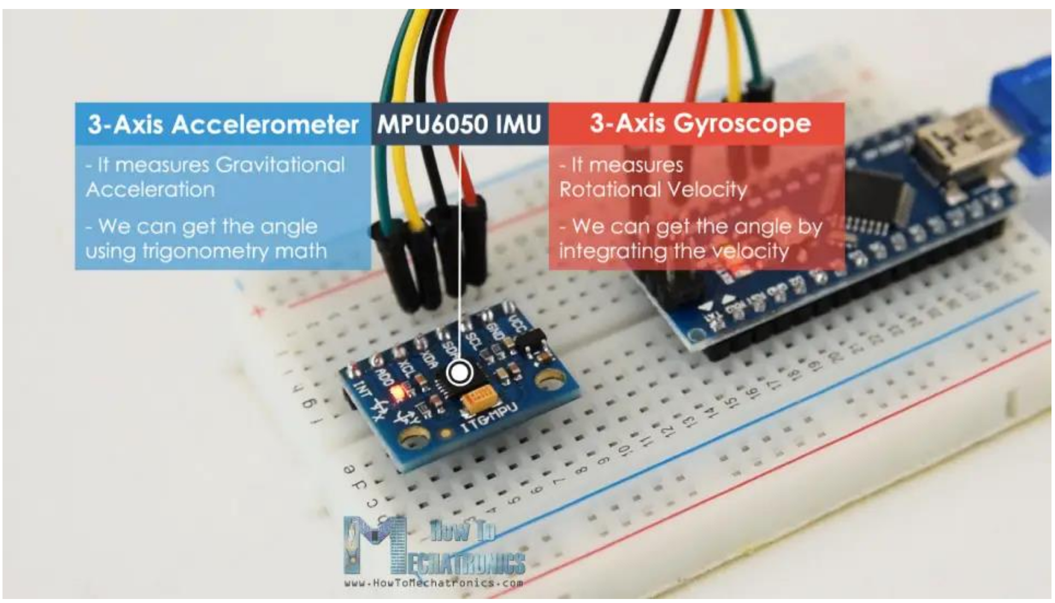


Figure 13 - Six-Axis MEMS Motion Tracking Device Information

ST-LINK V2

The ST-LINK V2 works as an in-circuit debugger and programmer for the stm8 and stm32 microcontrollers as an in-circuit debugger and programmer for the stm8 and stm32 microcontrollers. The single-wire interface module (SWIM) and JTAG/serial wire debugging (SWD) interfaces are used to communicate with any STM32 microcontroller located on the application board. In addition to providing the same functionalities as the ST-LINK/V2, the ST-LINK/V2-ISOL features digital isolation between the PC and the target application board which for the projects case is the main sensor board.

Features:

- 5 V power supplied by a USB connector.
- USB 2.0 full-speed compatible interface
- USB Type-A to Mini-B cable provided.

- **SWIM specific features**:

- 1.65 V to 5.5 V application voltage support on the SWIM interface.
- SWIM low-speed and high-speed modes support .
- SWIM programming speed rates: 9.7 kbyte/s in low-speed, 12.8 kbyte/s in high-speed
- SWIM cable for connection to an application with an ERNI standard connector Vertical connector reference: 284697 or 214017Horizontal connector reference: 214012.
- SWIM cable for connection to application with pin headers or 2.54 mm pitch connector.

- **JTAG/serial wire debug (SWD) specific features:**

- 1.65 V to 3.6 V application voltage support on the JTAG/SWD interface and 5 V inputs.
- JTAG cable for connection to a standard JTAG 20-pin 2.54 mm pitch connector.
- SWD and serial wire viewer (SWV) communication support
- Direct firmware update support (DFU).
- Status LED blinking during communication with the PC.
- Operating temperature from 0 °C to 50 °C.
- Withstands voltages of up to 1000 $V_{rm.}$
- STM32 applications use the USB full-speed interface to communicate with the STM32CubeIDE software tool or other third-party applications.

Figure 14 -ST Link V2

STM32 Microcontroller

STM32 microcontrollers offer many serial and parallel communication peripherals that can be interfaced with all kinds of electronic components including sensors, displays, cameras, motors, etc. All STM variants come with internal memory and RAM.

Apart from the software tools, an In-Circuit Serial Programmer (ICSP) is required to program and test the code on the actual microcontroller.

- Code framework already exists on GitHub – intention is to program functions that control overall motion of the robot; which leaves the brain and smart features for future development. Discussed more in **Future Scope for Further Development**.

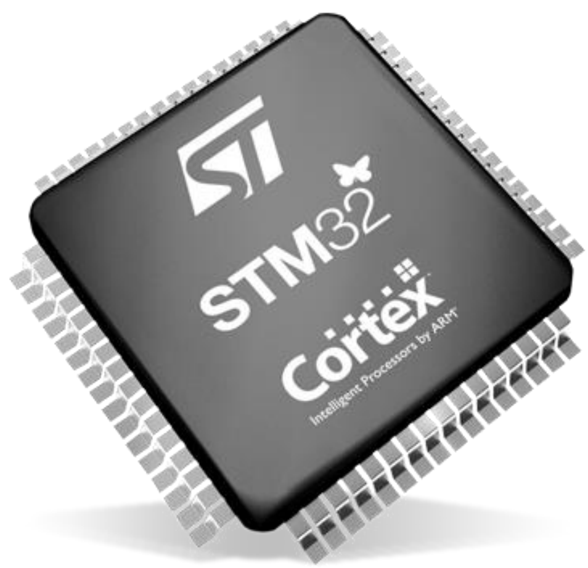


Figure 15 - STM32 Microcontroller

LP2992IM5-3.3 Voltage Regulator

A voltage regulator usually takes in higher input voltage and emits a lower, more stable output voltage. Their secondary use is also to protect the circuit against voltage spikes that can potentially damage/fry them. The power supply unit of an electronic device converts incoming power into the desired type (AC-DC or DC-AC) and desired voltage/current characteristics. A voltage regulator is a component of the power supply unit that ensures a steady constant voltage supply through all operational conditions. It regulates voltage during power fluctuations and variations in loads. It can regulate AC as well as DC voltages.

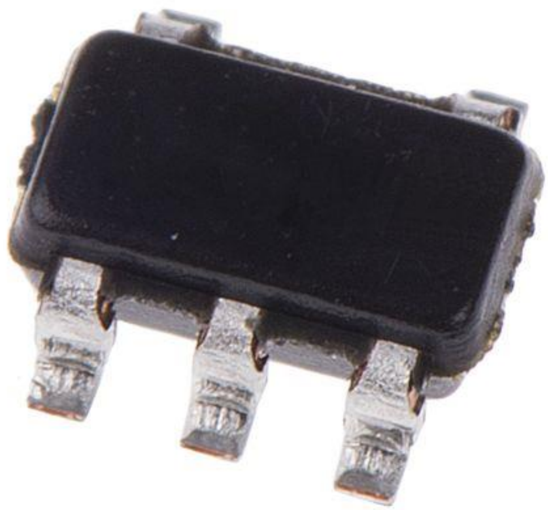


Figure 16 - Voltage Regulator

GMI4015P-2C-66db Electret Condenser Microphone

A type of microphone that use a thin diaphragm made of electret material to convert sound waves into electrical signals. It is highly sensitive and low in self-noise which makes it ideal for audio capture.

Key Features:

- Sensitivity: Its ability to converts soundwave into an electrical signal. It indicates how responsive the microphone is to sound.
- Operating Voltage: between 1.5V and 10 V

Figure 17 - Electret Condenser Microphone

PCM2912APJTR Stereo Audio Codec

It offers efficient and precise audio processing while maintaining excellent audio quality. It is a crucial component in audio systems due to its audio encoding and decoding capabilities. In addition, it provides a convenient solution for capturing, processing, and reproduction of input audio signals for USB-based systems.

Key features:

- Support a USB Interface
- Audio Codec Functionality: Designed for USB-based digital audio applications and ensures reliable and high-quality audio performance.
- Noise Reduction: The integrated circuit helps deliver clear and high-fidelity audio output.
- Voltage Regulation: The internal voltage regulator provides stable power for the device's operation.
- High Power efficiency
- Cost-effectiveness: Its low cost helps us keep the overall cost of our project within an affordable budget.

SC8002B Audio Power Amplifier

It receives audio signals from audio codecs and amplifies the low-power signal to a level suitable for driving our motors.

Key features:

- Low distortion and high-fidelity
- Power Efficiency
- Voltage supply: It supports voltage supply between 2V and 5V which is suitable for USB-connected devices.

Design Progress and Project Plan

Step by Step Plan after Ideation Current Progress

Green = Completed

Yellow = In progress/planned

Red = Out of scope/will not be completed

- 3D print components.
- Purchase all required materials.
- Solder the TYPE-C interface board and the sensor board together.
- Test the functions of the TYPE-C interface board and the sensor board.
- Solder the main board.
- Burn the firmware, and test if the computer can detect the CP2102 and a COM port.
- Install driver for the COM port and install lib-USB.
- Mainboard screen.
- Upload the screen test firmware (smiling face test firmware).
- Run the screen test special upper computer.
- Burn the servo control chip firmware (the firmware for F030 and F042 are different) into the TSSOP20 burning socket.
- Weld the servo driver board and the 4P connection wire (if not using the burning socket, you need to weld the burning wire and remove it after burning).
- Modify and solder the servo.
- Edit the main control firmware in CLION, add servo ID burning code, and burn it to the main control of the main board (there is a slight difference in operation between F030 and F042).
- Test all six servos to ensure that they can rotate 180 degrees in both positive and negative directions.
- Burn the firmware to the main control.

3D Printed Model Details

- Ensured all components are printed accurately.

Figure 18 - Empty 3D Printed CAD Model

Soldered PCBs:

- Sensor Board
- Central Control Board
- Servo Drive (6)

Modified Motor (4.7g/9g):

- Soldered the servo drive chip.
- Replaced the original control chip of the servo.

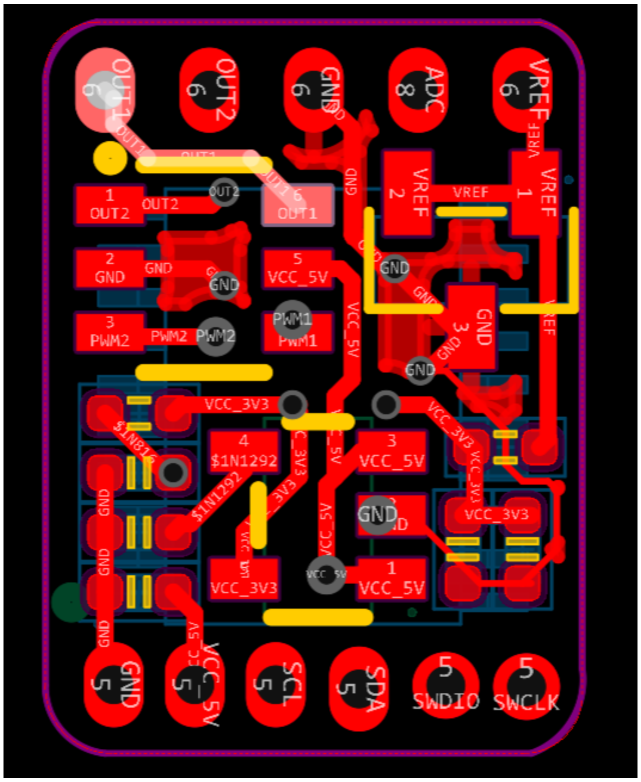


Figure 19 - Front end of PCB

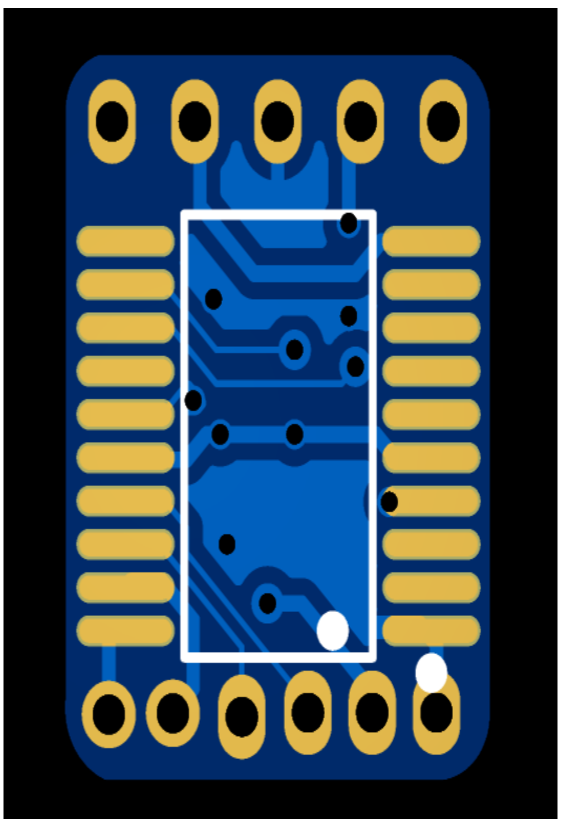


Figure 20 - Back End of PCB

Voice control PCB

- Hardware Selection
  - GMI4015P Electret Condenser Microphone
  - PCM2912APJTR IC STEREO AUDIO CODEC
  - SC8002B Audio Power Amplifier
- System Design
  - Microphone:
    - It converts sound waves into electrical signals. The output is a low-level analog signal.
  - Audio Codec:
    - It includes an analog to digital converter. The analog signal from the microphone is converted into a digital format.
    - Perform digital signals processing tasks include filtering, equalization and compression.
  - Microcontroller:
    - It interfaces with the audio codec to handle digital audio signals
  - Audio Power Amplifier:
    - Receive a low-level analog input from codec and amplifies it to a sufficient level to drive motor drivers.

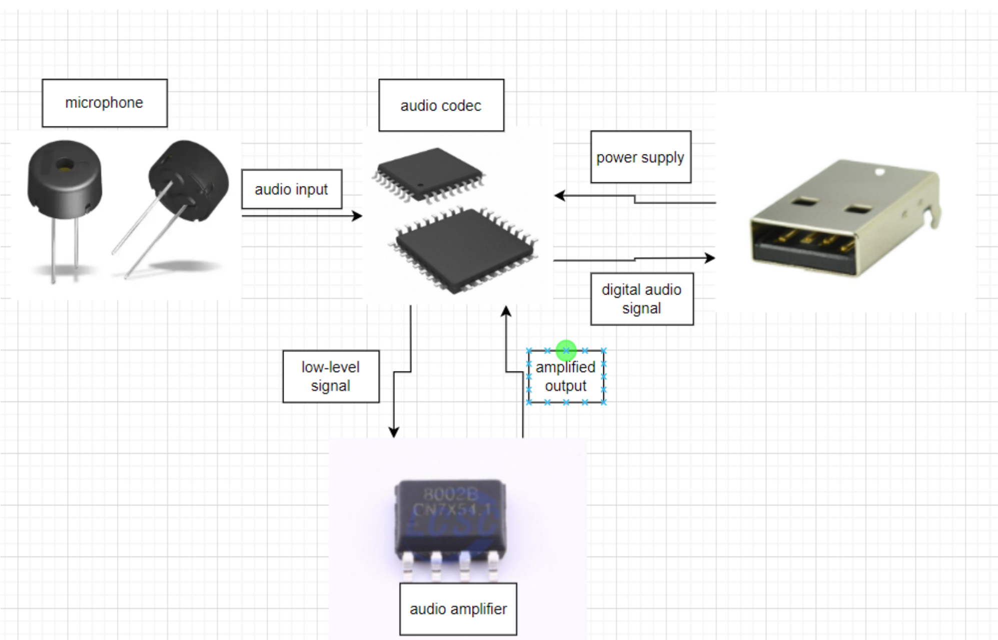


Figure 21 - Audio Schematic Flowchart

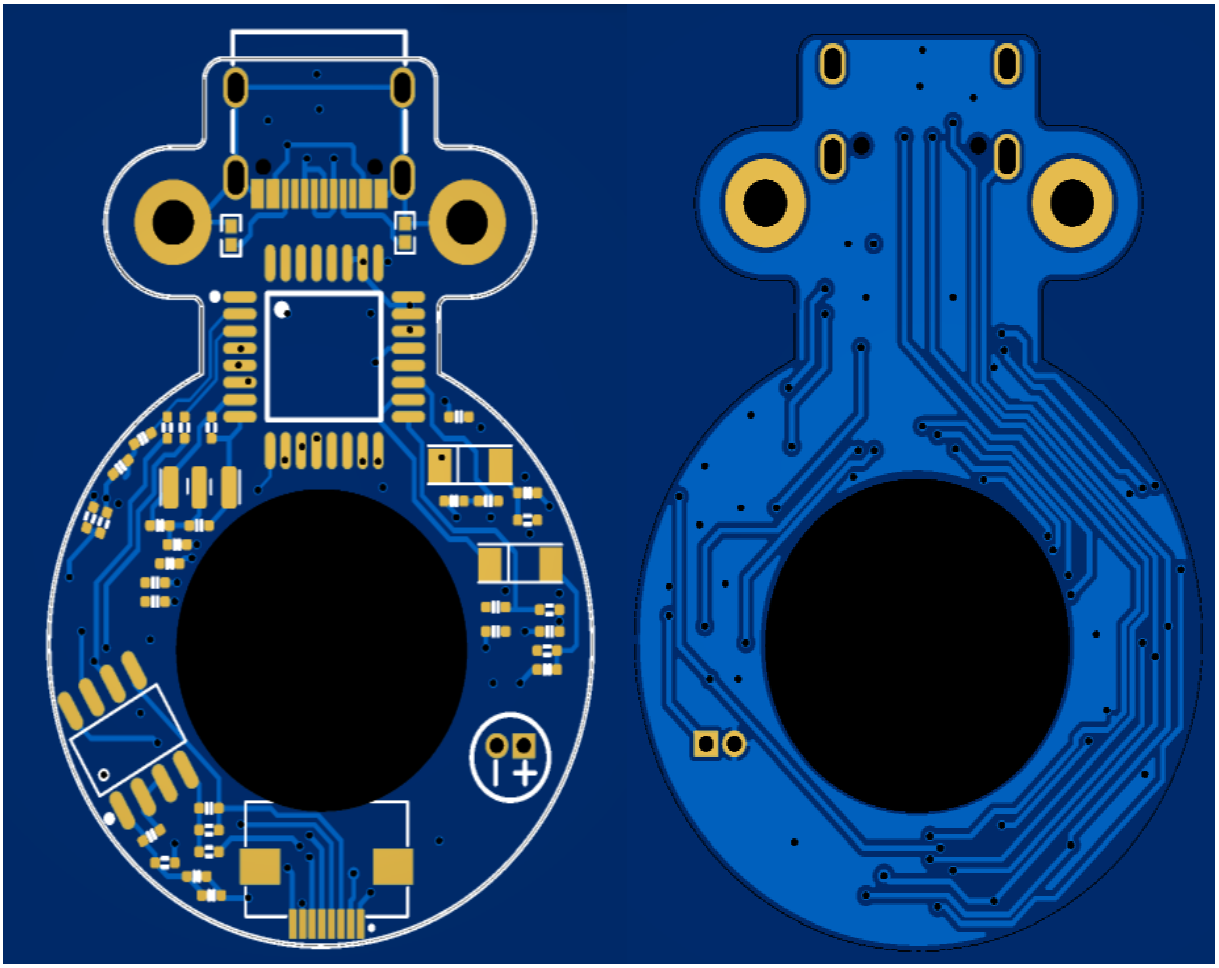


Figure 22 - PCB Circuit Design

Project Plan

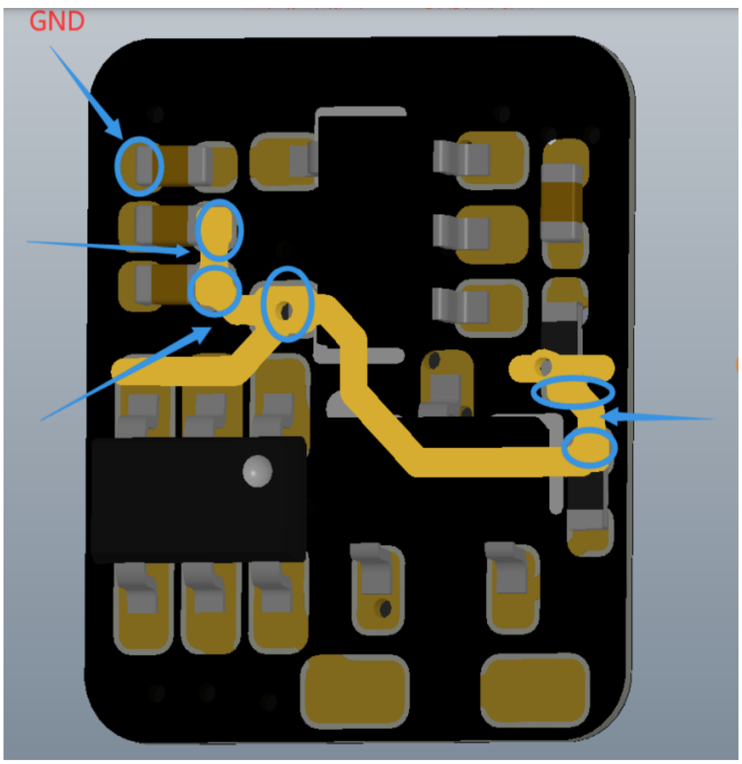
Deliverable	Activity/task	Task owner	Start date	Due date	Notes
1. Hardware Integration	1.1 Finalize Design (omit this step if the design is complete)	Deep, Zhen, Aryan			Done
	1.2 Purchase Material	Zhen			Done
	1.3 Integrate PCB (PCM2912APJTR), amplifier (SC8002B), microphone (GMI4015P-2C-66db) into system	Zhen	9/07/20230	9/14/2023	Done
	1.4 Sensor and Parts Research/Documentation	Deep	9/14/2023	10/20/2023	Done
	1.5 Servo Motor to hybrid of a stepper/servo motor which connects to the servo motor for motion control	Zhen, Aryan	9/14/2023	10/05/2023	Done
	1.6 Sensor Board Integration into System	Deep, Zhen, Aryan	9/7/2023	10/20/2023	Done
	1.7 Test the motion of the arm	Deep, Zhen, Aryan	9/14/2023	10/20/2023	Done
	1.8 Test the Whole System	Deep, Zhen, Aryan	9/14/2023	11/20/2023	Done
	1.9 Wire each sensor to the controller and confirm functionality	Deep, Zhen, Aryan	9/14/2023	10/20/2023	Done
2.Software Integration	2.1 Program ST-LINK V2 (device that burns code into controller)	Deep, Zhen, Aryan	10/05/2023	11/15/2023	Done
	2.2 Test the control system program for software	Deep, Zhen, Aryan	11/15/2023	12/01/2023	Done
3. Firmware Integration	3.1 Download Firmware to the controller, motors, and test firmware	Deep, Zhen, Aryan	10/01/2023	11/15/2023	Done
	3.2 Burn test firmware into the LCD screen to display a smiley face	Zhen	TBD	TBD	Done
	3.3 In CLION, edit the main control firmware, add servo ID	Zhen	TBD	TBD	Out of Scope

	programming code, and then burn it onto the mainboard controller.				
4. Documentation and presentations	4.1 Documentation for Reports	Deep, Zhen, Aryan	9/21/2023	Midterm Report: 10/19/2023 Final Report: December 7, 2023	Done
	4.2 Presentation Preparations	Deep, Zhen, Aryan	9/21/2023	Midterm Presentation: Oct 26, 2023 Final Presentation: December 7, 2023	Done

Table 1- Project Plan

Hardware Validation:

1. Motor drive chip



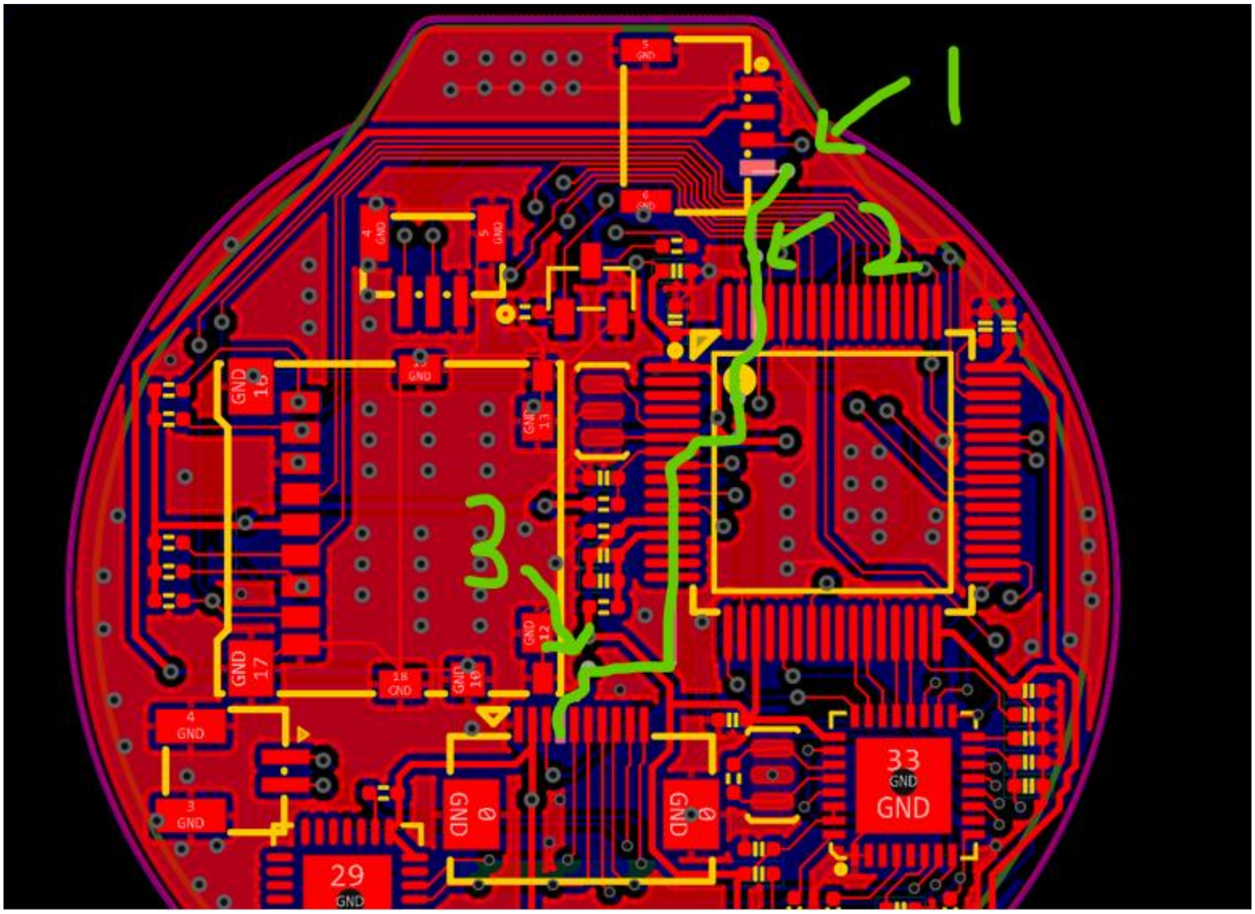
The servo driver chip needs to monitor 3 voltages:

5V: Board power supply, and the power supply for motor driving, a minimum of 4.5V is acceptable.

3.3V: 32 power supply, a deviation of 0.2 is acceptable

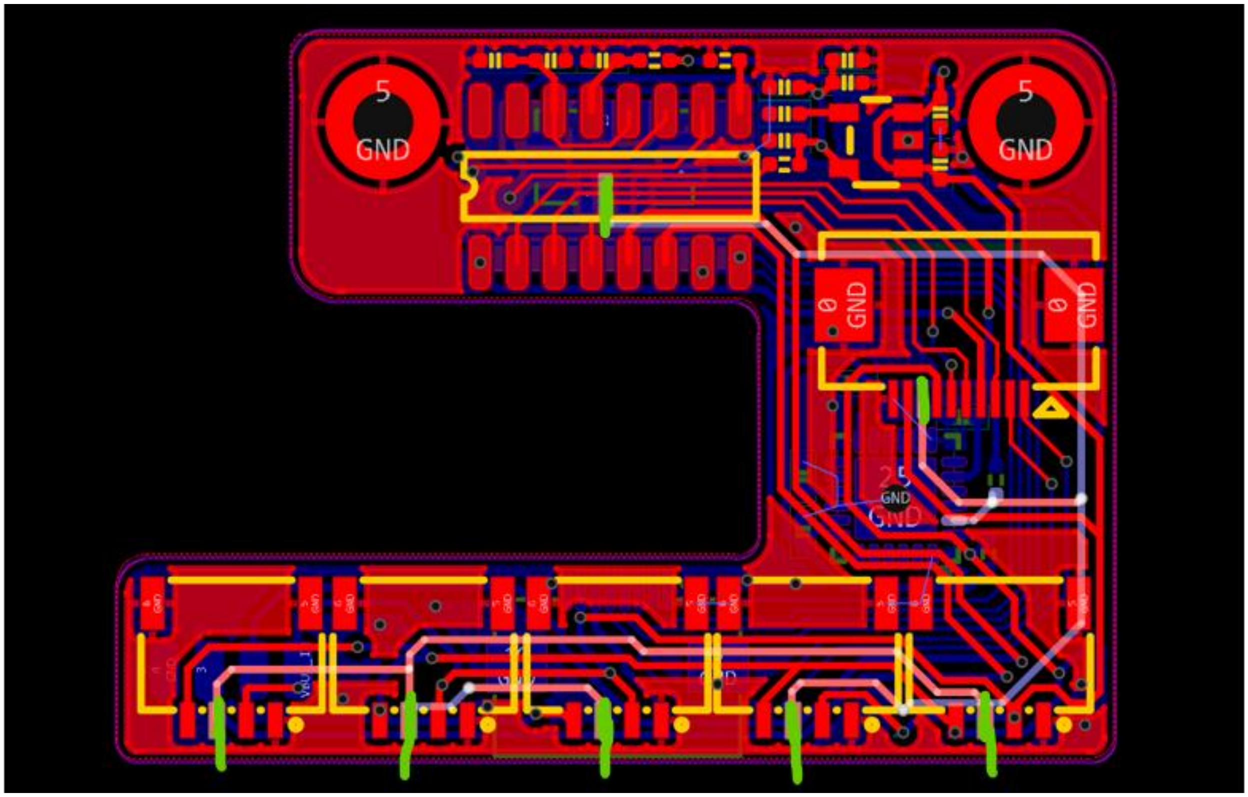
Central Control Board:

I2C communication path check:



Sensor Board:

Sensor Power Supply Check:



Problems Encountered:

1. Issues with the communication between the programmer and the 32-bit chip

Possible causes:

- connection error or loose programming cable
- error in soldering of the programming socket
- Defect in the PCB itself (check voltage in the communication path)
- Short circuit in the communication path

Solutions:

- Check the connections.
- Re-solder the programming socket
- add solder at the break.

2. low power supply for the stm32 chip

Possible causes:

- no power supply
- Poor soldering of 0R resistor

Solutions:

- Check if the mainboard is powered on and verify that the 5V to 3.3V conversion result is 3.3V
- Re-solder the 0R resistor

2. PC cannot detect the USB connection:

- Check if the HS8836A and the 3300 USB communication paths are connected.
- Check for short circuits between the HS8836A and the 3300 USB communication lines and adjacent pins. Inspect the 3300, two 8-pin sockets, and HS8836A.
- Use the oscilloscope (analog discover2) to check the waveform of the 24M crystal oscillator.

3. LCD screen Display Issue:

- Verify the installation of the driver if static snow is witnessed

4. Motor modifications issues:

Error:

- the motor keeps rotating.
- Motor is oscillating violently.

Solution:

- The potentiometer inside the motor is not functioning. Get another motor.
- The 3.3V and 5V voltage supply are falsely connected, reverse the connection.

Firmware

The Servo-Drive project is the code used to control the motor. This code includes ADC (Analog to Digital Converter) sampling, I2C slave communication and protocol analysis, motor control PWM output, and PID closed-loop algorithm implementation.

The steps taken for the ADC sampling configuration are:

1. ADC initialization:

Set the control registers to configure the ADC, including clock cycle, resolution, data alignment, scan mode, end of conversion (EOC) selection, low power modes.

```
/* ADC init function */
void MX_ADC_Init(void)
{

  ADC_ChannelConfTypeDef sConfig = {0};

  hadc.Instance = ADC1;
  hadc.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
  hadc.Init.Resolution = ADC_RESOLUTION_12B;
  hadc.Init.DataAlign = ADC_DATAALIGN_RIGHT;
  hadc.Init.ScanConvMode = ADC_SCAN_DIRECTION_FORWARD;
  hadc.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
  hadc.Init.LowPowerAutoWait = DISABLE;
  hadc.Init.LowPowerAutoPowerOff = DISABLE;
  hadc.Init.ContinuousConvMode = DISABLE;
  hadc.Init.DiscontinuousConvMode = DISABLE;
  hadc.Init.ExternalTrigConv = ADC_SOFTWARE_START;
  hadc.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_
  hadc.Init.DMAContinuousRequests = DISABLE;
  hadc.Init.Overrun = ADC_OVR_DATA_PRESERVED;
  if (HAL_ADC_Init(&hadc) != HAL_OK)
  {
    Error_Handler();
```

Figure 23- ADC Initialization

5. ADC channel configuration:

Configure the number, priority, and sampling rate at which the ADC samples the analog input signal.

```
/** Configure for the selected ADC regular channel to be converted.
*/
sConfig.Channel = ADC_CHANNEL_4;
sConfig.Rank = ADC_RANK_CHANNEL_NUMBER;
sConfig.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;
if (HAL_ADC_ConfigChannel(&hadc, &sConfig) != HAL_OK)
{
  Error_Handler();
}
/* USER CODE BEGIN ADC_Init 2 */

/* USER CODE END ADC_Init 2 */
```

Figure 24- ADC Channel Configuration

3. GPIO (General-Purpose Input/Output) configuration:

Configure the selected ADC channel as analog inputs.

```
void HAL_ADC_MspInit(ADC_HandleTypeDef* adcHandle)
{

  GPIO_InitTypeDef GPIO_InitStruct = {0};
  if(adcHandle->Instance==ADC1)
  {
  /* USER CODE BEGIN ADC1_MspInit 0 */

  /* USER CODE END ADC1_MspInit 0 */
    /* ADC1 clock enable */
    __HAL_RCC_ADC1_CLK_ENABLE();

    __HAL_RCC_GPIOA_CLK_ENABLE();
    /**ADC GPIO Configuration
    PA4     ------> ADC_IN4
    */
    GPIO_InitStruct.Pin = GPIO_PIN_4;
    GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
```

Figure 25 - GPIO Configuration

4. DMA initialization:

DMA (Direct Memory Access) channel is a hardware component that allows data to be transferred directly between peripherals and memory without the intervention of the CPUThis includes setting the direction, memory and peripheral increment modes, data alignment, transfer mode, and priority of the DMA channel.

```
/* ADC1 DMA Init */
/* ADC Init */
hdma_adc.Instance = DMA1_Channel1;
hdma_adc.Init.Direction = DMA_PERIPH_TO_MEMORY;
hdma_adc.Init.PeriphInc = DMA_PINC_DISABLE;
hdma_adc.Init.MemInc = DMA_MINC_ENABLE;
hdma_adc.Init.PeriphDataAlignment = DMA_PDATAALIGN_HALFWORD;
hdma_adc.Init.MemDataAlignment = DMA_MDATAALIGN_HALFWORD;
hdma_adc.Init.Mode = DMA_CIRCULAR;
hdma_adc.Init.Priority = DMA_PRIORITY_LOW;
if (HAL_DMA_Init(&hdma_adc) != HAL_OK)
{
  Error_Handler();
}
```

Figure 26 - DMA Initialization

5. ADC interrupt configuration

A software mechanism that allows the CPU to be notified when an ADC conversion is complete.

```
    /* ADC1 interrupt Init */
    HAL_NVIC_SetPriority(ADC1_IRQn, 1, 0);
    HAL_NVIC_EnableIRQ(ADC1_IRQn);
  /* USER CODE BEGIN ADC1_MspInit 1 */

  /* USER CODE END ADC1_MspInit 1 */
  }
}

void HAL_ADC_MspDeInit(ADC_HandleTypeDef* adcHandle)
{

  if(adcHandle->Instance==ADC1)
  {
  /* USER CODE BEGIN ADC1_MspDeInit 0 */

  /* USER CODE END ADC1_MspDeInit 0 */
    /* Peripheral clock disable */
    __HAL_RCC_ADC1_CLK_DISABLE();

    /**ADC GPIO Configuration
    PA4     ------> ADC_IN4
    */
    HAL_GPIO_DeInit(GPIOA, GPIO_PIN_4);

    /* ADC1 DMA DeInit */
    HAL_DMA_DeInit(adcHandle->DMA_Handle);

    /* ADC1 interrupt Deinit */
    HAL_NVIC_DisableIRQ(ADC1_IRQn);
  /* USER CODE BEGIN ADC1_MspDeInit 1 */

  /* USER CODE END ADC1_MspDeInit 1 */
  }
```

Figure 27 - ADC Interrupt

The steps taken for the I2C configuration are:

1. Create the I2C1 initialization function that sets up the I2C1 peripheral with the desired configuration:
    i) Set the I2C instance, timing, addressing mode, dual address mode, and other options.
    ii) Initialize the I2C peripheral with the configuration.
    iii) Configure the analog filter and digital filter by the calling Analog Filter and Digital Filter.
2. Configure the I2C1 peripheral's GPIOs, clock, and interrupt:
    i) Enable the GPIO clock and configure the SCL and SDA pins.
    ii) Enable the I2C1 clock.
    iii) Set the I2C1 interrupt priority and enable the interrupt.
3. Reinitialize the I2C1 peripheral's GPIOs, clock, and interrupt.
4. Create a user defined I2C initialization function that allows setting a custom I2C address.

Software:

General frame – SDK design

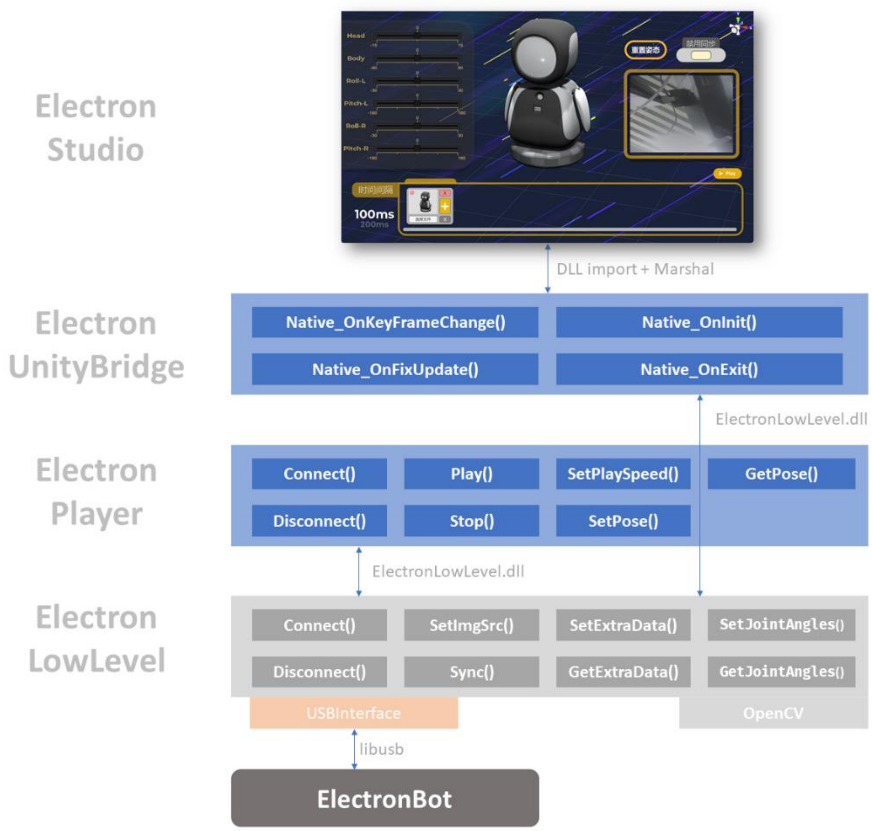


Figure 28 - SDK Design

Low-Level Electron (In Development): This component facilitates communication with the robot by offering functions to oversee connections, synchronize data, and share information, including images, additional data, and angles of the robotic arm.

Electron Player (Incomplete): This module is designed to develop algorithms that facilitate interaction between the user and the robot.

Electron UnityBridge (Incomplete): This component is responsible for establishing a connection between C++ code and the user interface created using Unity.

Electron Studio (UI Complete): This module represents the user interface application where operators interact with and control the robot.

Unity Engine UI

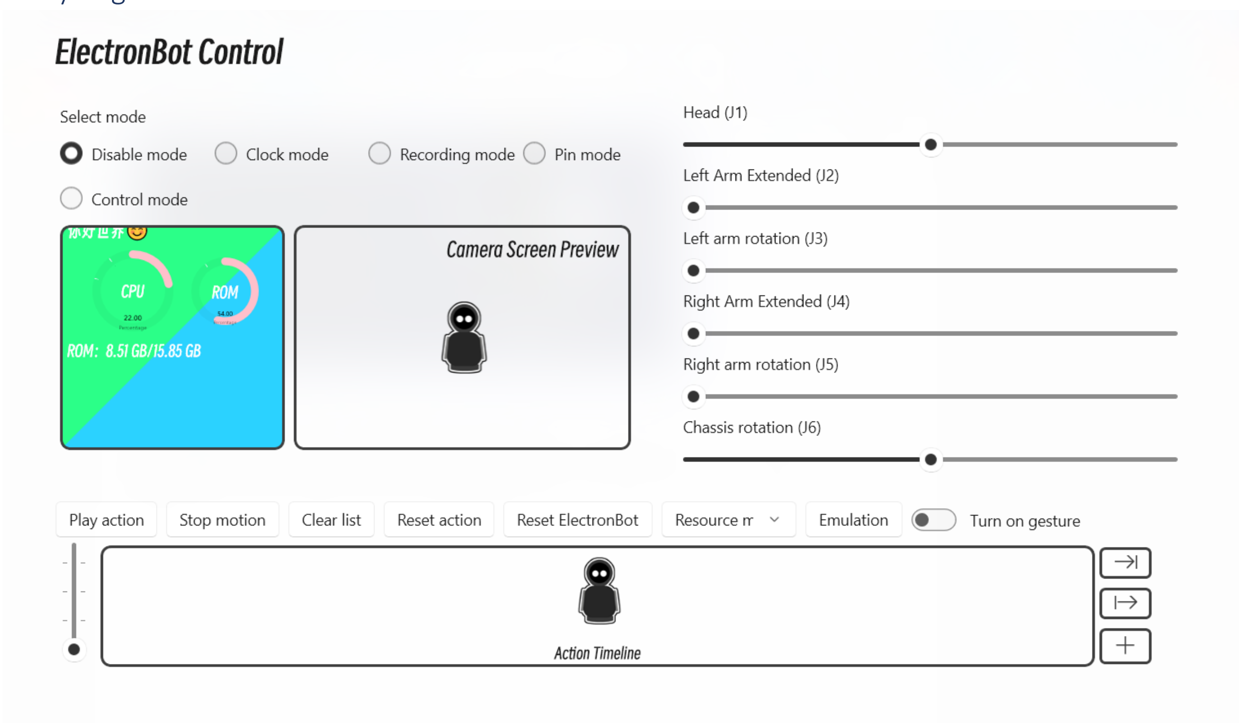


Figure 29 - Unity Engine User Interface

The 6 slide bars in **Figure 29 - Unity Engine User Interface** represent the 6 degrees of freedom of the robot.

The left panel is the preview of the screen display.

All recorded position are displayed in the action timeline.

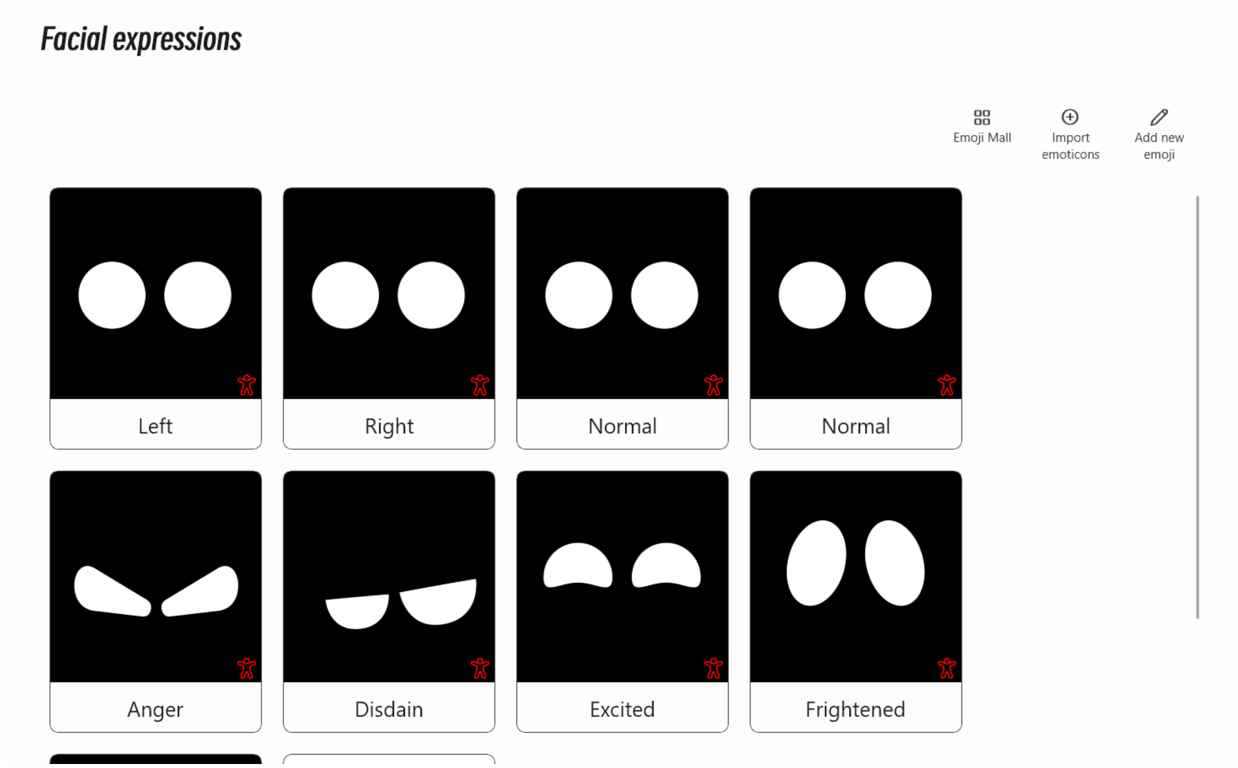


Figure 30 -Facial expressions

The system comes with some default emotional expressions.

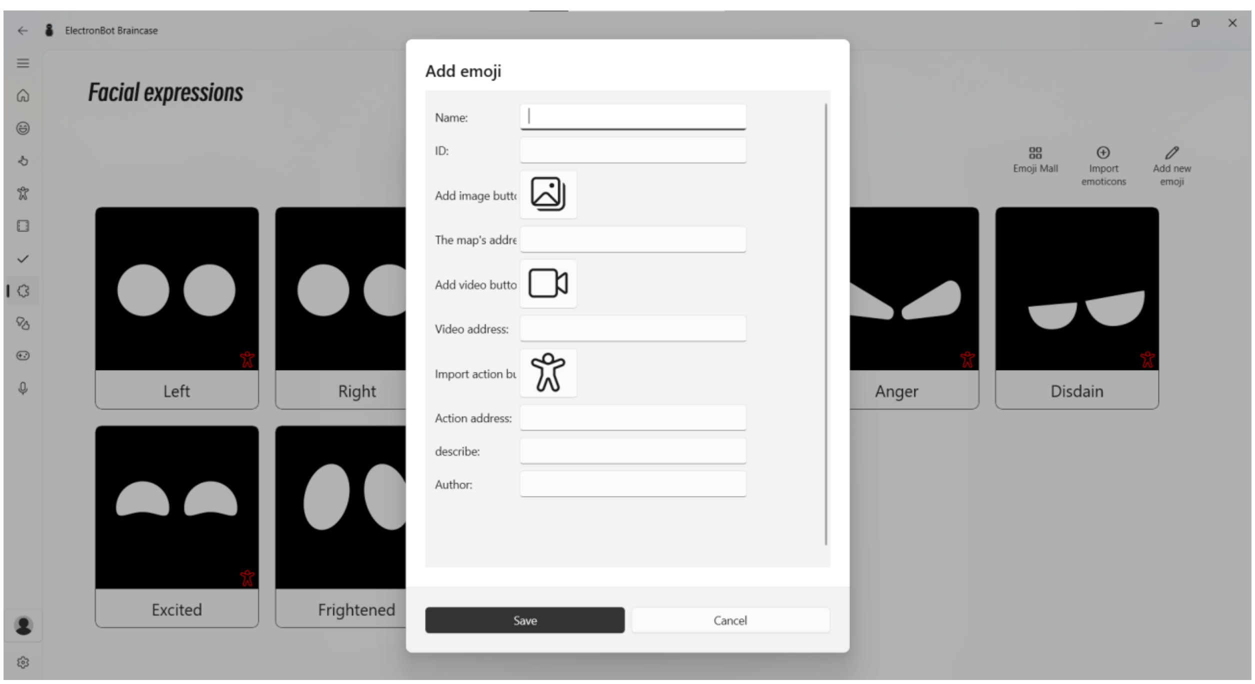


Figure 31 – Action Customization

Upload the video, image, and movements for action customization.

Movements are previously recorded in the control panel and can be exported and imported.

Figure 32 – Controller interface

Users can monitor the joint movement based on the angular feedback displayed in the controller panel.

Materials

Manufacturer Product Number	**Package**	**#of parts**	
CP2102 - Single Chip to UART Bridge	QFN-28	1	Chips
LP2992 3V3 - Voltage Regulator	SOT23-5	8	
STM32F405RGT6 - Microcontroller for the brain	LQFP-64_N	1	
STM32F030F6/STM32F042F6 - Microcontroller for the model	TSSOP-20	6	
USB3300-EZK-TR - USB Interface IC USB 2.0	QFN-32	1	
HS8836A - Simple and cheap 4-port USB2.0	SO-16N	1	
MPU6050 - 3 Axis Accelerometer Gyroscope	QFN-24	1	
PAJ7620U2 - Gesture Sensor	PAJ7620U2	1	
FM116B - DC Tail Motor/Steering Motor Driver	SOT23-6	6	
AO3400A - 30 V N-Channel MOSFET	SOT23-3	1	

TL431IDBZT - Voltage Detector	SOT23-3	6	
GMI4015P-2C-66db Electret Condenser Microphone	MIC-TH_BD4.0-P1.40-D0.4-L-FD		
PCM2912APJTR Audio CODEC with USB Interface	TQFP-32_L7.0-W7.0-P0.80-LS9.0-TL		
SC8002B 3W audio amplifier	SOP-8_L4.9-W3.9-P1.27-LS6.0-BL		
Type-C		1	Connectors
FFC_0.5_8	FFC_0.5_8	4	
SH1.0-3P-STAND - Electronic Wire Cable Connector	SH1.0-3P-STAND	1	
SH1.0-4P - Connectors	SH1.0-4P	6	
TF-Socket	TF-Card-Amphenol with TF	1	
LCD-240x240-ROUND	GC9A01	1	Miscellaneous
dial plate	31.5mm	1	
camera	USB camera	1	
FFC cable	5cm 0.5mm 8p	2	
3g steering gear		5	
9g steering gear		1	
SH1.0 3p wire		1	
SH1.0 4p wire		6	
bearing	shoulder 6x10x3mm	2	
bearing	waist 25x32x4mm	1	
putter	M2x25mm half tooth screw	2	
screw	M1x10mm, M2		
3D print shell			
SD card			

Table 2 - List of Materials

Discussion

Planned Practical Applications & Features

Practically, the intention is to allow the robot to have six degrees of freedom which could enable the robot to have a broader range of motion and versatility to aid in practical applications. Some practical applications for our mini desktop robot project include interacting with humans using voice control and a 3D Camera to do specific tasks, like dancing, showing emotes, and recognizing motions. Another plan that was initially in scope was to answer questions to operate using an AI algorithm like ChatGPT, that has been cut out due to time constraints. All in all, there is subject to change in terms of features that may come as a part of the further development of the project reaches its baseline for the initial phase and the existing applications/features listed above will be subject to a scale down terms of complexity and data analysis as discussed in **Future Scope for Further Development**.

Future Scope for Further Development

Due to the two original group members from the first part of the Capstone project being unable to work on the project this term, a decision was made to reduce the scope of the overall project from features and software capabilities to leave further development of the project from more of a software-based perspective to the two group members upon their return to school next fall.

Future Tasks may include:

**Note that these tasks are subject to change and may increase/decrease in scope depending on what the two group members want to do*

- Initialize web-accessing capabilities
- Build a voice command database
- Improve speech recognition accuracy
- Further SDK designs
- 3D Camera improvements/optimization
- Further development to expand capabilities for potential educational use
- Development of a user-based guide for the robot
- Build a voice recognition model
  - Data Collection
  - Model Selection
- Integrate the model with the application
  - API
  - Modal Quantization
- Initiate Wireless Connection
  - Protocol Selection
  - Broker Selection
- Improve Compatibility
  - Firmware
  - Development Environment

Potential Risks and Solutions

Risks are inherent in every project, whether it's the initial planning, implementation, or final evaluation phase. They are nearly unavoidable. However, with precautions, the project's overall progress can remain largely unaffected, adhering closely to the original timeline and plan even in the event of risks occurring. Below, we have identified risks that may arise during the implementation or testing phase, along with corresponding potential solutions, strategies for prevention, and contingency plans.

Risk	Explanation	Strategies to minimize potential risk	Solution
Material Damage/Loss Likelihood: Low Impact: Low	• Improper handling of parts/components can lead to hardware damage or loss.	• Handle the parts/components with care. Avoid mistakes when working with delicate parts. • Keep parts/components in a secure location to avoid losing them.	• Buy additional parts to replace or supplement existing ones.
Scope Creep Likelihood: Low Impact: High	• Changes in the Project that might disturb the timeline unexpectedly.	• Team members should prepare schedules. • Create a straightforward	• Record the progressing using project management software.

		documentation plan for each phase and possible changes that might affect the project scope. • Have weekly meetings with team members regarding the scope and progress.	
Hardware Failures Likelihood: Low-medium Impact: High	• The robot relies on mechanical and electrical elements like sensors, motors, and power supply. In the event of malfunctions, this could result in losing functionality, or even complete breakdown. Additionally, there's a potential risk of overheating, which could pose electrical hazards.	• Verify if the components are properly soldered and wired. • Consistently verify the circuit's wiring and test in simulated conditions • Limit the testing so minimum to increase the lifespan of components. • Avoid overloading the components.	• Buy additional parts to replace or supplement existing ones in case of hardware failure.
Physical Stability and Balance Likelihood: Low-medium Impact: Low	• The model design or construction can be unstable or prone to tipping over due to weight.	• Ensure the weight is evenly distributed. • Sturdy construction. • Testing the model	• If this problem does occur do the necessary changes to the model design and reprint it.
Software Failures Likelihood: Low-medium Impact: Low	• Poorly written code and unorganized documentation will cause problems when multiple people are working on it.	• Set up an SDK with clear and comprehensive documentation. • Handle errors beforehand. • Do code reviews and test in	• Optimize the code and SDK for performance. • Document the code to help with understanding and

	• Performance issues due to the code.	realistic conditions.	troubleshooting it.
Firmware Failures Likelihood: Low-medium Impact: Medium-high	• Incorrect or careless programming may result in program crashes or the robot not performing as intended, potentially resulting in component failure.	• Verify the code and run simulations. • Go through the guide for reference.	• Thoroughly review the code and run it to address any errors, as a precaution, without connecting it to the hardware.
Compatibility Likelihood: Low-medium Impact: Medium	• Software/firmware might not be compatible with the robot's hardware, leading to issues and delays	• Ensure that the software and drivers designed for your robot's specific hardware and firmware configuration are thoroughly checked for compatibility. Additionally, check compatibility with popular development environments	• If the hardware remains functional despite firmware errors, re-examine, and configure it.
Environmental Factors Likelihood: Medium Impact: Medium-high	• Dust accumulation can hinder the performance or damage the robot's components. • Extreme temperatures and high humidity can shorten components' lifespan and hinder its performance. • EMI, where electrical signals	• use dust-resistant materials and regularly clean the robot's components. • Testing in a temperature, humidity-controlled place. • Utilize shielded cables to protect the robot from electromagnetic interference, employ noise-cancelling	• If the component is still operational, work on the rest of the robot in an environment-controlled place with no EMI or noise. • use filters in the circuits and perform regular dust cleaning." • If the component is

	from nearby sources can disrupt the robot's sensors or communication capabilities. • Loud noises can interfere with robots' ability as it can affect its sensors and communication system. • Potential risks of malware, DoS attacks, and concerns regarding data privacy and security, all of which have the potential to disrupt the robot's functionality.	equipment to isolate it from noisy surroundings, and integrate low-pass filters in the circuits to minimize noise and EMI effects. • Set up strong security protocols, encompassing firewalls, antivirus software, encryption, and access controls.	not operational, use spare parts. • If any components have already sustained damage, replace the parts.
User Misuse Likelihood: low Impact: Medium	• Members of the group might not recognize the limitation of the components or robots.	• Read the datasheet and manual of the components to understand the limitations of the hardware.	• Be careful while working with the robot.

Table 3 - Potential Risks and Solutions

Challenges

Below are some of the challenges are team faced from project initialization in January 2023 to project finalization at the end of November 2023:

1) Lost two members from 4TR1 due to various reasons; resulted in heavier workload and restructuring
2) Slow hardware integration process
3) Firmware/Software issues pertaining to files
4) Windows 11 is unable to integrate with the firmware in use for the robot; Windows 10 is compatible with the robot only currently
5) Gesture and voice recognition modules were unable to be completed due to time constraints and knowledge gap
6) Camera issues pertaining to connectivity and reliability
7) Faulty USB connection that can be laggy/unreliable

Timeline Continued

Milestone	Anticipated Time	Actual Time
Final decision for ideas	January 17, 2023	January 30, 2023
Buying hardware	January 30, 2023	January 30, 2023
Project Proposal	February 6, 2023	February 6, 2023
Presentation of Project Plan	February 12, 2023	February 13, 2023
Hardware arrival	Late February 2023	Early March 2023
Printing prototypes	Late February 2023	Late February 2023
Midterm report	Late February 2023	March 13, 2023
Final presentation and report	Early April 2023	April 10, 2023 (Presentation) April 13, 2023 (Report)
Starting to assemble parts	Late May 2023	September 2023
Software Setup	Late May 2023	October 2023
Hardware Integration	Late October 2023	Reading Week, October
Midterm Report	October 19, 2023	October 19 2023
Midterm Presentation	October 26, 2023	October 26, 2023
Firmware Testing and Implementation	October - November 2023	Mid-November
Software Testing and Implementation	October - November 2023	Mid-November
Final Report	Late November 2023	December 7, 2023
Final Presentation and Demo	Late November 2023	November 30, 2023
Final Project Completion	Late November 2023	November 30, 2023

Table 4 - Project Timeline

Conclusion

In conclusion, our final capstone report underscores the substantial progress achieved in the development of the Mini-Desktop robot, aligning with the adjusted scope established at the semester's outset, as described in the **Objectives** section. Notably, while significant strides were made in various aspects, our gesture and voice recognition modules encountered challenges that hindered their completion within the designated timeline. These challenges encompassed issues related to software knowledge and compatibility with the required C# language as well as overall timeline constraints. Consequently, the integration of these modules has been designated to **Future Scope for Further Development,** forming a pivotal aspect of our task handoff to the succeeding team members, comprised of our two predecessors from 4TR1 and potential future contributors.

Throughout the project, there was a concerted effort to adhere to timelines and maintain a balance between speed and steadiness in progress. As emphasized in this report, the forthcoming 4TR3 capstone section's team will inherit the project, which ultimately prompted a reduction in the overall scope of features and software capabilities, as detailed in the **Objectives** section. This conclusive report encapsulates our accomplished body of work, incorporating supplementary information from previous stages, and outlines the trajectory for the continued development and enhancement of the mini-desktop robot.

References

- Peng, Z. (2020, January 12). ElectronBot: A DIY robot which can automatically follow a line or run freely in an unknown environment. GitHub. https://github.com/peng-zhihui/ElectronBot/blob/main/enREADME.md
- GreenShadeZhang. (2021). ElectronBot.DotNet. GitHub. Retrieved from https://github.com/maker-community/ElectronBot.DotNet
- Kinivi. (2020). Tello-gesture-control. GitHub. Retrieved from https://github.com/kinivi/tello-gesture-control
- McMaster University Library. (n.d.). 3D printer safety information. https://library.mcmaster.ca/sites/default/files/uploaded_files/3d_printer_safety_info.pd
- STMicroelectronics. (n.d.). STM32F042F6P6. Mouser Electronics. https://www.mouser.ca/ProductDetail/STMicroelectronics/STM32F042F6P6?qs=4b8myOmUP%252BsTtpJfUMo83g%3D%3D
- Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., & Sheikh, Y. (2019). OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 172-186. https://github.com/CMU-Perceptual-Computing-Lab/openpose
- AutoHotkey. (n.d.). GitHub. https://github.com/AutoHotkey/AutoHotkey
- Texas Instruments. (n.d.). TL431 Precision adjustable shunt regulator. TI.com. https://www.ti.com/product/TL431/part-details/TL431IDBZT
- RushPCB Inc. (n.d.). Detection and Prevention of Solder Joint Failure. RushPCB.com. https://rushpcb.com/detection-prevention-of-solder-joint-failure/#:~:text=According%20to%20Rush%20PCB%2C%20there,Fatigue%E2%80%94from%20cyclic%20loading.
- "PAJ7620U2 gesture sensor," *PAJ7620U2 Gesture Sensor - Waveshare Wiki*. [Online]. Available: https://www.waveshare.com/wiki/PAJ7620U2_Gesture_Sensor. [Accessed: 12-Apr-2023].

- *St-link/V2*. STMicroelectronics. (n.d.). https://www.st.com/en/development-tools/st-link-v2.html
- Teel, J. (2023a, September 20). *Introduction to programming STM32 ARM cortex-M 32-bit microcontrollers (updated 2022)*. PREDICTABLE DESIGNS. https://predictabledesigns.com/introduction-to-programming-stm32-arm-cortex-m-32-bit-microcontrollers/#:~:text=STM32%20microcontrollers%20offer%20a%20large,%2C%20cameras%2C%20motors%2C%20etc.
- Teel, J. (2023b, September 23). *Firmware programming – how to Format Serial Communication Data*. PREDICTABLE DESIGNS. https://predictabledesigns.com/firmware-programming-how-to-format-serial-communication-data/
- *Audio Products/Micromotors/Microphones: Gmi4015p-2c-66db plugin,D=4mm microphones RoHS*. Audio Products/micromotors/microphones: Gmi4015p-2c-66db Plugin,D=4mm Microphones Rohs - Buy Gmi4015p-2c-66db Audio Products/micromotors/microphones,Gmi4015p-2c-66db Component,Gmi4015p-2c-66db Product on Alibaba.com. (n.d.). https://www.alibaba.com/product-detail/Audio-Products-Micromotors-Microphttps://www.alibaba.com/product-detail/Audio-Products-Micromotors-Microphones-GMI4015P-2C_1600829741306.htmlhones-GMI4015P-2C_1600829741306.html
- *Electronics components distributor to the world.* X. (n.d.). https://www.xonelec.com/mpn/inghai/gmi4015p2c66db
- *LP2992*. LP2992 data sheet, product information and support | TI.com. (n.d.-a). https://www.ti.com/product/LP2992?utm_source=google&utm_medium=cpc&utm_campaign=app-lp-null-44700045336317407_prodfolderdynamic-cpc-pf-google-wwe_int&utm_content=prodfolddynamic&ds_k=DYNAMIC%2BSEARCH%2BADS&DCM=yes&gclid=CjwKCAjwyNSoBhA9EiwA5aYlb03eKzzckiKFFWt4phZaGkZpW0yFN_1Ldgz1LNz5xEB4nH7M2YklFBoCsRYQAvD_BwE&gclsrc=aw.ds
- *LP2992*. LP2992 data sheet, product information and support | TI.com. (n.d.-b). https://www.ti.com/product/LP2992?utm_source=google&utm_medium=cpc&utm_campaign=app-lp-null-44700045336317407_prodfolderdynamic-cpc-pf-google-wwe_int&utm_content=prodfolddynamic&ds_k=DYNAMIC%2BSEARCH%2BADS&DCM=yes&gclid=CjwKCAjwyNSoBhA9EiwA5aYlb03eKzzckiKFFWt4phZaGkZpW0yFN_1Ldgz1LNz5xEB4nH7M2YklFBoCsRYQAvD_BwE&gclsrc=aw.ds
- LP2992IM5-3.3/NOPB texas instruments | integrated circuits (ICS) | digikey. (n.d.-a). https://www.digikey.com/en/products/detail/texas-instruments/LP2992IM5-3-3-NOPB/483035
- PCM2912APJTR Texas Instruments | Integrated Circuits (ICS) | digikey. (n.d.-b). https://www.digikey.com/en/products/detail/texas-instruments/PCM2912APJTR/2047509
- *Shenzhen Fuman Elec SC8002B*. www.lcsc.com. (n.d.). https://www.lcsc.com/product-detail/Audio-Power-OpAmps_Shenzhen-Fuman-Elec-SC8002B_C82124.html
- *St-link/V2*. STMicroelectronics. (n.d.). https://www.st.com/en/development-tools/st-link-v2.html
- *STM32 32-bit ARM® cortex®-M mcus*. Mouser. (2008, February 21). https://www.mouser.ca/new/stmicroelectronics/stm32/
- What is a voltage regulator - digi-key electronics. (n.d.-c). https://www.digikey.ca/en/maker/blogs/2020/what-is-a-voltage-regulator