Robo Arm Assembly and Motor Setup Guide

Initial Setup

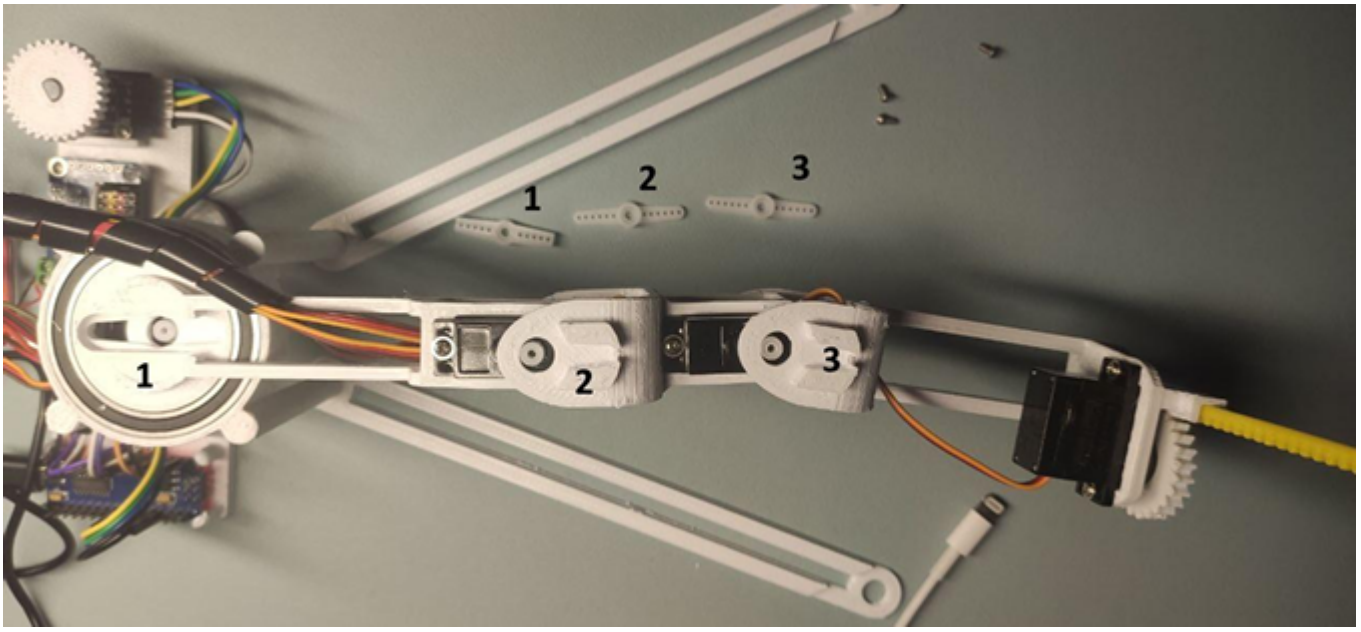
This guide walks you through the process of setting up your Robo Arm, aligning its motors, and preparing it for operation using the BotEditorPy application. Follow these steps carefully to ensure proper assembly and motor calibration before attaching the motor horns.

Prerequisites

- **Robo Arm Hardware**: Ensure your Robo Arm is assembled

[How To Assemble RoBoArm]

, but **do not attach the horns** for the FI, SE, and TH motors (motors 1, 2, and 3), nor the gear and rack system for the 4th motor, at this stage.



- **Power Supply**: Connect the Robo Arm to a 5V power adapter and ensure it is powered on.
- **Arduino and PCA9685 Board**: Verify that the PCA9685 board's power cable is connected to the Robo Arm.
- **Computer**: A system running Windows, Linux, or macOS with Python 3.8 or higher installed.
- **BotEditorPy Application**: Obtain the BotEditorPy project files, including main.py, requirements.txt, install_and_run_windows.bat (Windows), install_and_run_linux.sh (Linux), and install_and_run_macos.sh (macOS).

Step 1: Set Up the BotEditorPy Application

1. **Prepare the Project Files**:
    - Copy the BotEditorPy project files to a directory on your computer (e.g., [PathTo]\BotEditorPy on Windows, /path/to/BotEditorPy on Linux/macOS).
2. **Run the Setup Script**:
    - **Windows**:

Open a Command Prompt, navigate to the project directory, and run the setup script:
cd C:\reps\BotEditorPy

        - install_and_run_windows.bat
        - This script installs dependencies, sets up a virtual environment (myenv), and launches the application.
    - **Linux**:

Open a terminal, navigate to the project directory, and run:
cd /path/to/BotEditorPy
chmod +x install_and_run_linux.sh

        - ./install_and_run_linux.sh
    - **macOS**:

Open a terminal, navigate to the project directory, and run:
cd /path/to/BotEditorPy
chmod +x install_and_run_macos.sh

        - ./install_and_run_macos.sh

Step 2: Configure the Arduino Connection

1. **Launch the Application**:

If the setup script didn't automatically launch the application, manually start it after setup:
cd /path/to/BotEditorPy
source myenv/bin/activate # On Windows: myenv\Scripts\activate

    - python main.py
    - The BotEditorPy GUI will open, displaying the "Keys", "Scripts", and "Listen Mic" tabs.
2. **Select the COM Port**:
    - Navigate to the "Keys" tab in the BotEditorPy application.
    - Locate the COM port dropdown or field for your Arduino (e.g., COM3 on Windows, /dev/ttyUSB0 on Linux/macOS).
    - If the application auto-detects the Arduino, verify the selected port is correct. Otherwise, manually select the appropriate port.
    - Click the "Save" button to store your selection.

Step 3: Align the Motors

1. **Set Motors to Neutral Position**:
    - In the "Keys" tab, click the "Align all motors" button.
    - This sets all motors (FI, SE, TH, and the 4th motor) to their neutral (middle) position, ensuring they are properly calibrated before assembly.

Step 4: Attach the Motor Horns and Gear System

1. **Attach Horns to FI, SE, and TH Motors**:
    - With the motors aligned to their neutral positions, attach the horns to the FI, SE, and TH motors (motors 1, 2, and 3).
    - Ensure the horns are securely fastened and aligned according to the Robo Arm's design specifications.
2. **Configure the Gear and Rack System for the 4th Motor**:

    - Assemble the gear and rack system for the 4th motor, ensuring it is properly aligned with the neutral position set by the application.

Step 5: Verify Motor Alignment (Optional)

- **Retrieve Current Angles**:
    - In the "Keys" tab, click the "Get Actual Data" button to retrieve the current angles from the Arduino.
    - Verify that the GUI sliders for FI, SE, and TH reflect the neutral positions set earlier.
- **Fine-Tune with Rotary Encoder (Optional)**:
    - If you have a rotary encoder connected to your Arduino:
        - Click the encoder's button to cycle between the FI, SE, and TH motors.
        - Rotate the encoder to fine-tune the angle of the selected motor, ensuring precise alignment.
        - Changes are sent to the Arduino in real-time, and the GUI updates accordingly.

Next Steps

- **Test Functionality**:
    - Use the "Keys" tab to manually adjust motor angles and confirm they move as expected.
    - Test voice commands in the "Listen Mic" tab (e.g., say "Hello World" to execute a script).
    - Create and run scripts in the "Scripts" tab to automate sequences.
- **Refer to User Guide**:
    - See USER_GUIDE.md for detailed instructions on using all features of BotEditorPy.

Troubleshooting

- **Arduino Not Detected**:
    - Ensure the USB cable is connected and the Arduino is powered.
    - Verify the COM port in the "Keys" tab matches your Arduino's port.
- **Motors Not Moving**:
    - Confirm the PCA9685 board is powered and connected.
    - Check that the "Align all motors" step was completed before attaching the horns.
- **Application Fails to Launch**:
    - Ensure all dependencies were installed by the setup script.
    - Check the terminal output for error messages and resolve any missing dependencies