Braccio Arm Calibration Guide:

Objectives:

- Find the servo command position that results in a straight joint
- Check for and prevent any collision between different joints due to over rotation.
- Check for and remedy any wires which are preventing full range of motion or getting pinched by the arm.
- Determine the calibration values necessary for use in future programs.
- At the end of this calibration, the arm should stand straight up during the initial power on and, during normal operation, should not hit any physical end stops on any joints.

Important terms/things to know:

Servo command position – The value in software that is sent to the servo to tell it what position to go to, ranging from 0 to 180. The nominal unit is degrees but it may not line up completely, as we will see in the calibration.

Servo range – Each servo has a command range of 0 to 180, whole numbers only, which in an ideal situation will map to 180 degrees of rotation.

Joints – There are six joints on the Braccio arm. From the bottom, these are the names which they will be referred to by and in parentheses the name of the joint in software: base rotation (BASE_ROT), shoulder (SHOULDER), elbow (ELBOW), wrist (WRIST), wrist rotation (WRIST_ROT), and gripper (GRIPPER).

Physical end stops – A place where the arm cannot continue to move due to the physical parts of the arm contacting each other, see Figure 1 below.

Braccio shield – The printed circuit board that interfaces the Arduino microcontroller and the Braccio arm.

"send a __ character" – This shorthand implies the following set of steps:

- Ensure the Serial Monitor entry box, see Figure 2, item 2, is empty.
- Enter the character, without any quotes, lowercase, and if followed by a number, without any spaces.
- Press the send button, see Figure 2, item 3, or press the enter key to send the command.

Notes:

The wooden base has markings in degrees, but the position of the base will not necessarily correspond to servo command positions.

Calibration Instructions:

- Place the arm in a space where nothing will constrain movement in any dimension, ideally with 2 feet of open space in any direction.
- Open the Arduino IDE.
- Open the calibration example by navigating, in the top menu, to File → Examples → BraccioV2 → Braccio_Calibration.
- Plug in the power supply for the Braccio arm and connect the barrel plug to the jack on the Braccio shield.
- Connect the Arduino board to the computer with a USB cable.
- Upload the program to the Arduino, you may need to select the correct port for the board in the top menu, via Tools → Port.
- Open the serial monitor either through the top menu, Tools → Serial Monitor, or through the Ctrl+Shift+M key combination. Set the baud rate to 9600 via the drop down menu and set the line ending to “No line ending” See Figure 2, item 1.
- We will calibrate the arm joint by joint, starting at the bottom.
- Base rotation joint:
    - See if the base is aligned, meaning that the front of the base is lined up with one of the 90° positions, labeled 0°, 90°, -180°, or 180°. If it is, calibration is complete, and send the 's' command.
    - If the base is not aligned, try sending different values to get the base aligned straight. The serial monitor readout will show the current position, try setting values within +/- 10 of the current value. To set a new value, send a command of 'c' plus the desired value, i.e. 'c78' to change the joint to 78.
    - Once the base is aligned, send the 's' command to save the center position.
    - The base will now rotate to one extremity. Ensure that the arm is not hitting any physical end stops, if it is, adjust as necessary following the protocol of the previous step, otherwise send the 's' command to save the position. Note: the extremities of the joint will not necessarily line up with a straight 90° angle on the base due to minor tolerances in the servo/arm.
    - Repeat the previous step for the other extremity.
- Ensure that the Serial monitor shows that the current joint is SHOULDER. There should be a line saying “Current joint: SHOULDER” If that line displays BASE_ROT, go back and complete the calibration for the base rotation joint. If any other joint name is displayed, reset the Arduino and begin the calibration again.
- Shoulder joint:
    - See if the shoulder joint is facing straight up, meaning that the arm, at the shoulder joint, is not tilting more one way or the other. If the joint is already exactly straight up, send the 's' command to save that position.
    - If the shoulder is not straight up, try sending different values to get the shoulder aligned straight. The serial monitor readout will show the current position, try setting values within +/- 10 of the current value. To set a new value, send a command of 'c' plus the desired value, in the manner like the previous joint's calibration.
    - Once the shoulder is pointing straight up, send the 's' command to save the position. Note: Before doing this, hold down the wooden base as the arm will next move to the extremities of the shoulder joint and may cause the arm to tip if not supported. Also ensure that the arm will not hit any items in the way.

    - Check that the arm is not hitting any physical end stops around the shoulder joint. This joint is particularly susceptible to hitting the base at the far extremes of movement. If the arm is hitting the base, send values that are closer towards center, values closer to 90, using the 'c' command, until the arm is not contacting anything. Then send the 's' command to save the position.
    - Repeat the previous step for the other extremity.
- Ensure that the Serial monitor shows that the current joint is ELBOW. There should be a line saying “Current joint: ELBOW” If that line displays SHOULDER, go back and complete the calibration for the shoulder joint. If any other joint name is displayed, reset the Arduino and begin the calibration again.
- Elbow joint:
    - See if the elbow joint is facing straight up, meaning that the arm, at the elbow joint, is not tilting more one way or the other. If the joint is already exactly straight up, send the 's' command to save that position.
    - If the elbow is not straight up, try sending different values to get the elbow aligned straight. The serial monitor readout will show the current position, try setting values within +/- 10 of the current value. To set a new value, send a command of 'c' plus the desired value, in the manner as previously described.
    - Once the elbow is pointing straight up, send the 's' command to save the position. Note: The elbow joint will move abruptly to its extremities and may cause the arm to tip if not supported.
    - Check that the arm is not hitting any physical end stops around the elbow joint. If the arm is hitting another part of the arm, send values that are closer towards center, values closer to 90, using the 'c' command, until the arm is not contacting anything. Then send the 's' command to save the position.
    - Repeat the previous step for the other extremity.
- Ensure that the Serial monitor shows that the current joint is WRIST. There should be a line saying “Current joint: WRIST” If that line displays ELBOW, go back and complete the calibration for the elbow joint. If any other joint name is displayed, reset the Arduino and begin the calibration again.
- Wrist joint:
    - See if the wrist joint is facing straight up, meaning that the arm, at the wrist joint, is not tilting more one way or the other. If the joint is already exactly straight up, send the 's' command to save that position.
    - If the wrist is not straight up, try sending different values to get the wrist aligned straight. The serial monitor readout will show the current position, try setting values within +/- 10 of the current value. To set a new value, send a command of 'c' plus the desired value, in the manner as previously described.
    - Once the wrist is pointing straight up, send the 's' command to save the position.
    - Check that the arm is not hitting any physical end stops around the wrist joint. If the arm is hitting another part of the arm, send values that are closer towards center, values closer to 90, using the 'c' command, until the arm is not contacting anything. Then send the 's' command to save the position.
    - Repeat the previous step for the other extremity.
- Ensure that the Serial monitor shows that the current joint is WRIST_ROT. There should be a line saying “Current joint: WRIST_ROT” If that line displays WRIST, go back and complete the calibration for the wrist joint. If any other joint name is displayed, reset the Arduino and

begin the calibration again.

- Wrist rotation joint:
    - See if the wrist rotation joint is positioned center with regards to the rest of the arm. If the joint is already exactly centered, send the 's' command to save that position.
    - If the wrist rotation is not properly aligned, try sending different values to get the joint aligned. The serial monitor readout will show the current position, try setting values within +/- 10 of the current value. To set a new value, send a command of 'c' plus the desired value, in the manner as previously described.
    - Once the joint is centered, send the 's' command to save the position.
    - Check that the arm is not hitting any physical end stops around the elbow joint. Make sure to check for any wires that are strained by the rotational movement. If the arm is hitting another part of the arm or pinching a wire, send values that are closer towards center, values closer to 90, using the 'c' command, until the arm is not contacting anything. Then send the 's' command to save the position.
    - Repeat the previous step for the other extremity.
- Ensure that the Serial monitor shows that the current joint is GRIPPER. There should be a line saying “Current joint: GRIPPER” If that line displays WRIST_ROT, go back and complete the calibration for the wrist rotation joint. If any other joint name is displayed, reset the Arduino and begin the calibration again.
- Gripper joint:
    - The gripper is special in that it has a reduced range of servo command values. Testing with some platforms yielded a rough average range of 10 to 90. This calibration will help determine the values that make the gripper open or close without straining the servo.
    - This first calibration value establishes a neutral position for the gripper. This position is not of great importance, if you would like to change it, use the 'c' command plus a value to change it, otherwise save the default position with the 's' command.
    - The extremities are of greater importance. As the servo moves to each extremity, ensure the servo is not buzzing, meaning that it can't reach its command position, and if it is, adjust as necessary. Also ensure that, when closed, the gripper has enough force to allow it to grip small objects without the “jaws” being too far apart and losing their grip. Send the 'c' command with an updated value to calibrate the positions for opened and closed and send the 's' command to save them.
- Now you should see a message in the Serial monitor along the lines of 'Calibration done, copy values into your program to use.' If you don't see this message, make sure you have completed the calibration for all the joints.
- To extract the calibration code, scroll through the serial output and copy any lines between the blocks “/***** Begin JOINT Configuration *****/” and “/***** End JOINT Configuration *****/” Paste these values into the setup function of your code or into any of the examples to use the calibration with the library. Make sure to save those lines for use in future programs as this calibration will apply in any program using the arm.

Figure 1, physical end stop picture

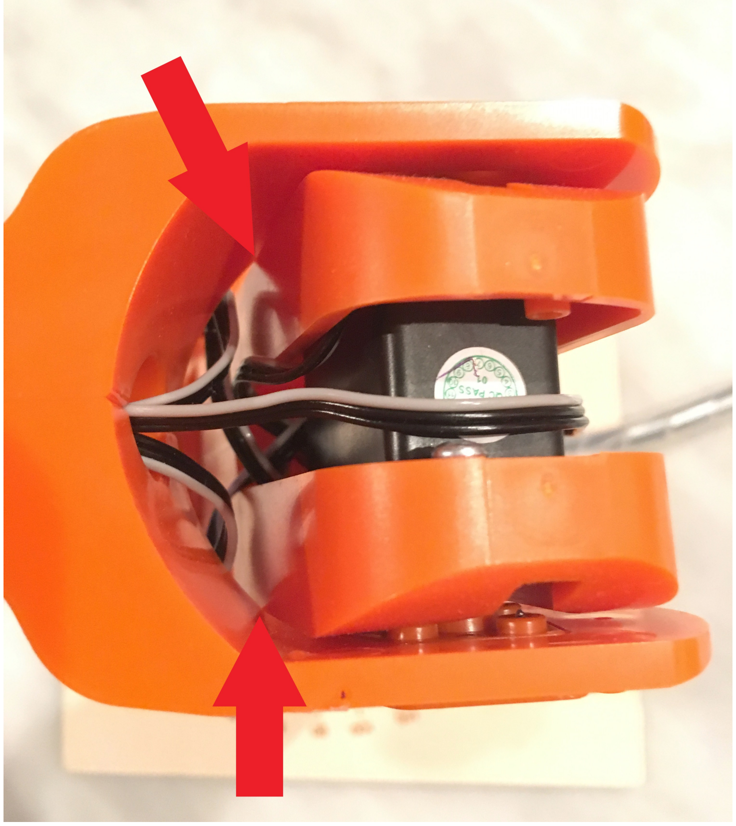
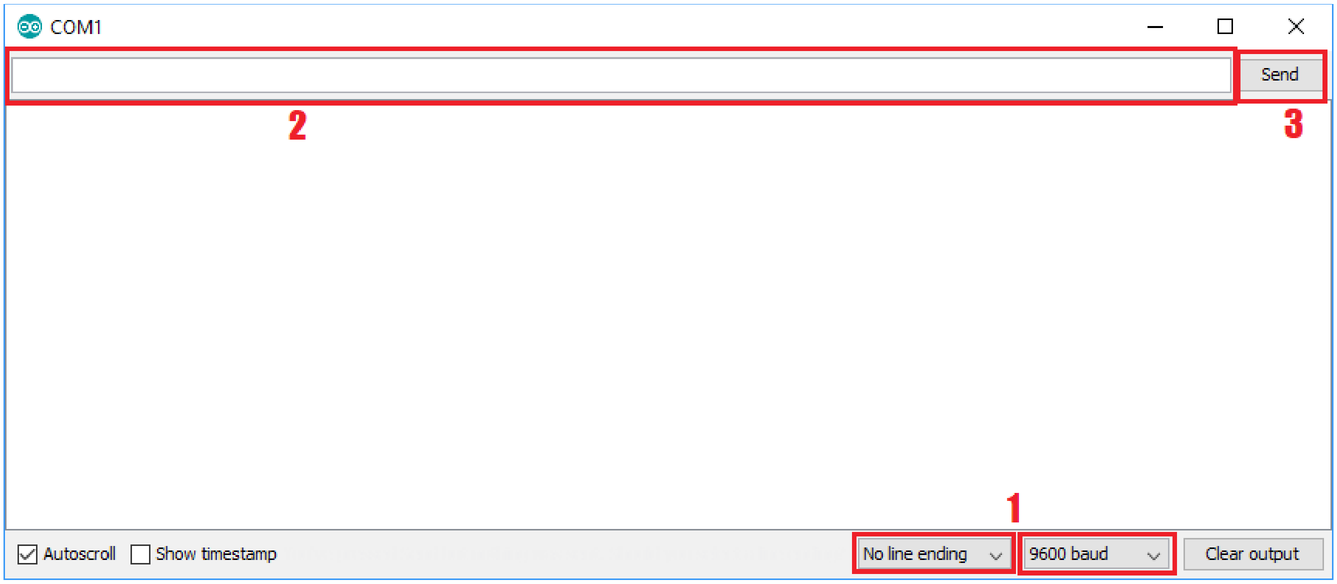


Figure 2, Serial monitor configuration



☑ Autoscroll □ Show timestamp

No line ending

9600 baud