

# Open Source Rover: Calibration

Authors: Michael Cox, Eric Junkins, Olivia Lofaro



**Jet Propulsion Laboratory**  
California Institute of Technology

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. ©2018 California Institute of Technology. Government sponsorship acknowledged.

## Contents

<b>1</b>	<b>Calibration</b>	<b>2</b>
1.1	RoboClaw Setup . . . . .	2
1.2	Drive Motor Calibration . . . . .	5
1.3	Corner Motor Calibration . . . . .	9

## 1 Calibration

In order for the motors to all work properly and drive the rover correctly, we must calibrate the motors and encoders. We will use a program called Ion Studio to assist with this. Ion Studio can be downloaded at

- <http://www.ionmc.com/downloads>.

We will now go through and check the encoders and motors one at a time and perform the calibration process on each of them. First, make sure that either the robot is fully hanging such that every drive wheel is not in contact with the ground, or that you elevate each drive wheel as you attempt to spin that wheel.

### 1.1 RoboClaw Setup

1. Connect your computer (with Ion Studio installed) to the RoboClaw via USB-micro USB cable
2. Open Ion Studio
3. If prompted, update the RoboClaw Firmware

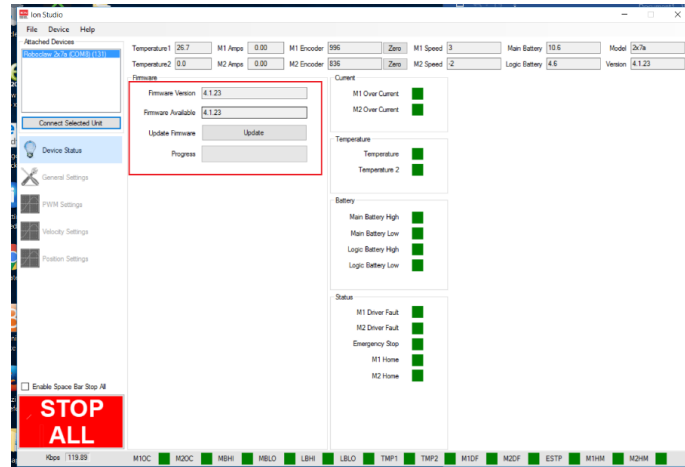


Figure 1: Ion Studio Firmware Update

4. Click the 'Connect Selected Unit' box in the upper left
5. Verify that all the motor control parameters are roughly nominal:

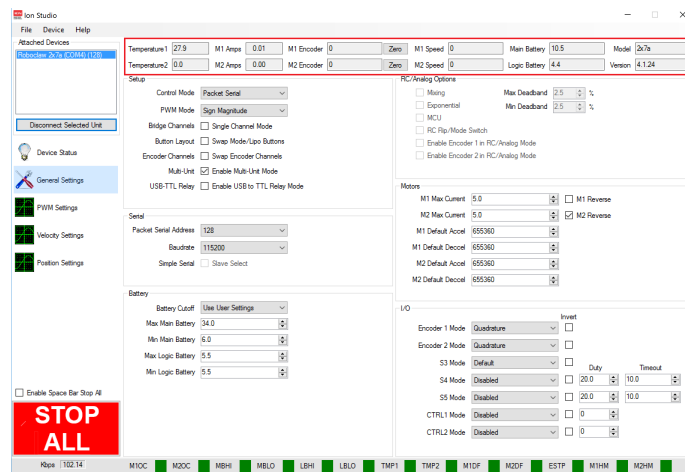


Figure 2

- **Temperature1:** 30
- **M1/M2 Amps:** 0.01
- **M1/M2 Encoder:** 0
- **M1/M2 Speed:** 0
- **Main Battery:** Between 11.5-16.7V

- **Logic Battery:** 5V
- **Model:** 2x7a

As you perform the calibration and testing outlined in this document, make sure to monitor the values above and make sure they are changing in the way that you expect they should. For instance, the temperature going much higher than 35 would indicate that something is going wrong, or if you are pulling a large amount of current you might have a short in the wiring.

6. Under the General Settings Tab, you'll need to change the following settings for each RoboClaw:

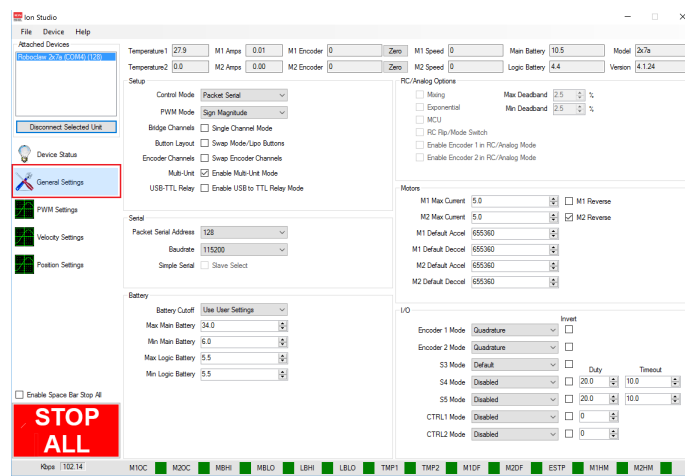


Figure 3

(a) **Setup:**

- Control Mode: Packet Serial
- Multi-Unit: Check Enable Multi-Unit Mode

(b) **Serial:**

- Set the address to 128,129,130,131,132 for each respective RoboClaw (RoboClaw 1 is address 128, 2 is address 129, etc)
- Baudrate: 115200

(c) **Battery:**

- i. Max Main Battery: 17.5V
- ii. Min Main Battery: 11.5V
- iii. Max Logic Battery: 5.5V
- iv. Min Logic Battery: 4.5V

(d) **Motors:**

- i. M1 Max Current: 3.0A
- ii. M2 Max Current: 3.0 A

(e) **I/O:**

- i. Encoder 1 Mode: Quadrature for addresses 128,129,130; Absolute for addresses 131 and 132
- ii. Encoder 2 Mode: Quadrature for addresses 128,129,130; Absolute for addresses 131 and 132

**After changing these settings, make sure to save them! (*Device* – > *Save Settings*)**

## 1.2 Drive Motor Calibration

7. Perform the following steps for the **RoboClaw** addresses 128,129, and 130:

- (a) Go to the PWM settings tab

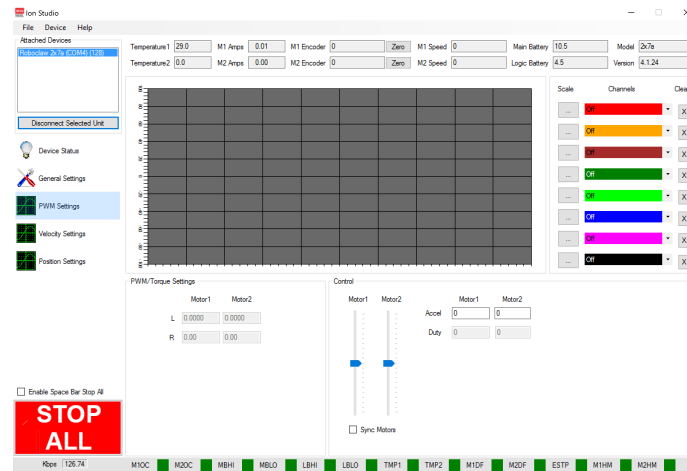


Figure 4

(b) Under the control pane, slowly move the slider bar up for Motor1.

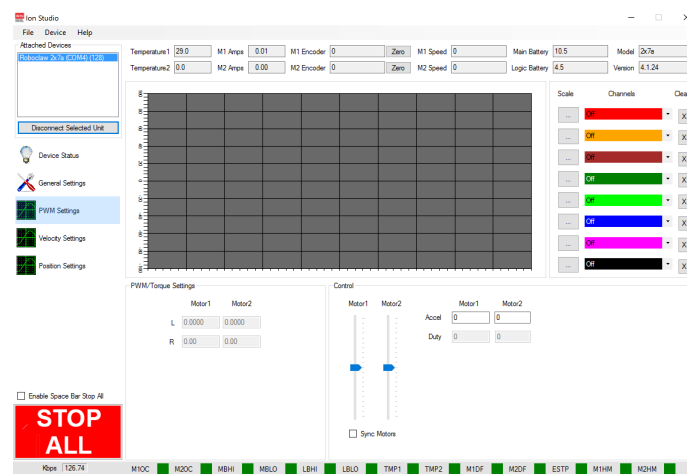


Figure 5

Verify that when the forward signal is sent to the motor (the Motor1 slider is above 0), the wheel spins in the direction that would move the rover as a whole forward (note this is different clockwise vs counterclockwise based on which wheel you are testing). **If the wheel moves backwards with respect to the rover direction, switch the motor leads coming from the motor controller, which are M1A and M1B.**

(c) Now as you drive M1 motor forward (which now corresponds to the rover moving

forward), verify that M1 Encoder value increases<sup>1</sup>. If the encoder value decreases, you have to switch the A and B channels of the encoder signals. These are the EN1 +/- pins on the header pins.

(d) Repeat steps 7b and 7c for M2 motor.

8. Once both motors are spinning the correct direction and the encoders respond accordingly when commanded through the PWM Signal tab, move on to the *Velocity Settings* tab.

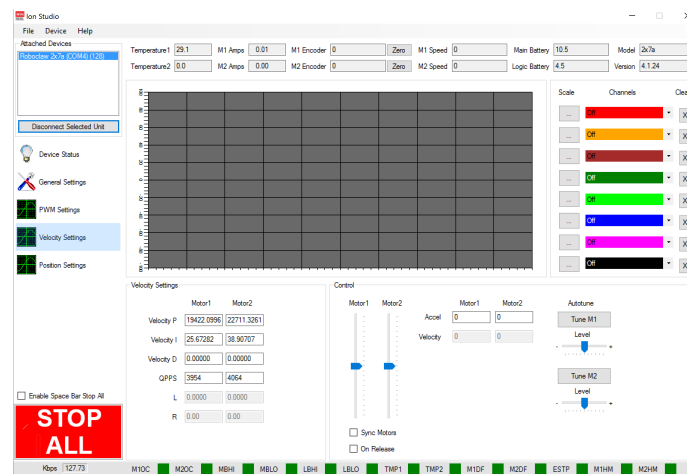


Figure 6

Here we'll be tuning the motor settings. We need to do this in order to have fluid control of the motor, and it needs to be done for each individual motor as they will all differ slightly out of the box. Therefore, each motor must be individually tuned. For more information on motor PID controls and what they do you can visit:

- <https://medium.com/pollenrobotics/an-introduction-to-pid-control-with-dc-motor-1fa3b26ec661>
- <https://www.elprocus.com/the-working-of-a-pid-controller/>

(a) Begin by entering in the QPPS (quadrature pulses per second), or the maximum

<sup>1</sup>If you get no reading values for the encoder, re-check your wiring and connections

number of encoder ticks per second. To find this value, drive the motor at max speed in the PWM setting and look at the box labeled 'M1/M2 Speed'.

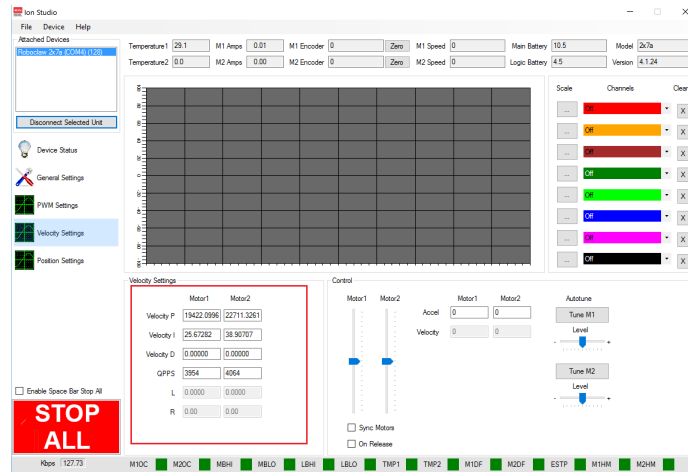


Figure 7

- (b) Now hit the Tune M1 button. This will populate the *Velocity P*, *I*, and *D* variables on the M1 side. Repeat this step for M2 motor as well.

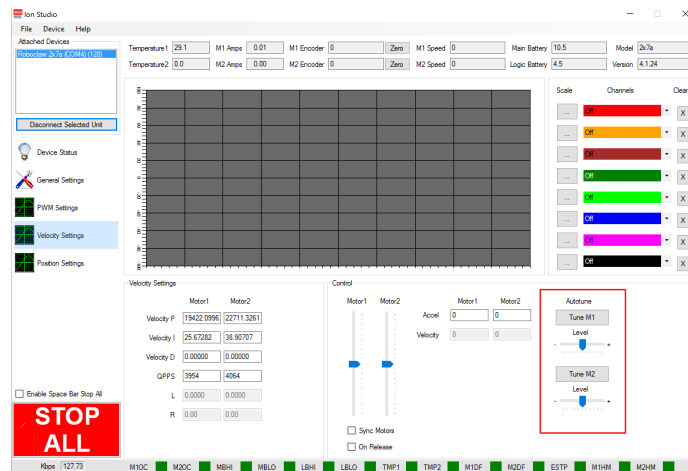


Figure 8

- (c) Once both motors on this motor controller have been tuned, go to *Device* – *> Save Settings* and then exit. Repeat all of steps 6 through 8 for the other drive motors.



## 1.3 Corner Motor Calibration

9. Perform the following for the **RoboClaw** addresses **131** and **132**:

- (a) Perform step 6 above for the corner motors, making sure to set the encoder type to absolute.
- (b) Go to the *PWM Settings* tab and **slowly** drive the M1 corner forward. Verify that the encoder value goes up. If it does not, switch the motor wire leads at M1A and M1B.<sup>2</sup>
- (c) Repeat step 9b for M2 motor
- (d) You may need to set the absolute encoder "zero" point in a different location. In order to avoid "clipping"<sup>3</sup> of the range of the encoder, we'll set the mid-point of the absolute encoder to be the middle of the turning arc.
  - i. Navigate to the *PWM Settings* tab to move the corner to its "center" position, which is where the wheel is directly forward.
  - ii. Unscrew the small gear at the bottom of the absolute encoder
  - iii. Once the absolute encoder shaft is uncoupled from the robot, spin the encoder until you determine the max value. This should be somewhere in the range of 1400-1800<sup>4</sup>.
  - iv. Spin the encoder until it reads a value of roughly half its max value and reattach the gear, coupling it back to the rover.
- (e) For each motor M1 and M2, we now need to obtain what is the new min and max encoder values. Using the *PWM Settings* tab, move the motors until they run

---

<sup>2</sup>As opposed to the drive motors, the corners use absolute encoders that are defined in one direction only, so you cannot switch the direction of the encoder

<sup>3</sup>Clipping would be where you go past the 360 deg mark on the encoder and it starts again at 0

<sup>4</sup>If you are getting encoder values of close to 2000 and the value stays above 2000 and does not reset to 0 while the shaft is spinning, it means that the voltage divider board is not working properly. Check to make sure the resistors on the voltage divider board are the correct values

into the physical hard stops. Record the min and max encoder values for each, but for the min values add at least 15 and the max subtract at least 15<sup>5</sup>.

- (f) Perform the Velocity tuning for each of the corners similar to the drive motors. You will have to estimate the QPPS value for the corners as you cannot spin them at full speed without running into the physical hard stop very quickly.
- (g) Navigate to the *Position Settings* tab.

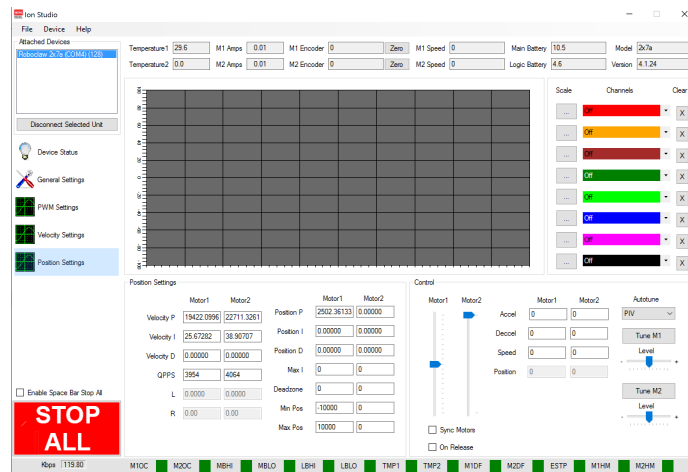


Figure 9

- (h) Enter your Min and Max encoder values for each M1 and M2 motor

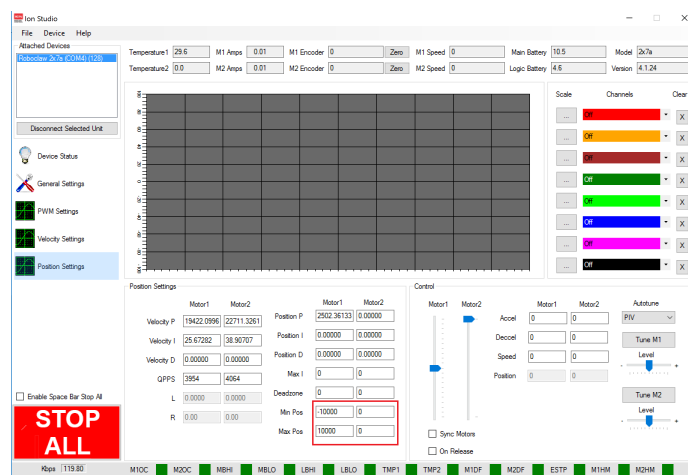


Figure 10

<sup>5</sup>We add a buffer of at least 15 encoder ticks to avoid driving the motors all the way into the hard stop; In other words, we'll impose a software 'soft stop'

- (i) Change the Autotune type to **PID** and then tune the M1 and M2 motors by clicking the 'Tune M1' and 'Tune M2' buttons. We've found that this autotuning process is more successful if you run it with the wheels down on a smooth surface, but still supporting the weight of the robot.

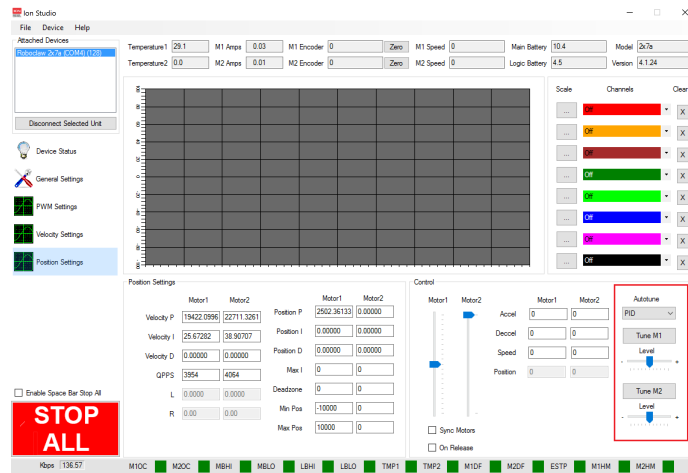


Figure 11

- (j) To verify that the corner steering motors have been calibrated, move the M1 and M2 sliders to their middle positions (halfway between min and max) and verify that the wheels all point exactly straight forwards. If they do not point straight forwards, verify that your min and max values are correct and ensure that no signal clipping is happening on the encoders.
- (k) Repeat Step 9 for the remaining corner motors