

ExoActuator

Operations Guide

Abdulrahman Alshammari

Alex Frieden

Chancelor Cuddeback

Callum Fisher

Joshua Davidson

2021



Project Sponsor: Biomechatronics Lab at NAU

Faculty Advisor: Dr. Oman

Sponsor Mentor: Dr.Lerner

Instructor: Dr. Oman

DISCLAIMER

This report was prepared by students as part of a university course requirement. While considerable effort has been put into the project, it is not the work of licensed engineers and has not undergone the extensive verification that is common in the profession. The information, data, conclusions, and content of this report should not be relied on or utilized without thorough, independent testing and verification. University faculty members may have been associated with this project as advisors, sponsors, or course instructors, but as such they are not responsible for the accuracy of results or conclusions.

Using the Arduino Code:

The Arduino Code features a simple implementation using the MotorController class. The implementation code has been well commented. If you would like to change the ID of the motor that you are using you should change the define at the top of the page. The implementation code is used independently of the iOS app. There is a separate script intended to be used with the iOS app.

Configuring the Motors via Teraterm:

Teraterm may be installed [here](#).

1. Connect the TMotor to an FTDI board using the serial cable. (Rx to tx, tx to Rx, Vin to 5v and GND to ground).
2. Connect the motor to a 24V battery, if there is not a connector available you will need to solder one.
3. Connect the FTDI board to your computer via usb cable.
4. Run Tera Term
5. Select Serial (The port should be automatically selected)
6. Press OK
7. In the window labeled with the aforementioned port, Navigate to Setup->Terminal and (other than the CodePage) configure the terminal as you see in **Figure 1**.

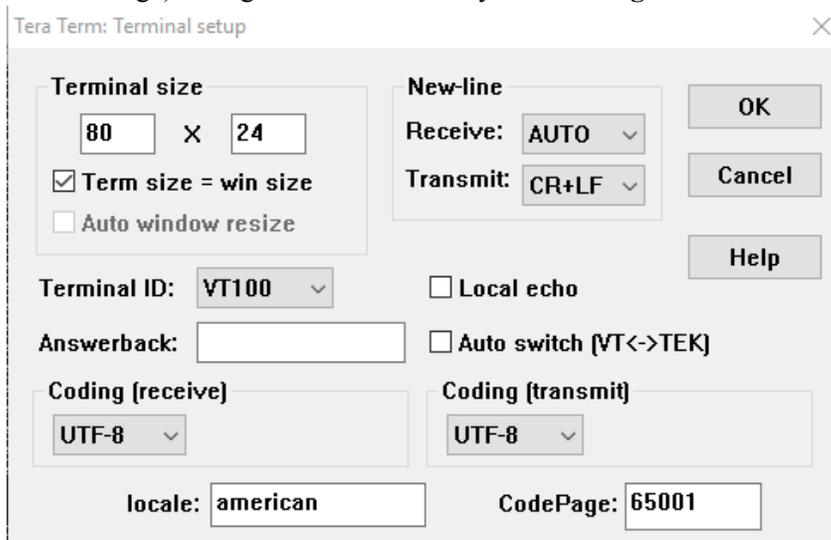


Figure 1: Terminal Config

8. Press 'OK' to close the terminal setup window.
9. Now navigate to Setup->Serial Port and configure your window as you see in **Figure 2**.

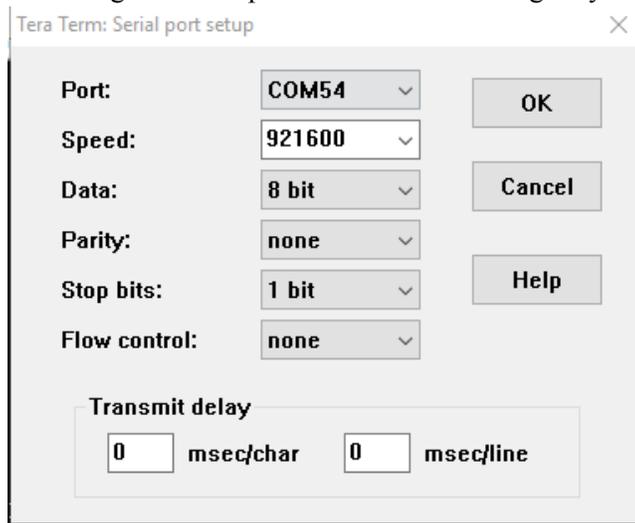
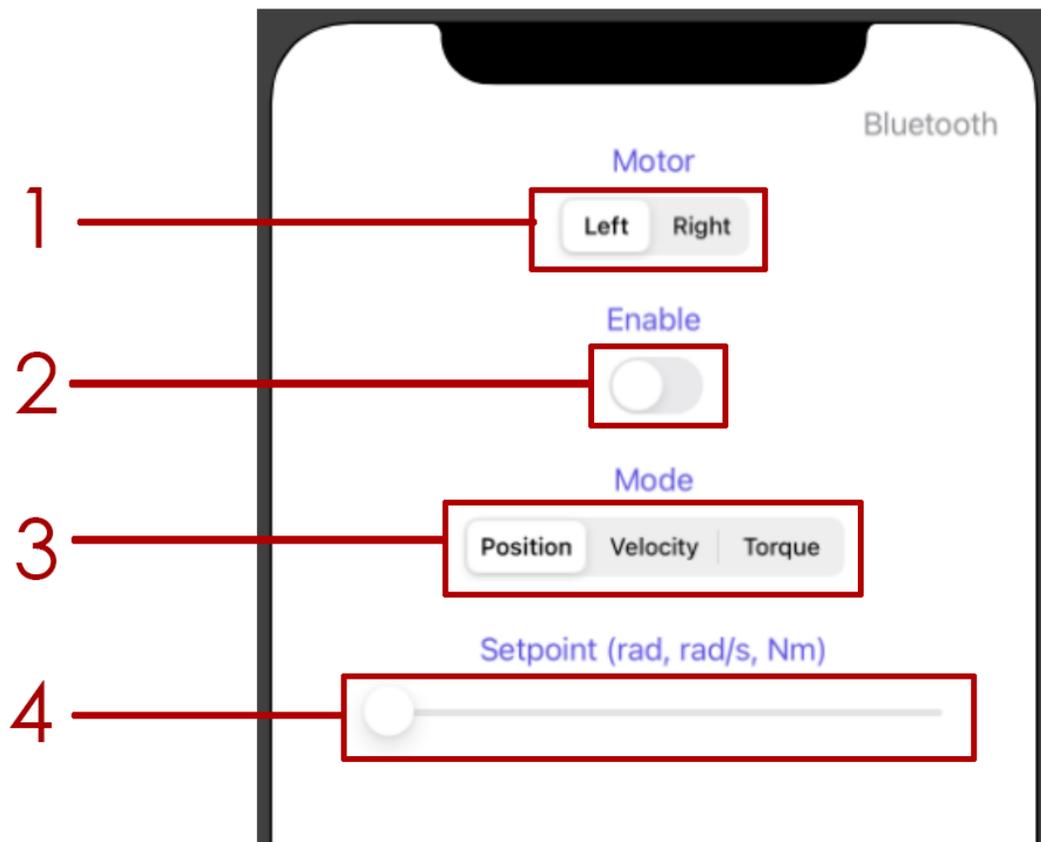


Figure 2: Serial Port Config

10. As before, press 'OK' to close the window and save your changes.
11. Select the Normal terraterm window (Black Screen) such that the windows cursor is blinking white.
12. Press the escape key on your keyboard and you should be met with a prompt.
13. If you are not met with a prompt power cycle the motor and press the escape key again.
14. To configure the motor press s.
15. Then modify the CANID such that it is unique in your pool of motors.
16. Feel free to calibrate the encoders!

The iOS app:



1 - Selects Motor (This design contains two, a left and right motor)

2 - Enable/Disable motor mode (This allows the motor to move/receive input)

3 - Selects Input

Position - Sets position of the motor based on radians, starting at 0 and ending/restarting at 2π

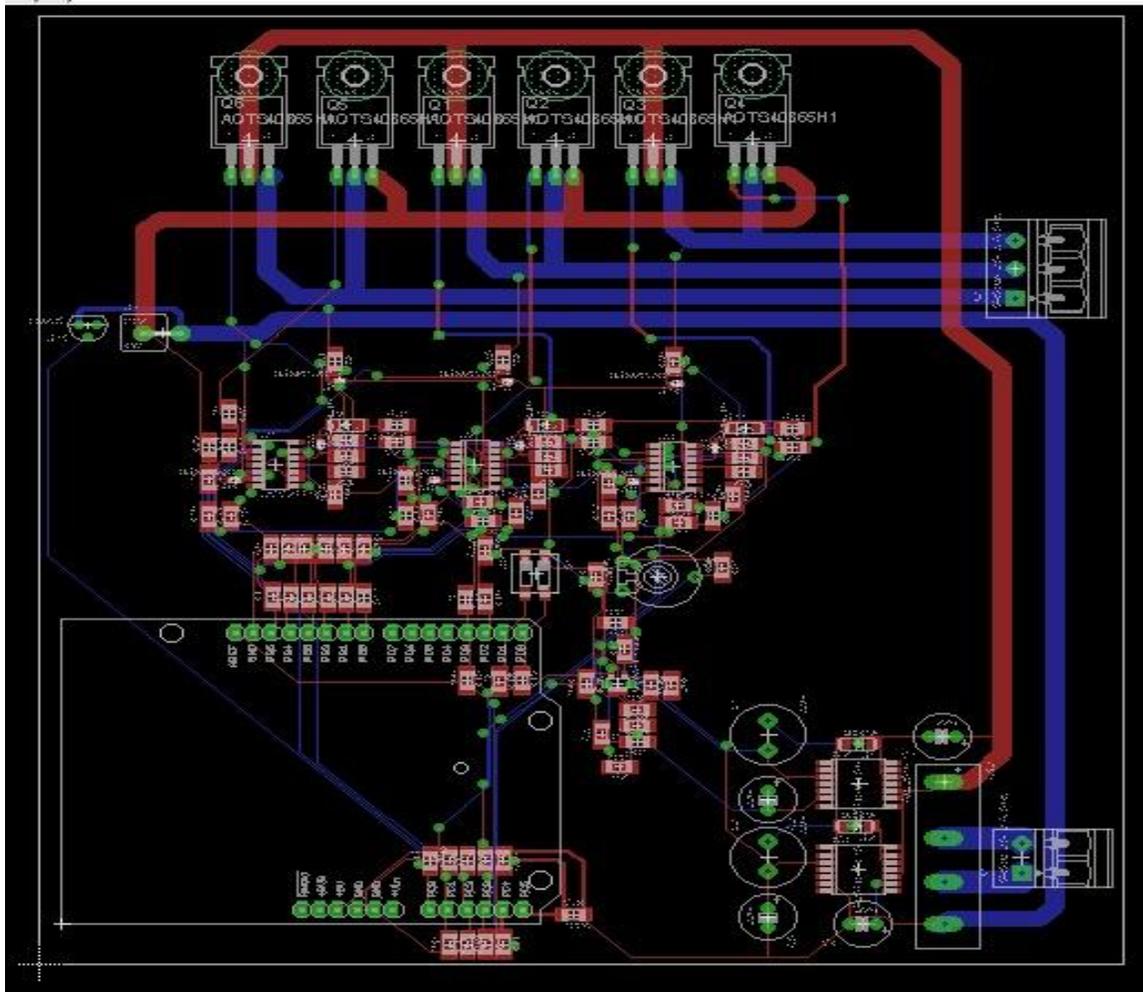
Velocity - Sets the velocity at which the motor spins in rad/s

Torque - Sets a torque value for the motor in Nm

4 - Sets the input value (in rad, rad/s or Nm as shown above the bar)

App connects to the motors via bluetooth (as indicated top right of the app)

PCB



The core of the design is the commutation device, we will use the AOTS40B65H1 which is a 650V -40A IGBT transistor, Case TO220. Six transistors (Two to control each coil) can supply each coil by the desired voltage based on the control signal generated by the microcontroller. At each phase, voltage of each coil is measured (after voltage scaling to make voltage in the range of the A/D of the microcontroller). In BLDC motors driven by trapezoidal control, the DC bus current is equivalent to the winding current. To measure the DC Bus Current we use a very small resistance (0.02W) that can support high current (30Watts). A simple Operational Amplifier with high bandwidth and high slew rate (OPA374) will convert this current to voltage in the range of the A/D. The connectivity between the signals generated by the microcontroller and the IGBT transistor is assured by the Half Bridge Gate Driver UCC27714D. A temperature sensor LM235 is used to monitor the temperature of the heat sinks needed for the transistors and the current measure resistor, to prevent overheating. As a Controller Arduino Uno , that contains 6PWM signals, 6 Analog Input, and 8 general purpose Input Output pins. This controller will generate the desired control signals for the IGBT transistors, monitor different sensors inputs (Current, Speed, temperature, Voltages). Finally, the Power Supply module is to convert the 48V input to +5V to supply the Controller and OP-Amp. And +15 V to supply the Gate driver of the control signals. The conversion is based on the LM2574HVM-5.0 and LM2574HVM-15.