Instructions for Wiring and Using the Probe Motor EPPDyL

Johan Sweldens

August 12, 2020

1 Background

This is the motor setup that will be used by the Electric Propulsion and Plasma Dynamics Laboratory to perform a variety of experiments on plasma sources. To learn more about the purpose and design process of the probe setup, please refer to the work of Jordan Brown '18 (2) and Sydney Hsu '21 (1). The purpose of this packet is to provide information on how to set up and use the motor. Regrettably this was not able to be finalized back in the Spring of 2020 because of the COVID-19 pandemic. This is why the author, Johan Sweldens, took on the role of completing the software hardware interface during the Summer of 2020 to ensure that it will work by the time lab based research resumes again at the University. Feel free to send any questions to jws426@cornell.edu

2 Hardware

Here is a list of all the components that allow the motor to work. Please note that this is before it has been shielded and placed within the hollow cathode setup.

Stepper Motor: Bipolar, 200 Steps/Rev, 35x28mm, 10V, 0.5 A/Phase `https://www.pololu.com/product/1208` This is the motor itself. It operates on a basis of 200 steps per one revolution. It draws 500 mA at 10V. The way it works is by having two different coils in its wiring. This gives it very high accuracy for position and velocity, but at a trade off of the motor having to be using energy to maintain its position. The motor will get hot if left on for more than two minutes. It is recommended to turn it off using the 'Stop' function which will de-energize the motor. Otherwise, adequate cooling is a must in order to preserve the long life of the motor.

Wall Power Adapter: 9VDC, 3A, 5.5x2.1mm Barrel Jack, Center-Positive `https://www.pololu.com/product/1464` This component is what provides the power to the motor and the board during operation. It converts alternating the voltage from the outlets to 9V which is what the board and motor can operate on.

DC Barrel Jack to 2-pin Terminal Block Adapter `https://www.pololu.com/product/2449` This is the adapter from the outlet style jack to the two basic wire setup that can easily be connected to the Tic Controller. An electronics screwdriver is necessary to ensure a good connection.

Tic T500 USB Multi-Interface Stepper Motor Controller (Connectors Soldered) `https://www.pololu.com/product/3134/resources` This is the board that controls the motor. It has numerous methods of interfacing with a computer or microcontroller. The easiest way to do this is with a USB cable and the 'Tic Control Center' application which is free to download from the Pololu website. Alternatively one could use an arduino to operate the Tic Controller. The 'Tic Stepper Motor Controller User's Guide' is a great reference for anything regarding the board.`https://www.pololu.com/docs/0J71` The version of controller that is used by the lab has all of its pins soldered on and requires an electronics screwdriver to properly attach everything.

Mini Snap-Action Switch with 13.5mm Lever: 3-Pin, SPDT, 1A `https://www.pololu.com/product/1528` This switch is very small and has two output pins: Normally Closed and Normally Open. With this setup only the Normally Open pin and the input pin are used. The pins on the switch are very flat and stubby, this makes them very impractical as they cannot interface with dupont wires. The switch's pins can fit in a breadboard, but it must be held down in place. Wiring up the switch in the lab is likely going to be more difficult than anticipated.

DSD TECH SH-U09C2 USB to TTL Adapter Built-in FTDI FT232RL `https://www.amazon.com/DSD-TECH-SH-U09C2-Debugging-Programming/dp/B07TXVRQ7V/ref=sr_1_4?dchild=1&keywords=usb+to+ttl+adapter&qid=1594344550&sr=8-4` The adapter is what allows the computer to communicate with the Tic T500 using serial communication ports. This is very vital to being able to run the program and in fact, it is impossible to do so without it. Current versions of the Tic T500 do not have the capability to communicate with a computer on their own as they lack the FTDI chip. The adapter can work with Windows 7 through 10 and multiply Linux and Mac operating systems. Additionally, the adapter has three voltage settings: 5V, 3V, and 1V. It must be set on 5V because the Tic T500 can only operate between 4.5 and 35V. The Tic T500 will actually have problems with its logic if the adapter is not on the 5V setting. The setting is determined by where the short is located. Additionally, a good way to ensure the RXD and TXD pins are working is by placing a short on those two components. The LEDS should light up. The adapter comes with its own set of 6 conjoined dupont connectors. These are very useful.

3 Hardware Setup

a. To test that the Tic T500 is functioning properly and not damaged from shipping, use the USB to USB micro cable to connect it to a computer. The green LED should turn on signalling that it is sending and receiving information via the USB. The red should turn on signaling that an error is happening. This is normal as the power supply has yet to be attached. The yellow LED should be blinking at 2Hz. This signals that there is no power supply. Once the board has been inspected and ensured that it will function properly, unplug the USB to USB Micro cable. You will not be using that cable in this setup.

b. Now connect the stepper motor to the Tic T500 board. This is done by connecting the blue wire to the B2 on the board, red wire to B1, green wire to A1, and black wire to A2. These are secured in place by turning the corresponding screws on the top of the buffer. A micro electronics screwdriver makes this job significantly easier.

c. Connecting the barrel jack adapter is roughly the same process. It is important to look carefully at the markings on the top of the adapter to see which terminal is positive and negative. Connect the positive terminal to the VIN pin and the negative terminal to the GND

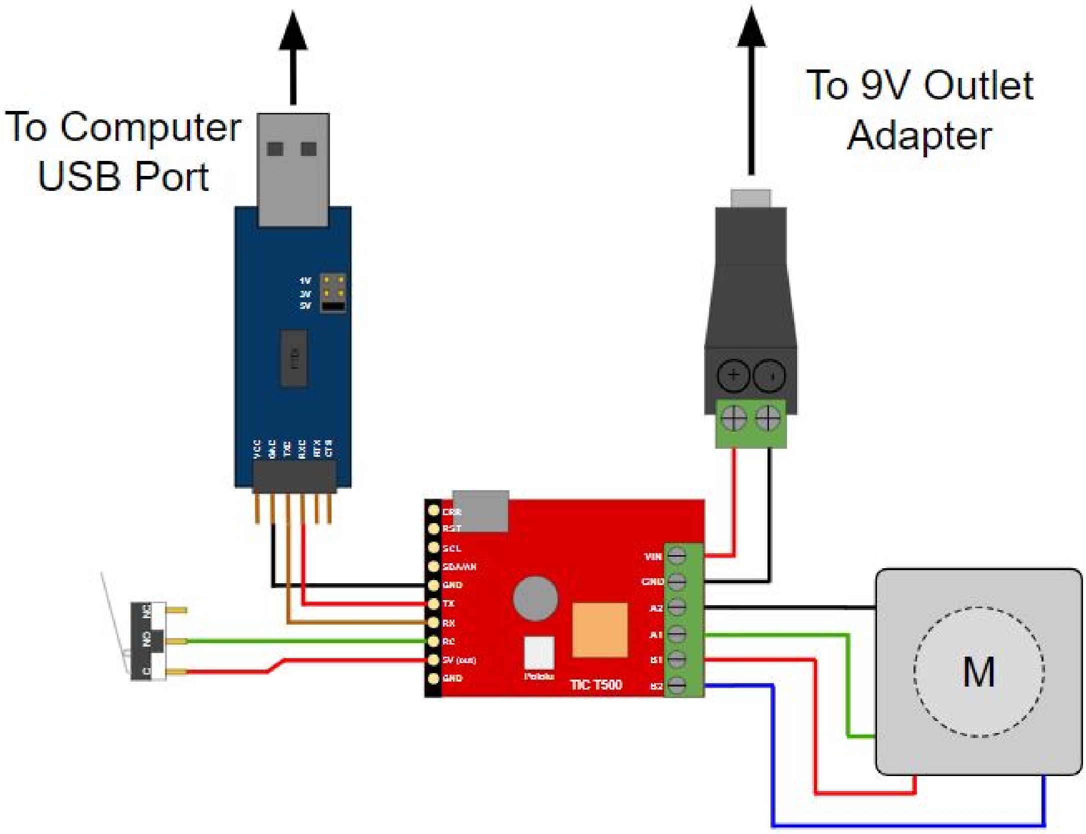

Figure 1: Wiring Schematic

pin. These are the top two pins on the green buffer on the right side of the Tic T500 board. Note that while not critical, red positive and black negative color correspondence is always appreciated. These connections are secured with the same type of screw mechanism on both parts. An electronics screwdriver is recommended.

d. The next step is to attach the USB to Serial TTL adapter to the Tic T500 board. The USB to Serial TTL Adapter is the long blue board in the protected case. It is extremely important that the short switch is on the 5V setting on the board. There are 6 pins that stick out of the protective case that the adapter board sits in. The switch must be on the leftmost setting which is the 5V. Please refer to the pictures of the setup to see how this is done. Normally this be left in this position and will not have to be changed. The adapter comes with a 6 pin dupont connector. This works really well with the board, but I suspect that its limited length will cause problems in the setup within the experiment. The 6 wire dupont connector can be disregarded if one has a better setup. Anyways, the GND pin on the USB to Serial TTL adapter (blue board) must be connected to the GND pin on the Tic T500 (red board). Note that while not necessary, It makes for a much more compact and less cluttered setup if you use the GND pin next to the TX and SDA/AH pin on the black buffer on the left. Next the TXD pin on the USB to Serial Adapter (blue board) must be connected to the RX pin on the Tic T500 (red board). Also, the RXD Pin on the USB to Serial Adapter (blue board) must be connected to the TX pin on the Tic T500 (red board). The TXD pin is what sends data from the computer to the Tic T500 where it receives it on the RX pin. This is also done in reverse where the Tic T500 sends data to the computer via the TX pin to the RXD pin on the adapter.

e. The last component that needs to be wired up is the limit switch. Based upon settings within the Tic T500, the limit switch must be configured to the normally open setting. This is done by connecting the normally open pin (the middle one) to the RC pin on the Tic T500 board. The RC pin is located underneath the RX pin. The RC is designed to be used with a remote controller, like an analog joystick. However it works for this purpose. The 5V (out) pin on the Tic T500 (located directly underneath the RC pin) must be connected to the C pin on the limit switch. The marks on the switch are hard to read, but it is the pin on the side of the switch where the little lever is attached. It is very important to ensure a solid connection between the limit switch and the Tic T500.

f. To get power into the motor, the wall adapter must be connected to an outlet and the barrel jack must then be connected to the barrel jack adapter. If everything is connected as described and the outlet is providing power to the system, a variety of LEDS on the Tic T500 and the USB to Serial Adapter will light up. The RED LED will be on solidly on the Tic T500. The yellow LED will blink twice at 4HZ and then turn off for 0.5 seconds on the Tic T500. This indicates that the motor is de-energized but the system is properly powered. The USB to Serial adapter (which is yet to be plugged into a computer) has the TXD and RXD LEDs light up solidly indicating that they are ready to send and receive data. If this is not the case, something might not be wired up correctly.

g. The last physical step is to plug the USB to Serial adapter into the lab computer's usb port. From here the next steps are all to be done on the computer.

4 Software Setup

a. The next steps continue from the hardware instructions. The first step is to log onto the lap computer and ensure that python has already been installed. Open the command prompt or terminal on Mac and enter in 'python'. This shows the current version installed. The program is designed to work on version 3.8.5 but will likely work on earlier versions so long as it is still python 3. If python has not been installed, go to this website to download the latest version . `https://www.python.org/downloads/`.

b. Once python is installed and verified to be a compatible version, type in and enter 'pip install serial' and 'pip install pySerial'. The terminal will either give the download process or it will give back the current version of these installs. pySerial is a addon to serial that allows for the use of communication ports on a computer. Additionally you need to enter in 'pip install numpy'.

c. The USB to Serial TTL adapter has a driver that need to be installed in order for it to function correctly. Go this this website `https://www.ftdichip.com/Drivers/VCP.htm`, and look for the 'Currently Supported VCP Drivers:'. Assuming Windows is being used, look on

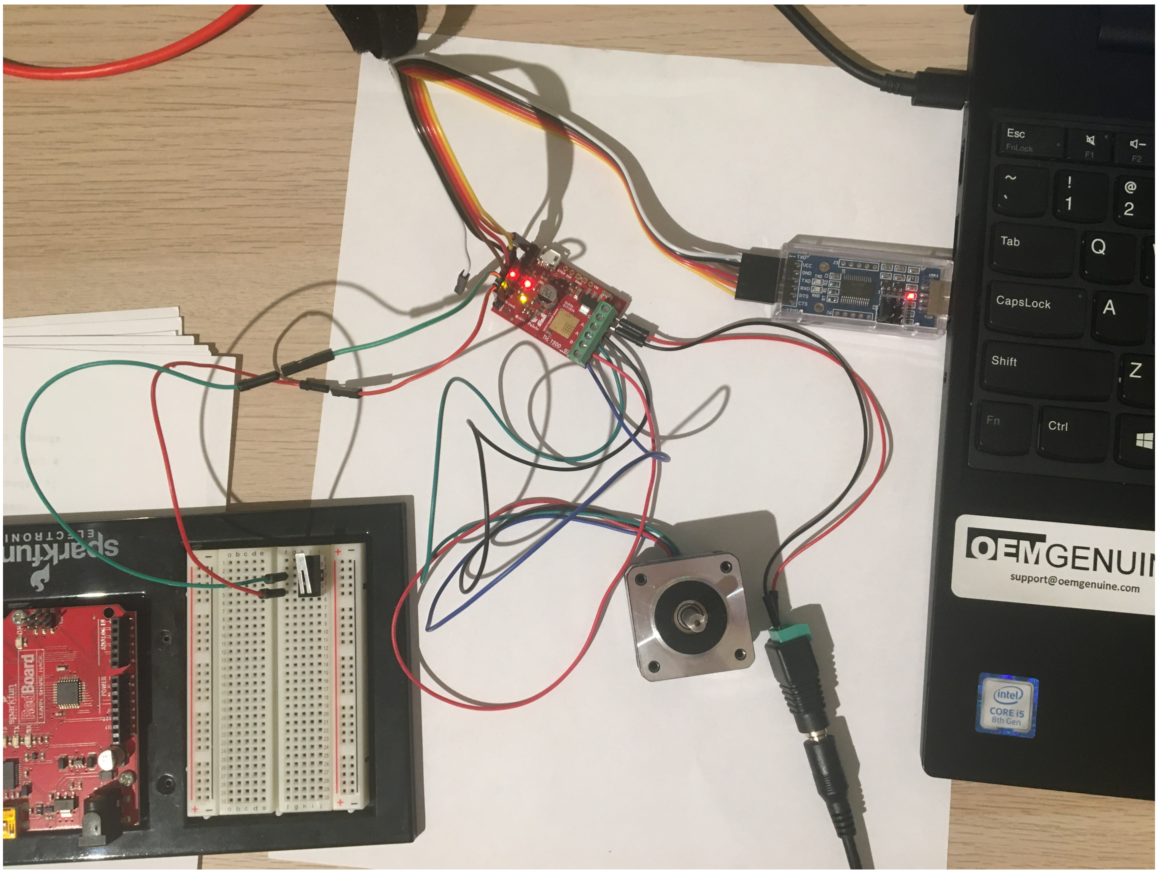

Figure 2: Picture of Hardware Setup

the right side column 'Comments' and click on the 'setup executable' link to download and automatically install the appropriate drivers for the computer. If the computer being used in the lab setup is not a recent version of Windows, there are other available drivers for the corresponding operating system. It is important to ensure that the driver is installed properly or the serial communication with the computer will not work.

d. Enter in `'python -m serial.tools.list_ports.'` once all the packages have been installed, If everything has been wired correctly and plugged in, then the command prompt will list the communication ports being actively used. As seen in the picture below, my computer, which runs Windows 10, says that 'COM4' has been found. This means that there is a virtual communication port that can be used to send information to the motor. It is important to note that all computers will likely say something different. Other Windows computers will likely give a different number to the communication port. If it is a holder machine that actually has an old fashioned serial port on it, like for old printers, then it might list that as well. If there are multiple serial ports listed, try unplugging the Serial to TTL adapter and see what ports are listed then. The missing one is the port that the adapter uses. Linux and Apple computers will list something entirely different. Most commonly used is '/dev/ttyACM0' and

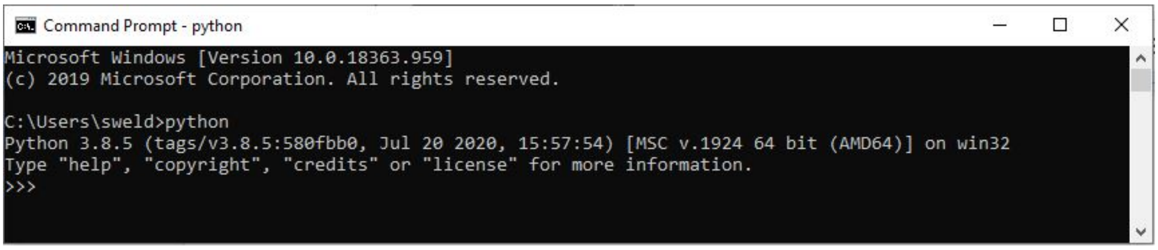

Figure 3: Python Version

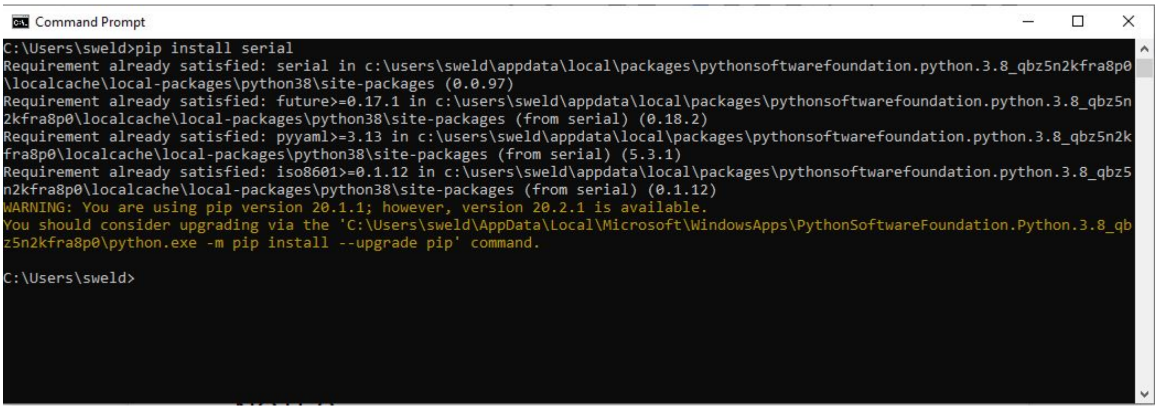

Figure 4: Installing pySerial

'/dev/ttyUSB0'. Once again the numbers might change but this is generally what they will look like. From my experience testing this on an old mac I have, they always have some form of a bluetooth serial virtual port being utilized. I'm not entirely sure that is but be careful to correctly identify which port is being used by the adapter. It is crucial to write down or remember exactly what the name of the port being used is. This will come into play later.

e. Download the program from github at `https://github.com/sweldensj/EPPDyL-Motor` or gain access to the file if it was already downloaded on the computer. The program is labelled '`tic500_control_gui2.py`'. Open the python file in a text editor or python api. I used Sublime to do this. The most default way to edit the file is in notepad or some other code editor. Go down to line 104 and change the string for `port_name` to fit whatever the communication port that is being used by the adapter as seen in the previous step. As seen below the computer uses "COM4". There are other settings included in that section: baud rate, device number, default home value, maximum motor speed, calibration time, and calibration offset. You can leave these things alone for now. These are meant to be changed later if something does not fit with the physical setup. Save once the changes are made and exit out of the file.

f. Once all the previous steps are complete, open and run the program '`tic500_control_gui2.py`' by either double clicking on it or by copying its directory and entering the path address as shown below. If there are no issues the gui will pop up and the motor is ready to be used.

```
Command Prompt
C:\Users\sweld>python -m serial.tools.list_ports
COM4
1 ports found

C:\Users\sweld>
```

Figure 5: Using the listports function

```
#---------------------------------------------------------------
    #SETTINGS
#---------------------------------------------------------------
#Choose the serial port name. !!!!THIS NEEDS TO BE CHANGED DEPENDING ON THE COMPUTER!!!!!!
#Windows use "COMx" where x is a number between 2 and 5. Most common is 3 and 4 ex "COM3"
#Linux and Macs use things like "/dev/ttyACM0" or "/dev/ttyUSB0"
#Once everything is settup use this function in command prompt to see which port you need to use
#   python -m serial.tools.list_ports
port_name = "COM4"
```

Figure 6: Changing the Communication Port Name

The RXD and TXD LEDS on the adapter will be blinking. The red LED on the Tic T500 will steady as power is not directed to the motor and the yellow LED will be blinking three times at 6hz at then alternate off. The yellow LED indicates that the motor is de-energized and is receiving signal via serial commands from the computer.

5 Using the Motor

The GUI has seven 7 functions: Connect, Forward, Reverse, Stop, Set speed, Go home, and Calibrate. Additionally it has a read out on whether or not the motor has been connected, the port address being used, the step size, current speed, and current position in steps.

Connect This button needs to be pressed before any other function or else the motor won't move. The connect function will attempt to energize the motor. This means that current is now flowing through the coils of the motor and it is ready for use. This can be detected by an audible hissing sound. Even though it sounds like a WW2 dive bomber, do not be alarmed, this is as expected. The motor will get hot from this point onward so be careful to not leave it on for too long.

Forward This function moves the motor by the imputed steps. The step size is imputed by typing

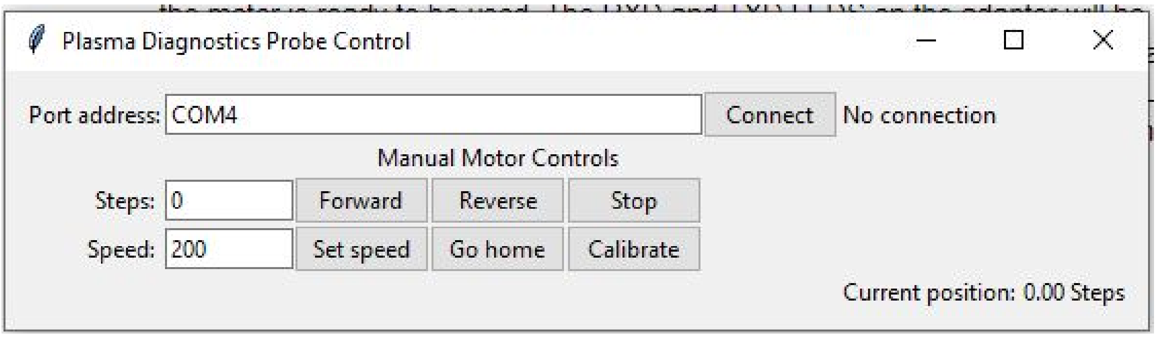

Figure 7: The GUI upon being opened

in the numbers in the box next to the Step. There are 200 steps per revolution so changing the default value of 0 to 200 and pressing Forward will spin the motor exactly 360 degrees.

Reverse The reverse function does basically the same thing as the Forward function but in the opposite direction. The reverse direction is limited by the limit switch. If the switch is pressed, the motor will not turn in the reverse direction.

Stop The stop function does two things. It first acts as the motors break bringing it to an immediate halt. The second thing is to de energize the motor once the break is complete so that it is no longer drawing current to maintain position. It is recommended to use this function liberally as the motor can get very hot very quickly by just staying still once energized.

Set Speed This function takes the input on the left and sets that value as the speed of which the motor moves at. All other functions except for calibrate will move at this speed. Note that the program has a maximum speed hardcoded in so that the motor is not damaged. The recommended maximum speed is 200Hz. This is equal to 200 steps per second or one revolution per second.

Go Home This brings the motor back to its home position. By default it is set to the middle once started. Using the calibrate function will reassign a home value and the go home function will move to that new home value.

Calibrate The calibrate function works by slowly moving the motor in the reverse direction until the limit switch is pressed. From there it moves 1/10nth of a revolution forward to give some room before the limit switch. It sets home value to the new position so that the Go home function will bring the motor back to the calibrated position. Additionally it changes the current position read out to reflect this change.

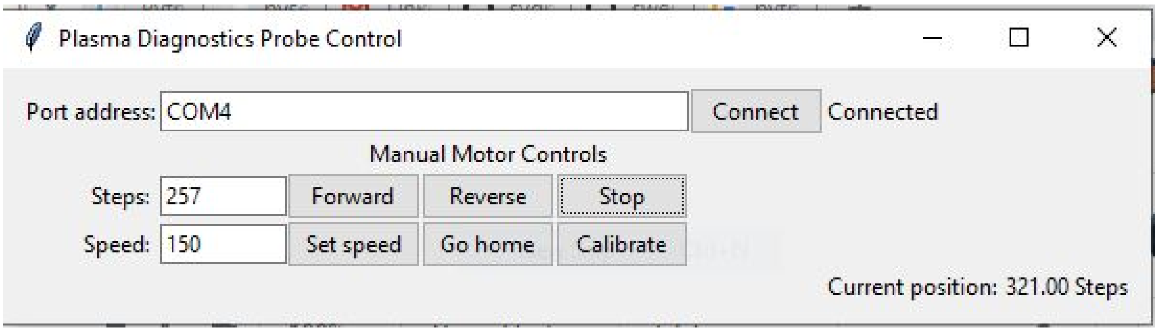

Figure 8: The GUI while in use

6 Notes

Tic Control Center The best way to deal with any suspected hardware difficulties or to ensure components are working is to download and use the Tic Control Center from `https://www.pololu.com/docs/0J71/3.1`. An issue that may arise is that a new computer might have to download the drivers for this software even though the motor is only communicating with the serial adapter. To use the software, plug the board into the computer using a USB to USB microcable and not the USB to Serial TTL adapter. The board will light up with a green light indicating USB data is being transferred. This application can change a variety of settings that might need to be altered during the course of using the motor.

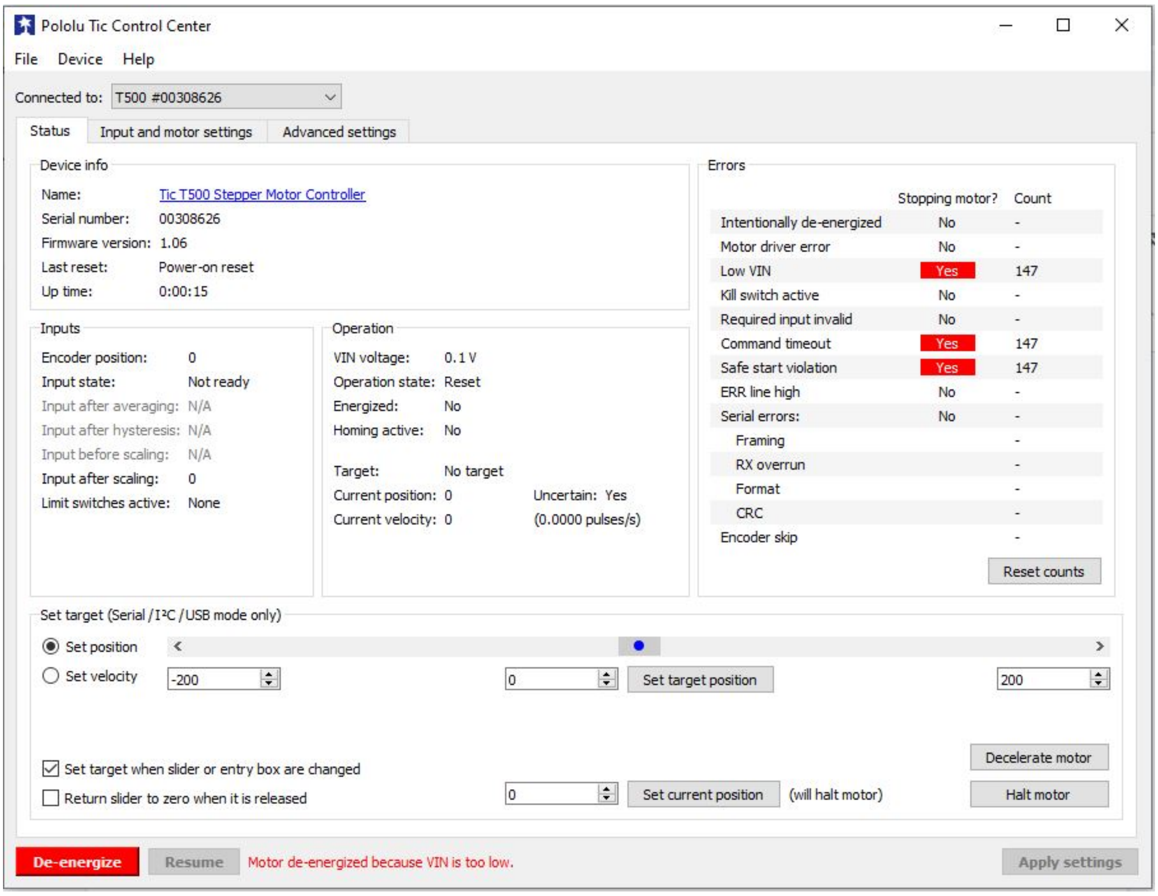

Figure 9: Tic Control Center

Command Time Out All Tic T500s have a default command timeout of one second. That means that after one second after a command has been sent, the motor will stop moving. This is an issue if one would try to do more than 200 steps at 200 steps per second. This setting was changed to be 5 seconds. That allows for 5 full revolutions before the motor is 'timed out'. This is a safety feature built in to the motor. This might get changed when using a new computer so if this is noticed to be an issue, go into the 'Input and motor settings' tab in the Tic Control Center application and change the 'Enable Motor Timeout'. In the code for the go home function, I added a loophole to always circumvent this issue. No matter how long it takes to get back to the home position, the motor will see it the way through.

Limit Switch Setup and Concern The active high setting in the application is used and the 'NO' pin is used to ensure that the limit will be detected when pressed against. I am not entirely sure but I believe that these settings could be reversed and nothing would change. As in the 'NC' pin with the deactivated high setting would be equivalent.

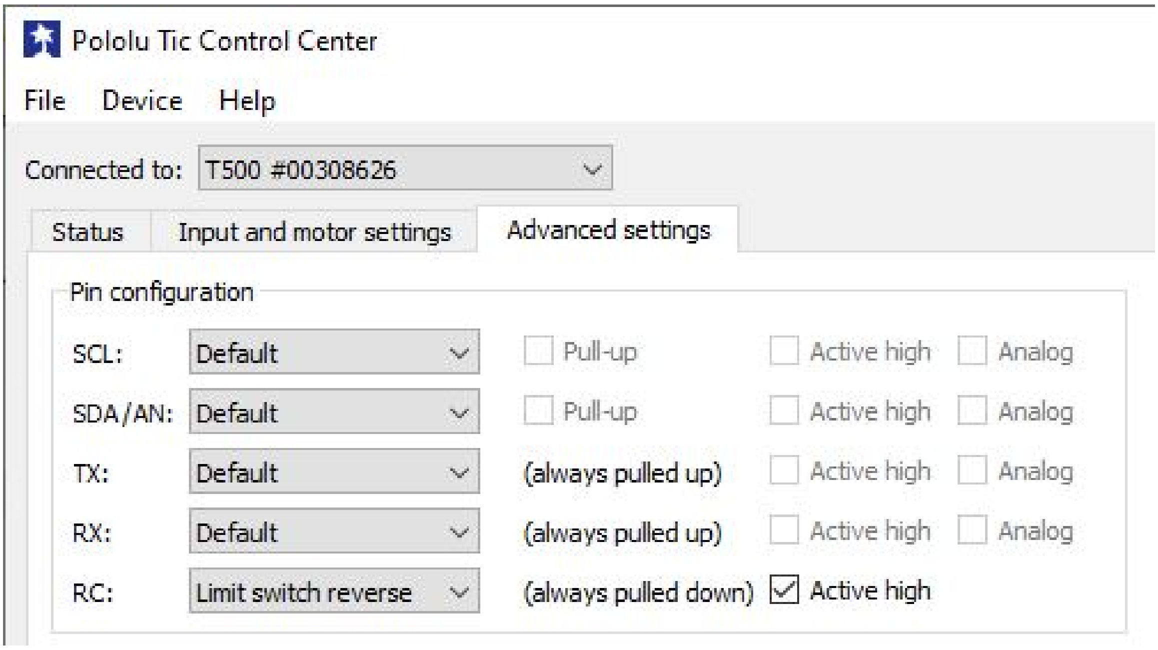

Figure 10: Settings for the Limit Switch

To change the direction of which the limit switch acts on, go in the 'Advanced settings' of the Tic Control Center. The RC Pin is configured to Limit Switch Reverse and change that to Limit Switch Forward if the direction needs to get changed. The 'Calibrate' function in the code would have to be changed though since it only calibrates in the reverse direction. Change the minus sign in line 271 to a plus sign and the plus sign in line 282 to a minus sign to achieve this.

The limit switch provided in this setup seems highly problematic. The pins are not long enough to interact with female dupont connectors. Additionally they only fit into a breadboard when they are held down. This is very frustrating. Any setup in the lab would require soldering the pins to connectors because of their impracticality.

Because of the small size and poor pin quality, I would recommend in the future changing this limit switch out with a larger higher quality one. Ensuring that the probe does not crash into the walls of the cathode is pretty important. That might be too much faith to put into such a small sensor.

Wire Clutter When fully connected the Tic T500 has eleven wires coming out of it. I am concerned that these wires might not fit into the shielding mount assembly in the motor cage as shown in Figures 3 and 5 of "Developing an Actuated Probe Suite for Collecting Plasma Diagnostics". The rap around shielding would seem to collide with all of the cables on the left side of the board.

Additionally, a much long cable is probably needed for the USB to Serial TTL adapter for it to connect to the lab computer and the Tic T500, assuming that the two of them will not be inches away from each other in the lab. I would recommend finding a way to secure the adapter in place and then use a USB Female to USB Male extension cord to connect the adapter to the lab computer.

Calibration The way calibration was supposed to work in the old code is very different from how it works in this edition. The old program called for using two limit switches, inputting voltage, and saving a file onto the computer. This is probably a better method of calibrating, but given that only one limit switch is now going to be used means that the calibration method is much more simplistic. This function might have to be changed if the mission of the motor gets altered.

Debugging Communication Ports The only frustrating part of the process of preparing the motor was dealing with the computer's communication ports. The Tic T500 does not have the ability to communicate with a computer using serial commands by itself since it lacks the FTDI chip.

Additionally, figuring out issues with the communication ports is very difficult. If you need help, use the device manager application built into the computer. Opening the properties tab of the communication ports can reveal a lot of information like driver issues and the port address.

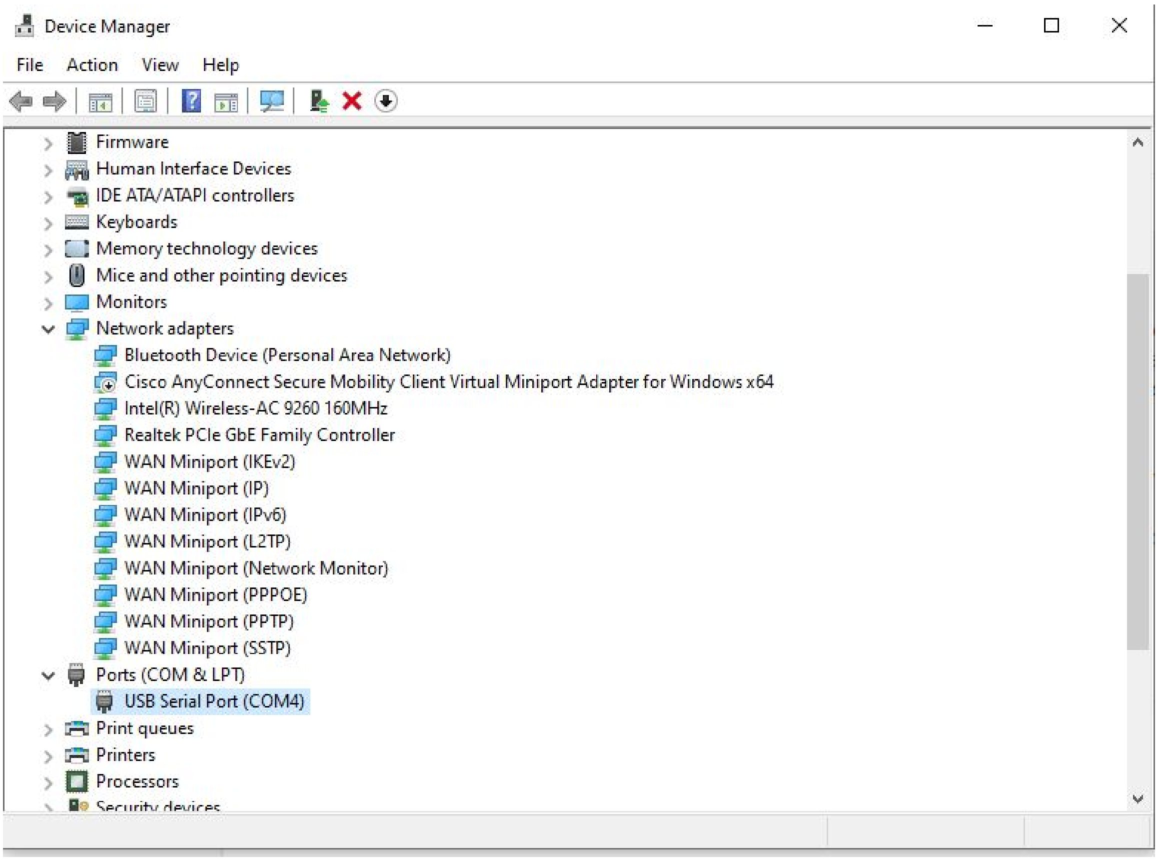

Figure 11: Device Manager

Helpful Resources Here is a list of websites which provide very useful information for dealing with some of the electronics and setting up the Tic T500.

Tic T500 LED guide `https://www.pololu.com/docs/0J71/3.4`

Configuring Limit Switches `https://www.pololu.com/docs/0J71/4.14`

Getting variables from the Tic T500 `https://www.pololu.com/docs/0J71/7`

Sending commands to the motor `https://www.pololu.com/docs/0J71/8#cmd-get-variable`

Setting up serial control `https://www.pololu.com/docs/0J71/4.5`

USB to Serial Adapter Drivers `https://www.ftdichip.com/Drivers/VCP.htm`

Pololu support support@pololu.com

References

[1] Sydney K. Hsu. *Developing an Actuated Probe Suite for Collecting Plasma Diagnostics*. Princeton University, Princeton, New Jersey, 2020.

[2] Jordan Brown. *Design and Implementation of an Actuated Probe Suite for an Orificed Hollow Cathode* Princeton University, Princeton, New Jersey, 2018