



G-ROM Mazda RX-8 EPAS Enabler Boost Adjustment Interface

User Guide

This manual will describe how to use the prototype boost adjustment interface on the G-ROM Mazda RX-8 EPAS Enable. If you have any questions, feel free to send me an email (garrettgloceri@comcast.net) or a message on the G-ROM Facebook page: www.facebook.com/GROMElectronics. I'm just one dude, so like.. please be patient on a response. I promise I will get to you!

How the G-ROM RX-8 EPAS Boost Interface Works

The G-ROM RX-8 EPAS Enabler has a prototype interface that allows you to adjust the steering boost amount. This works by sending the unit a command line type command to tell the unit the percentage of boost you would like from 0 to 100%. The unit will save this value to its memory. Upon startup of the device, the EPAS Enabler will read this value from memory, and output the amount of boost as defined by the user.

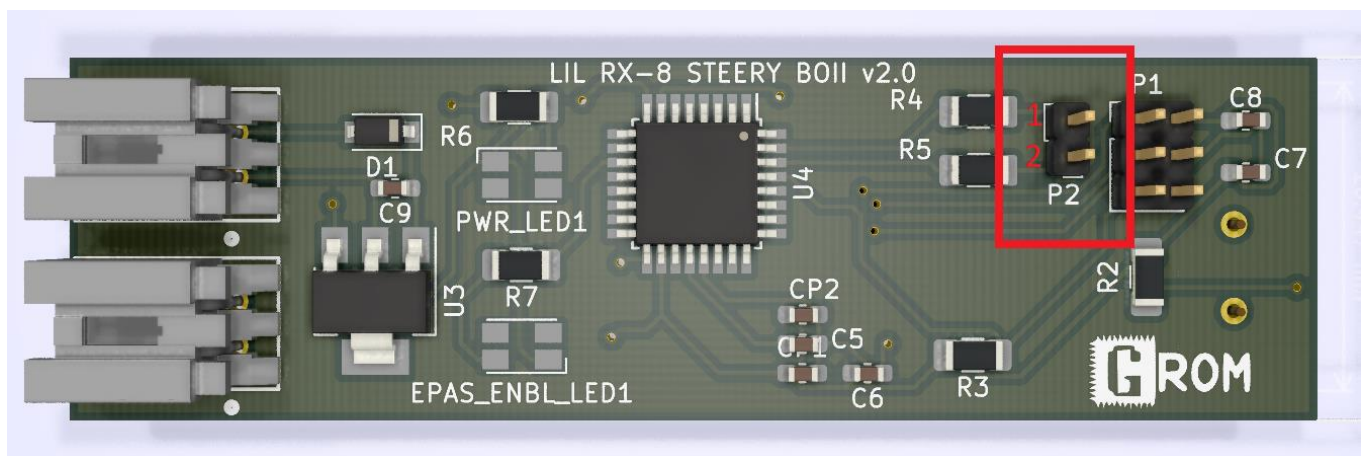
This interface maps a 0-100% boost amount to a "vehicle speed" as seen by Mazda EPS controller. The higher the boost, the lower the vehicle speeds the units sees and vice versa.

This interface is meant to be a set and forget type interface, and does not allow for dynamic adjust. This may change in the future.

Wiring and Connectors

To interface with the boost adjustment interface, the EPAS Enabler is assumed to be wired up according to the standard user manual. Failure to wire the EPAS Enabler according to the manual will give erratic results in both regular operation as well as interfacing with the boost adjust interface.

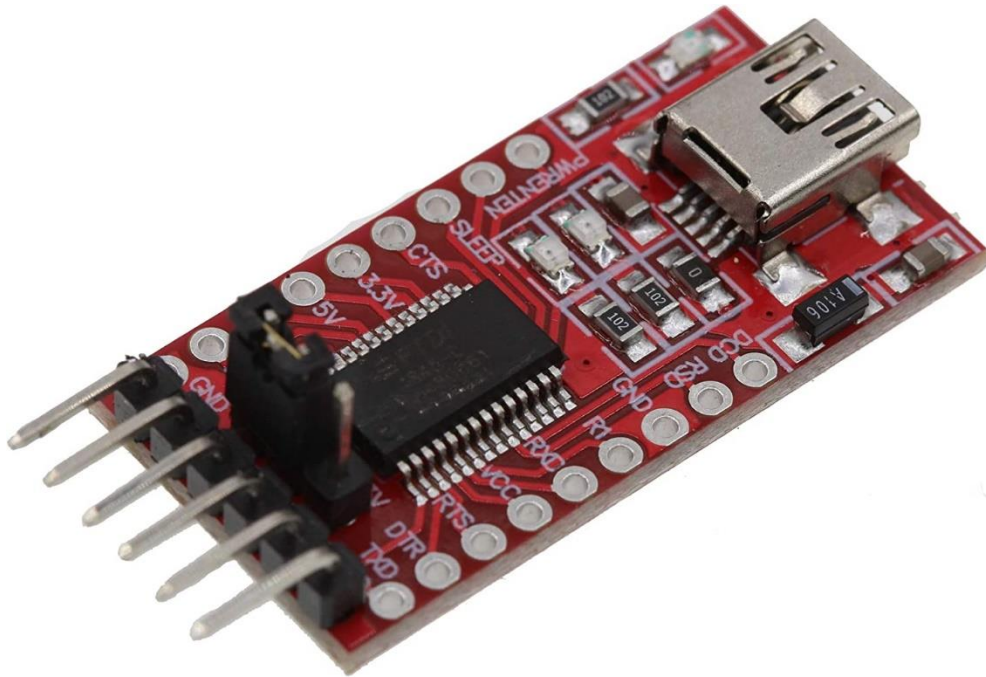
In addition to the standard wiring, there are two small serial pins that are populated as shown below:



Pin 1 – RX-8 EPAS Enabler Serial TX

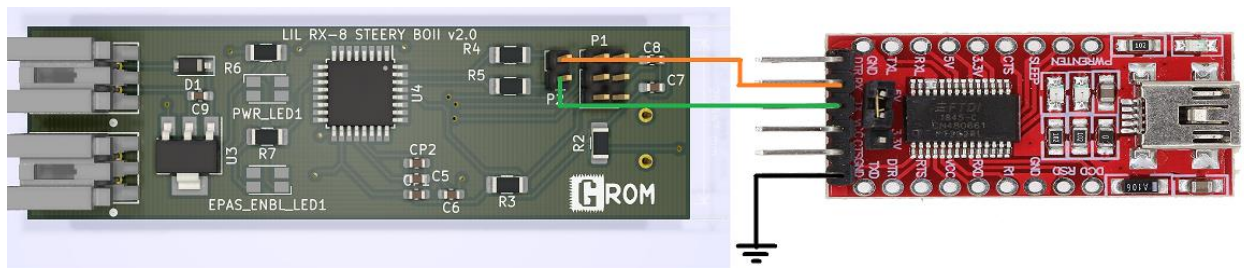
Pin 2 - RX-8 EPAS Enabler Serial RX

To hook up to this interface, you will need a USB to serial FTDI or similar device. I personally like to use these FT232RL USB to TTL devices. They can be found on Amazon for ~ \$7



No matter which serial to USB device you use, you just need to ensure the USB device's RX pin goes to the EPAS Enabler's TX pin, and that the USB device's TX pin goes to the EPAS Enabler's RX pin. This is commonly done backwards, so I am sure you will flip the wires around at least 3 times before getting them correct.

The ground of the USB device should also be connected to the same shared ground as the EPAS Enabler board as well.

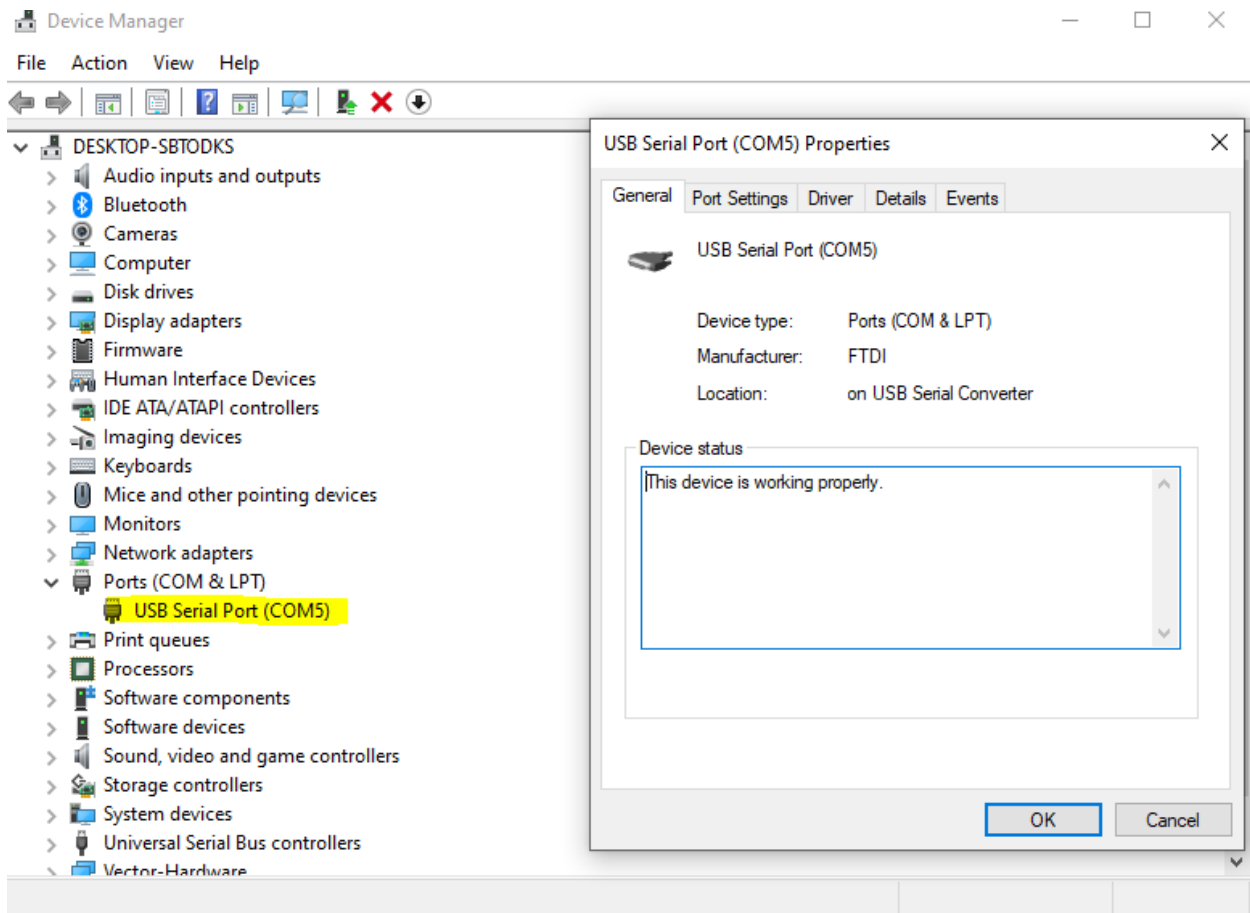


Interface Use

Before we plug anything in, we will need to download a terminal interface program. I like to use PuTTY, however any terminal program will work.

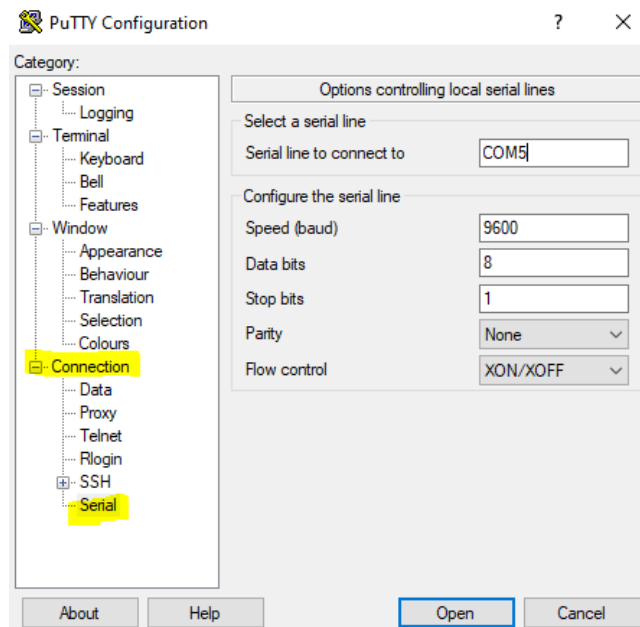
Download PuTTY here: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Once the device is wired up, and you've got a terminal program you need to fire up a Windows laptop or other computer that has a USB port. Plug the USB from the FTDI device in, and open up Device Manager. Inside of Device Manager you should look under "Ports (COM & LPT)" and see your device. Note down the COM port that is used. In this example, the COM port is COM5.

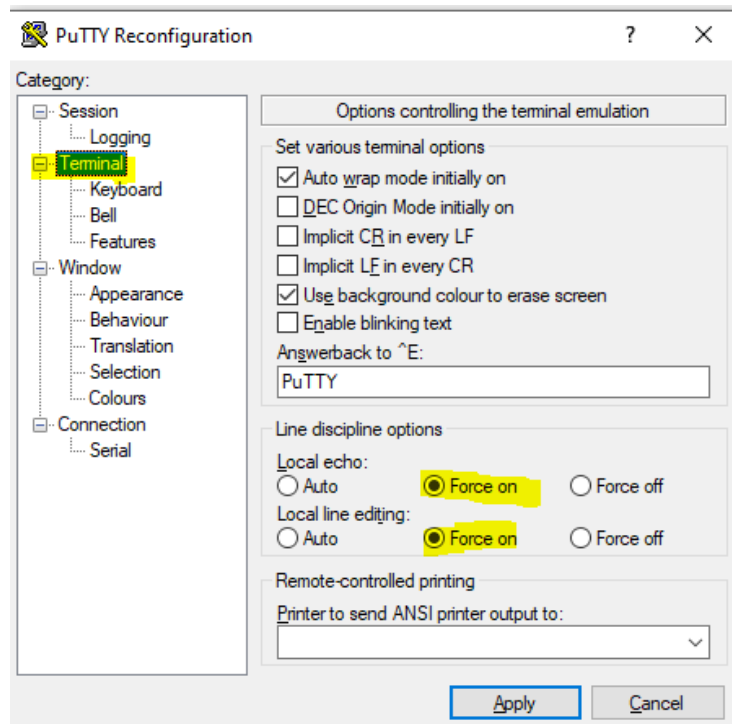


If you do not see a COM port, your device is not working.

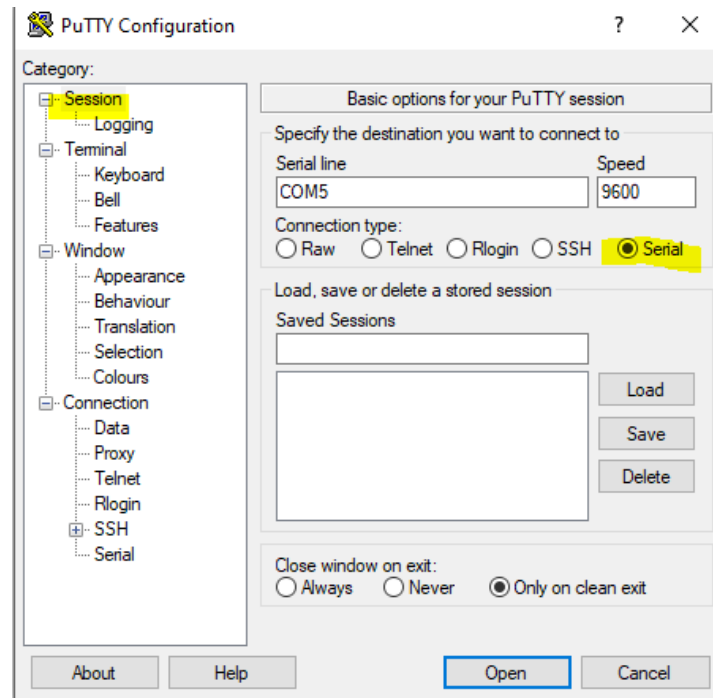
Next, open up PuTTY and go into the Connection settings and input your COM port number, and set the baud rate and other options as such:



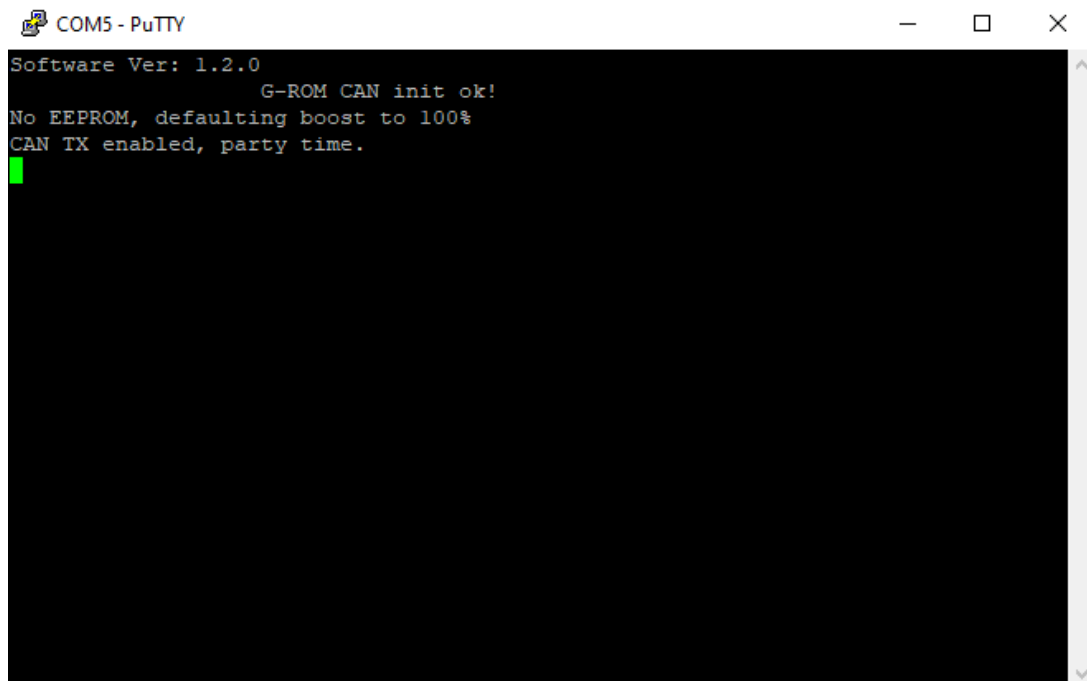
Next, click on “Terminal” and make sure the “Local echo” and “Local line editing” are set to “Force On”



Once the settings look okay, click on “Session” and select “Serial”



Click on “Open” and a blank window should pop up. Power up the EPAS Enabler, and if everything was successful you should see similar text:



If you do not see this text, something is not correct and needs to be fixed before the next step.

Now we have a command line interface open with the device, it is time to send it some commands. To change the boost levels, the syntax to tell the device to set the boost amount is as such:

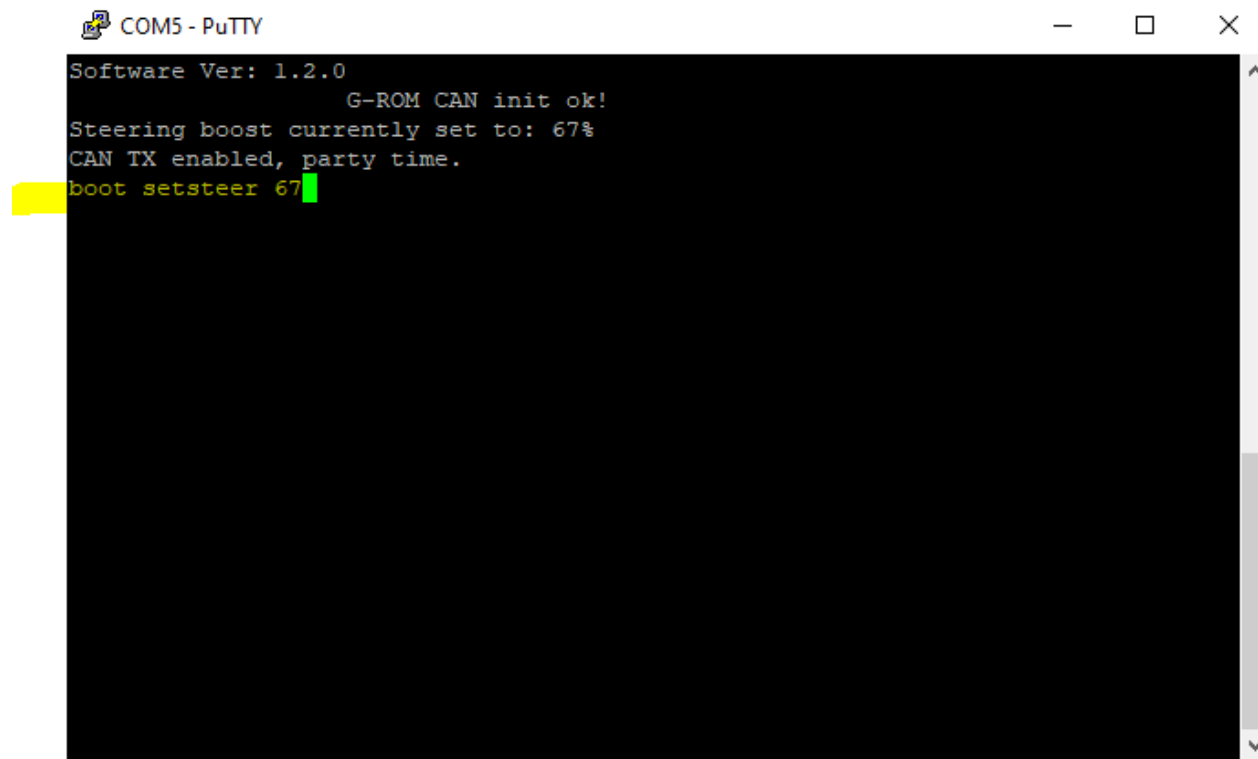
boot setsteer AMOUNT

The “boot” command, tells the device you want to talk to the interface.

The “setsteer” command tells the interface you want to set the steering boost amount

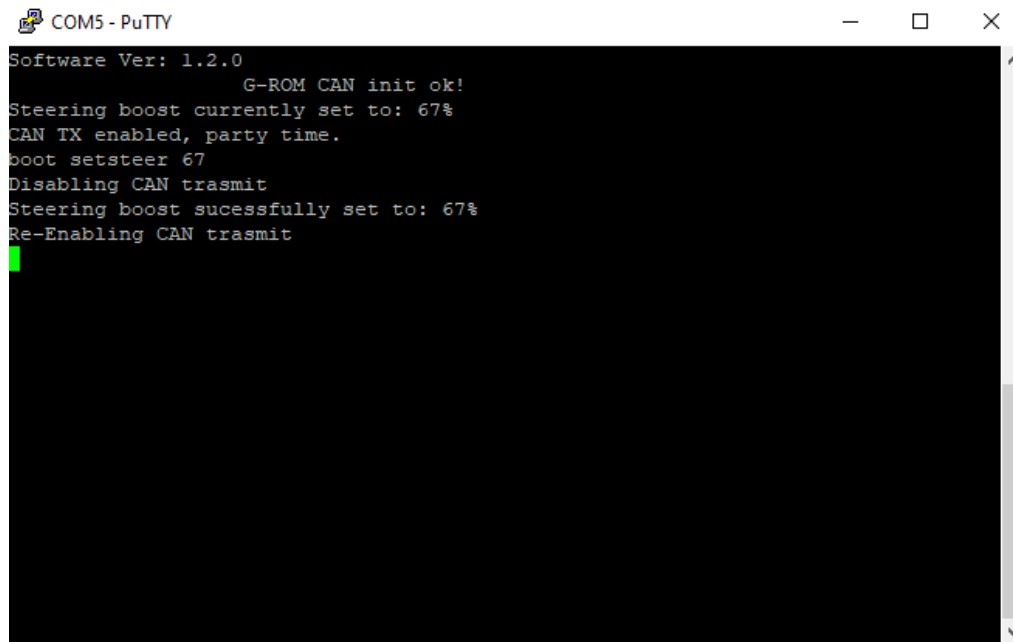
The “AMOUNT” is a number from 0-100 which sets the boost assist percentage.

For example, if you wanted to set the device to 67% boost, you would type this:



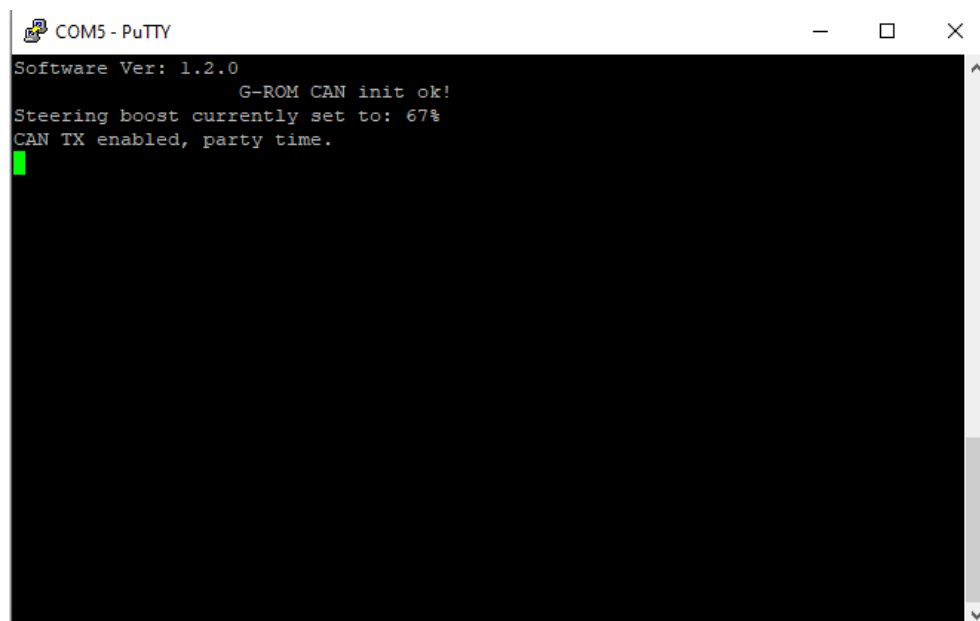
```
COM5 - PuTTY
Software Ver: 1.2.0
          G-ROM CAN init ok!
Steering boost currently set to: 67%
CAN TX enabled, party time.
boot setsteer 67
```

Once this is typed out, hit the “Enter” key. The device will respond back saying that is temporarily stopping CAN transmit, and will let you know if it was successful in setting the desired boost amount, and will automatically reenale the CAN transmit.



```
COM5 - PuTTY
Software Ver: 1.2.0
          G-ROM CAN init ok!
Steering boost currently set to: 67%
CAN TX enabled, party time.
boot setsteer 67
Disabling CAN trasmit
Steering boost sucessfully set to: 67%
Re-Enabling CAN trasmit
```

The device will immediately change the boost amount, no restart is required. If you would like to confirm what the boost is set to, power off and then power back on the device, and it will output the currently set boost value on startup.



```
COM5 - PuTTY
Software Ver: 1.2.0
          G-ROM CAN init ok!
Steering boost currently set to: 67%
CAN TX enabled, party time.
```

Disclaimers/Limitations

As mentioned, this is prototype software and has not been put through the paces and tested. What this means is that set boost levels may not actually reflect reality, which is to say 0% boost may not be 0% boost, and 100% boost may not be 100% boost. The amount of boost given from 0-100% may also not be linear, so use accordingly.

Once this becomes more tested, I will happily send out software releases to update the modules for more deterministic control.

This software is to be taken completely as is, and no guarantees are made. If you're reading this, I didn't even charge you extra for more features, so if you're unhappy with the adjustable boost you are entitled to 100% of your \$0 refund.