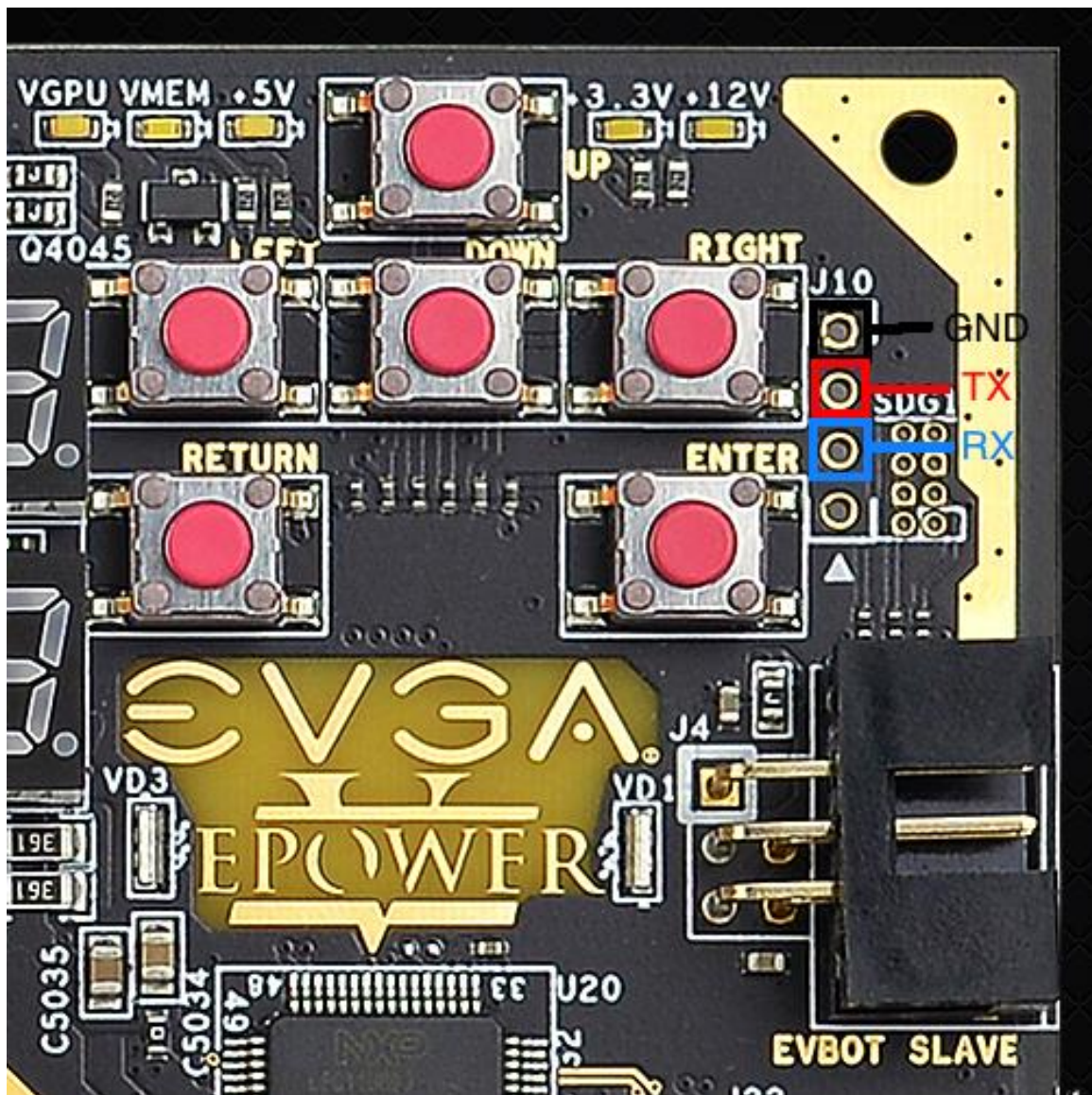


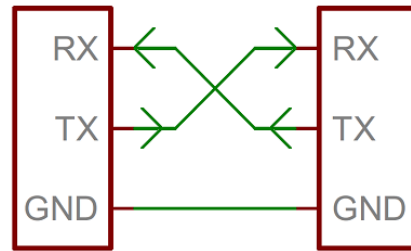
Software Voltage Control on EVGA EPOWER V

Disclaimer: This guide is provided as is, without any kind of liability or warranty, either from me or EVGA. I have no affiliation with EVGA.

Currently the only way to control the EPOWER V via software is by talking to the lpc11u68 using serial. The Pins for that are broken out to the top right of the PCB, however since there is no header soldered in you will have to either do that yourself or solder wires directly on the pads.



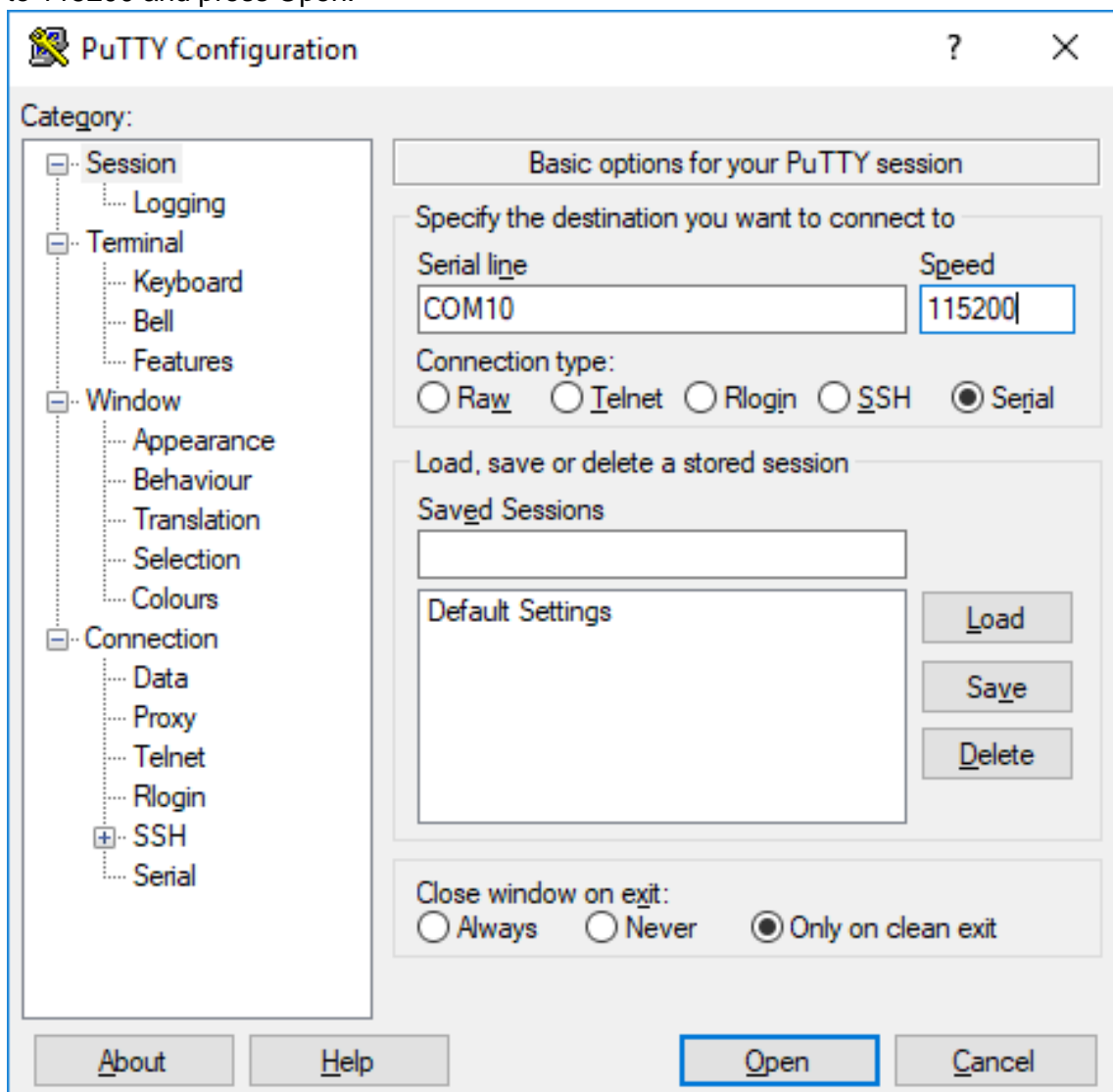
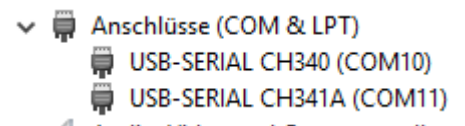
There are many ways you can communicate with the serial interface, the easiest is probably a USB to UART converter. Searching on eBay for “USB UART” you will find many different variants and clones. If you already have a ch341A based EEPROM programmer, you can also use that. The pins on the EPOWER are 5v tolerant so both 3.3v and 5v version will work. If you use a ch341A you will have to put it into serial mode, usually with a jumper on the PCB.



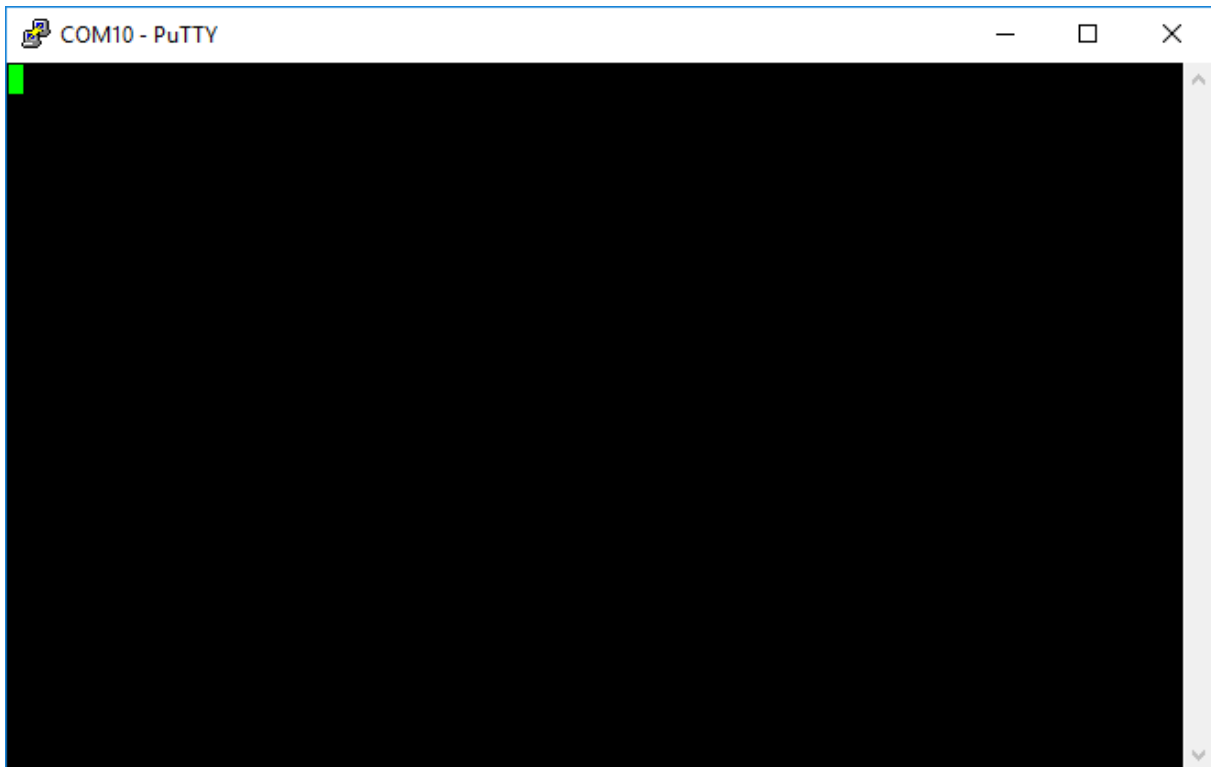
Connect the EPOWER to the converter as seen in the images and plug the converter into your PC. The following steps are going to differ slightly depending on if you are using Windows or Linux.

For Windows

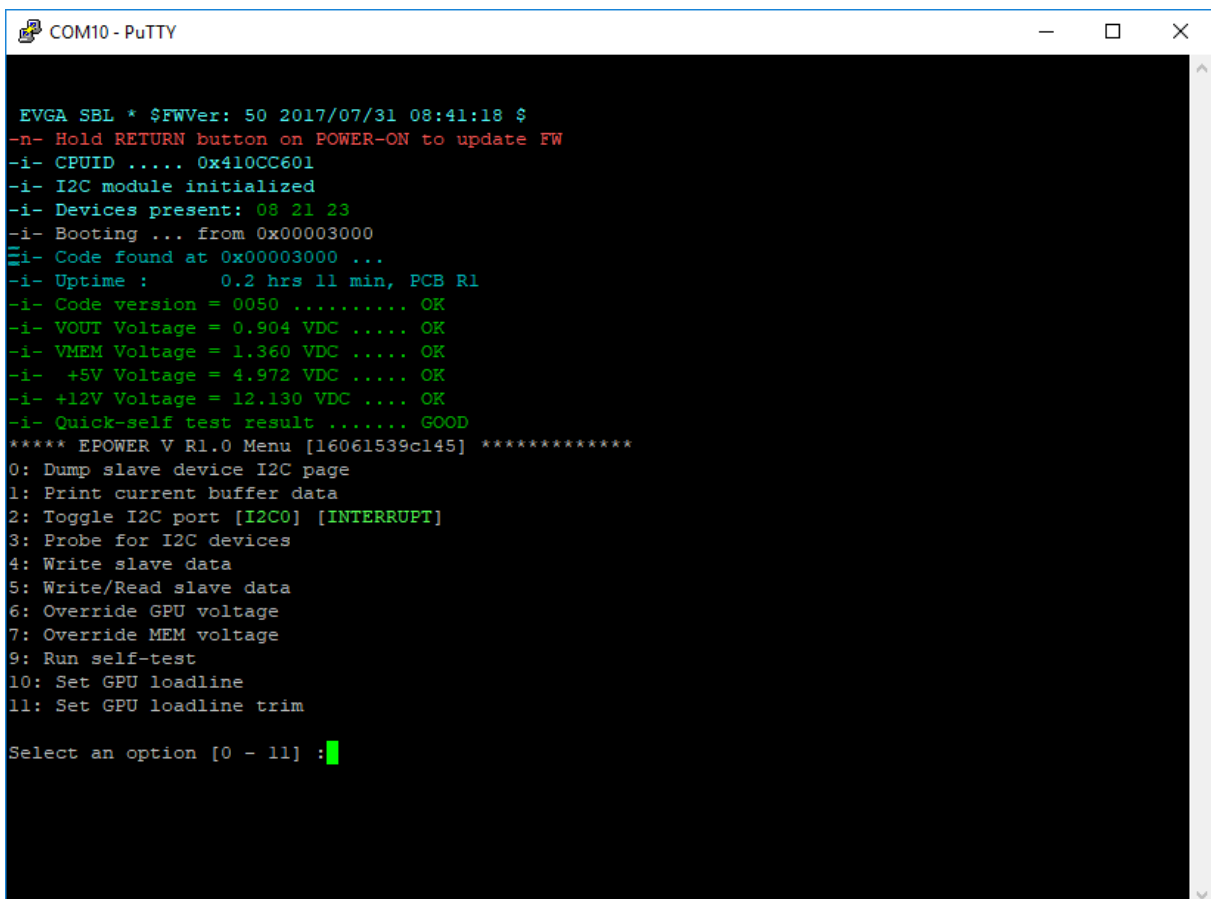
Open Device Manager and find your converter of choice under Ports. If you have multiple devices and are not sure which is the right one, just unplug it briefly and see which disappears. Next download and open Putty. First select Serial as your Connection type, then enter the COM number you see in device manager as Serial line, finally set Speed to 115200 and press Open.



An empty terminal window should open. If not troubleshoot your USB adapter and check your PuTTY settings.



Now when you power on the EPOWER you should see the following text. If not troubleshoot your connections from adapter to EPOWER. From here on there is no difference between Linux and Windows so skip ahead to “using the menu”.



For Linux

Verify that your adapter is detected by entering the `ls /dev/ttyUSB*` command. If you have multiple devices and are not sure which is the right one, just unplug it briefly and see which disappears. There are many different programs you can use for the next step. For this guide I will be using `screen`. Enter the command as seen in the picture, change the number if your adapter is not 0.

If you run into an error up to this point troubleshoot your USB adapter and make sure you are using the correct device and speed of 115200 8N1.

```
File Edit View Search Terminal Help
florian@zen:~$ ls /dev/ttyUSB*
/dev/ttyUSB0
florian@zen:~$ screen /dev/ttyUSB0 115200
```

Now when you power on the EPOWER you should see the following text. If not troubleshoot your connections from adapter to EPOWER.

```
File Edit View Search Terminal Help
EVGA SBL * $FWVer: 50 2017/07/31 08:41:18 $
-n- Hold RETURN button on POWER-ON to update FW
-i- CPUID ..... 0x410CC601
-i- I2C module initialized
-i- Devices present: 08 21 23
-i- Booting ... from 0x00003000
+i- Code found at 0x00003000 ...
-i- Uptime :      0.2 hrs 13 min, PCB R1
-i- Code version = 0050 ..... OK
-i- VOUT Voltage = 0.898 VDC ..... OK
-i- VMEM Voltage = 1.358 VDC ..... OK
-i- +5V Voltage = 4.972 VDC ..... OK
-i- +12V Voltage = 12.130 VDC .... OK
-i- Quick-self test result ..... GOOD
***** EPOWER V R1.0 Menu [16061539c145] *****
0: Dump slave device I2C page
1: Print current buffer data
2: Toggle I2C port [I2C0] [INTERRUPT]
3: Probe for I2C devices
4: Write slave data
5: Write/Read slave data
6: Override GPU voltage
7: Override MEM voltage
9: Run self-test
10: Set GPU loadline
11: Set GPU loadline trim
Select an option [0 - 11] :
```

Using the Menu

Options 0 to 5 allow you to play around with the i2c devices. You probably will not use these while overclocking.

Options 6 and 7 are the main event. They allow you to set both the primary and secondary voltage in the range of 800mv to 2000mv. If you enter a value outside of this range, you will return to the main menu. If you enter 0 the voltage will be reset to firmware default.

There are 2 issues/bugs to be aware of.

First the readout "VMEM Voltage readout" that you get when you first enter option 7 is half the actual value. The readout after you have entered your desired voltage is correct.

Second if you select 6 or 7 and then return to the menu without setting a value first it will also apply default voltages.

```
Select an option [0 - 11] :7

VMEM Voltage readout = 672 mV
VMEM OCP Setting     = 124.00 A
DVID CTRL            = 84
Enter voltage, in millivolts (0 = default, or <800,>2300 to exit) :1600
  VMEM: 1.611491 V
Enter voltage, in millivolts (0 = default, or <800,>2300 to exit) :700
**** EPOWER V R1.0 Menu [16061539c145] ****
0: Dump slave device I2C page
1: Print current buffer data
2: Toggle I2C port [I2C0] [INTERRUPT]
3: Probe for I2C devices
4: Write slave data
5: Write/Read slave data
6: Override GPU voltage
7: Override MEM voltage
9: Run self-test
10: Set GPU loadline
11: Set GPU loadline trim

Select an option [0 - 11] :7

VMEM Voltage readout = 797 mV
VMEM OCP Setting     = 124.00 A
DVID CTRL            = 86
Enter voltage, in millivolts (0 = default, or <800,>2300 to exit) :1400
  VMEM: 1.413103 V
Enter voltage, in millivolts (0 = default, or <800,>2300 to exit) :█
```

Option 8 is not listed in the menu and just prints "This card is OKAY to RMA :)".

Option 9 runs some sort of self-test.

I do not know if 8 or 9 are supposed to be here or if they do anything. I assume they are left over from the 980 ti KP that this menu is based on.

```
Select an option [0 - 11] :8
This card is OKAY to RMA :)
***** EPOWER V R1.0 Menu [16061539c145] *****
0: Dump slave device I2C page
1: Print current buffer data
2: Toggle I2C port [I2C0] [INTERRUPT]
3: Probe for I2C devices
4: Write slave data
5: Write/Read slave data
6: Override GPU voltage
7: Override MEM voltage
9: Run self-test
10: Set GPU loadline
11: Set GPU loadline trim

Select an option [0 - 11] :9
-i- Code version = 0050 ..... OK
-i- VOUT Voltage = 0.905 VDC ..... OK
-i- VMEM Voltage = 1.354 VDC ..... OK
-i- +5V Voltage = 4.974 VDC ..... OK
-i- +12V Voltage = 12.126 VDC .... OK
-i- Quick-self test result ..... FAILED
```

```
Select an option [0 - 11] :10
L1 Loadline Active = 1
L1 RES = 0.30 mOhm
L1 TRIM = -10 %
Enter LL (0 = OFF, 1 = ON):0
LL set to 0
***** EPOWER V R1.0 Menu [16061539c145] *****
0: Dump slave device I2C page
1: Print current buffer data
2: Toggle I2C port [I2C0] [INTERRUPT]
3: Probe for I2C devices
4: Write slave data
5: Write/Read slave data
6: Override GPU voltage
7: Override MEM voltage
9: Run self-test
10: Set GPU loadline
11: Set GPU loadline trim

Select an option [0 - 11] :11
L1 Loadline Active = 0
L1 RES = 0.30 mOhm
L1 TRIM = -10 %
Enter LL (0 = 0%, 1 = 5%, 7 = 30%, 8 = -45%, 15 = -5%):0
LL trim set: 0
```

Option 10 and 11 allow you to set the load line of the primary VRM. 10 is just disabling and enabling load line. 11 allows you to control how much load line you want to have. Note that I have not tested if either of these applies.

Thank you for reading! If you have any questions or suggestions, you can find me as 1whiteshark1 on hwbot or as whiteshark in the /r/overclocking discord. Happy benching!



Disclaimer: This guide is provided as is, without any kind of liability or warranty, either from me or EVGA. I have no affiliation with EVGA.