**Rev B Bringup Documentation**

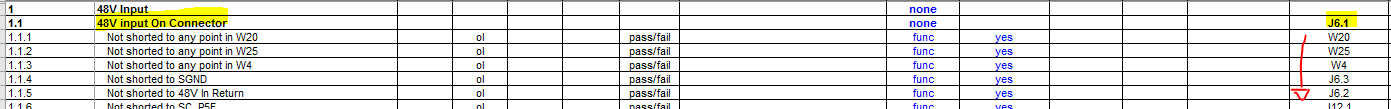
The purpose of this document is to outline the steps I would take as soon as I get the 661-349-03 Rev B Gemini board to make sure it is working and bring it up to RFUBO.

It will also include steps for DQV

FPGA bitfile location: [\\ter.teradyne.com\hwnet\design\_release\hydra\REL\_06\_01\_2018\_11\_14\bitfiles](file:///\\ter.teradyne.com\hwnet\design_release\hydra\REL_06_01_2018_11_14\bitfiles)

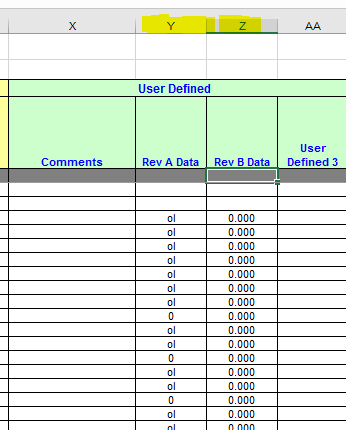
* The first step is to do a regular visual inspection of the board. Then a visual inspection of the items changed in Rev B. Make sure that:
  1. LON connector J14 has the right orientation (matches silk outline)
  2. Anchoring point for both rider cards is just a keepout and not a hole (In Rev A this point was a hole)
  3. Humidity sensor is installed in the header (J27) and PIN 1 is in the right position
  4. Temperature sensor U23 has right orientation (matches silk outline)
  5. Connector J13 can be plugged in without interference (This problem should have been caught during HLA assembly if there was a problem)
  6. VR3 silkscreen is completely visible
* Next use the Rev B DQV spreadsheet located at [\\trail\public\Dragon\SMC\Gemini\DQV\DQV\_Gemini\_RevB.xls](file:///\\trail\public\Dragon\SMC\Gemini\DQV\DQV_Gemini_RevB.xls) and go to the “Ohm Check” tab. There I have documented the measurements to do on the board to verify that the power planes are not shorted or looking weird as a bare board. Do not plug anything to the board when doing this step.

You can take a look at the “Measurement Point” column to guide you on which pin to probe. For example on the first line 1.1 48V Input On Connector. The test point is J6.1 so I grab one probe in continuity mode, put the first end on J6.1 (or any point for that net) and the other end will go to w20, w25, w4, etc.



Here I have the code “ol” for overload, which is open. Else the resistance measurement given.

On this tab I have set up all the way to the right. On Columns Y and Z a way to compare the new Rev B to old Rev A measurements



This tab is not part of the formal DQV stats. So it is not included in the Summary tab.

* After you are happy with the measurements, the next step is a smoke test. Assuming you have the HLA already built, plug in one end of the Cosel to connector J6, and the other end to the wall. If there is no smoke, some of the LEDs at the bottom side of the board will turn on. Since they are difficult to see when the board is on the tray, I like to move the jumper of J23 from 1-2 to 2-3. This will light up the LED on the top. It is easier to debug power problems this way.

Now you are ready to measure the power section of the RFUBO DQV tab. This section goes up until 4

The items here have are self-explanatory in the spreadsheet. It also contains instructions on which points to short the relay to enable the bricks without the rider cards. Here also all the way to the right on columns Y and Z you can compare the previous with the current values

* Once the power has been verified and everything looks normal, you have to program the FPGA. You need to have Quartus installed. The version I used if Quartus Prime lite 16.1 but if this is done in the future I imagine any version above should work. I put an installer on the following folder, but in theory you can download a copy from the Altera website

\\trail\public\munozdav\Quartus

Install Quartus in the default directory.

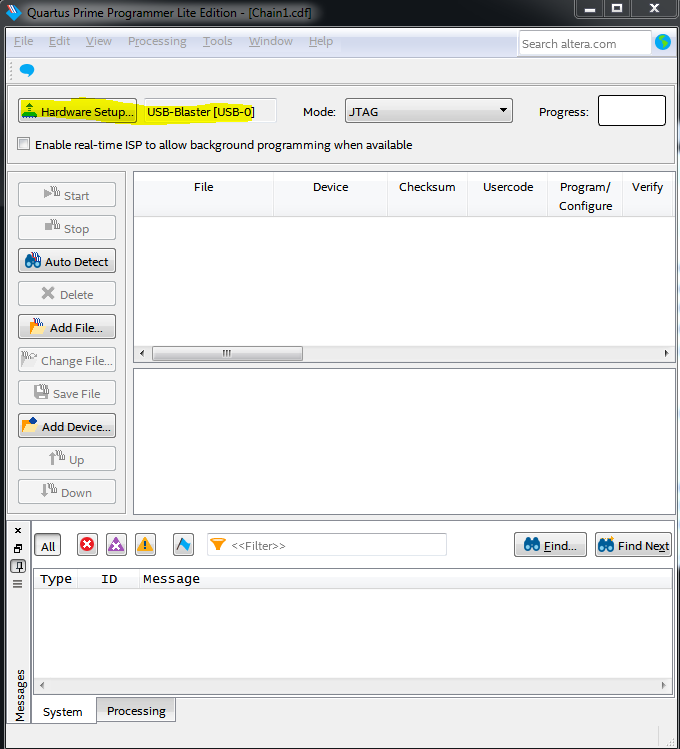
Now you can grab the .pof file from Adam and put it in an easy to access local folder (Note that Adam can also generate .sof files. You want to ask specifically for the .pof file since this one is the only one that can program to the internal flash memory for persistence)

Plug in to your computer the Altera USB blaster (I left it by the lab cabinet closest to the door, in al altera box labeled “Oregon Design Center” This is how it looks like

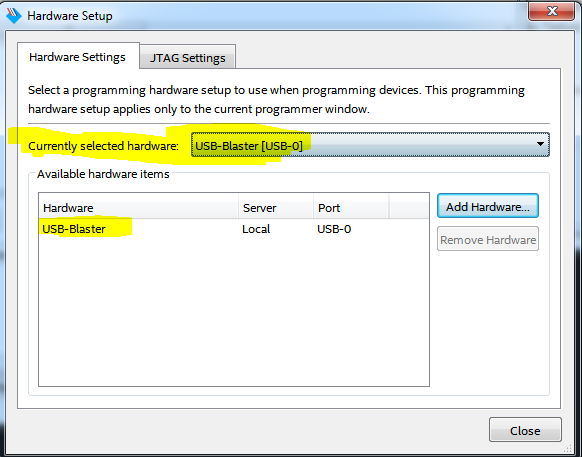


If it is the first time you plug it in to your computer, it will take some time to install its drivers.

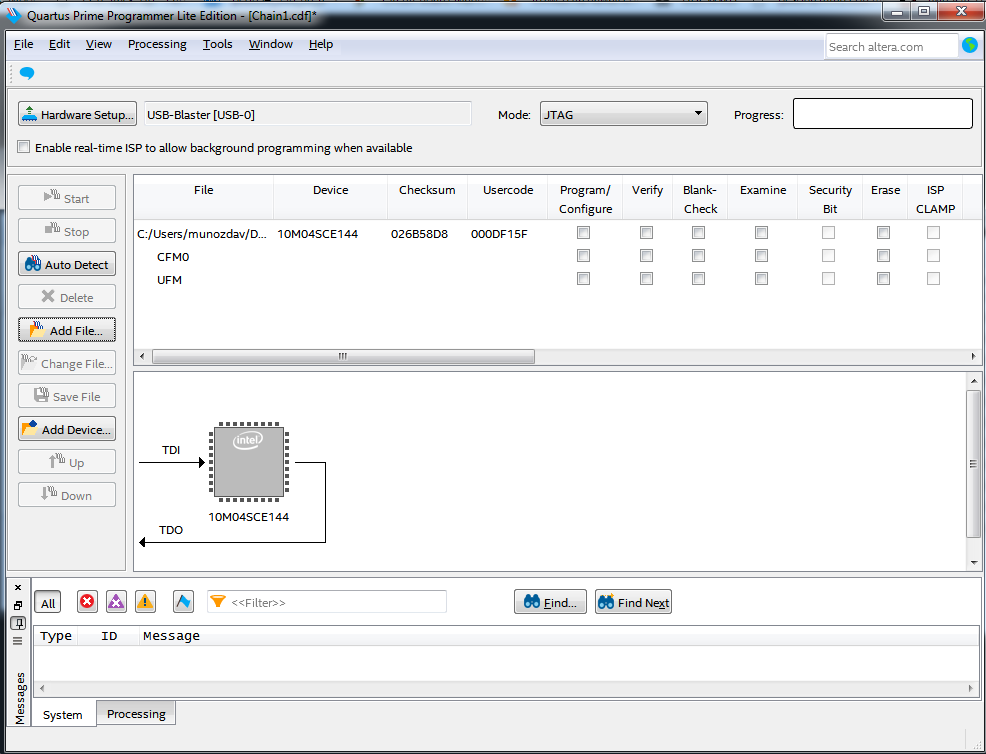
After that is done, open the Quartus programmer located at C:\intelFPGA\_lite\16.1\quartus\bin64\quartus\_pgmw.exe



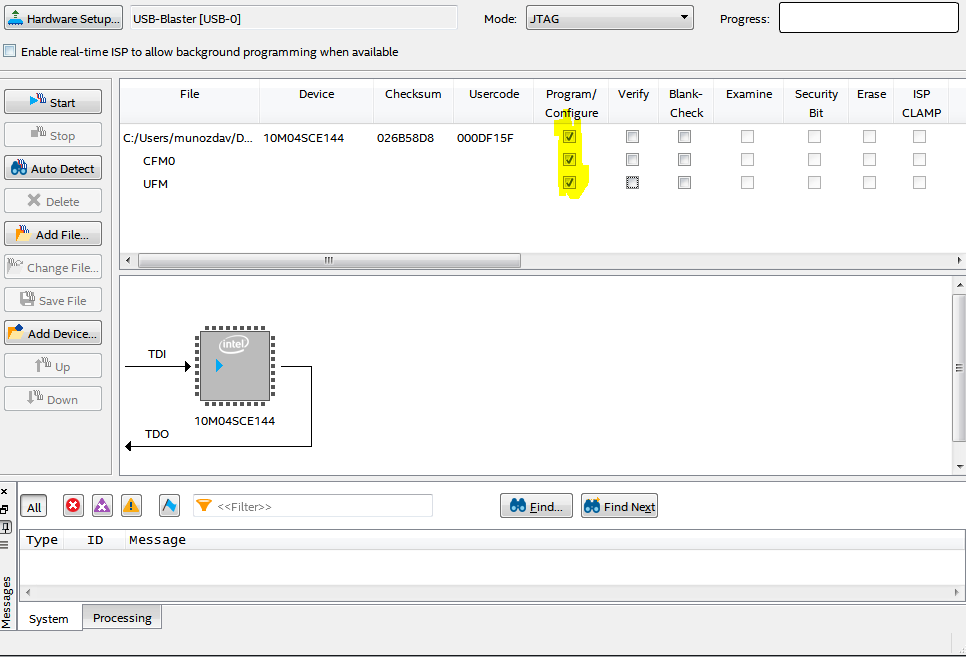
Make sure the Hardware Setup button says “USB-Blaster” next to it. If it doesn’t you can click on the Hardware setup button and select it from the dropdown menu of “Current selected hardware. If the USB Blaster does not show up in the Available hardware items the computer did not recognize the USB Blaster. Maybe the drivers did not install correctly or the USB port is not working correctly. You can also try exiting the Quartus programmer and try again when opening it



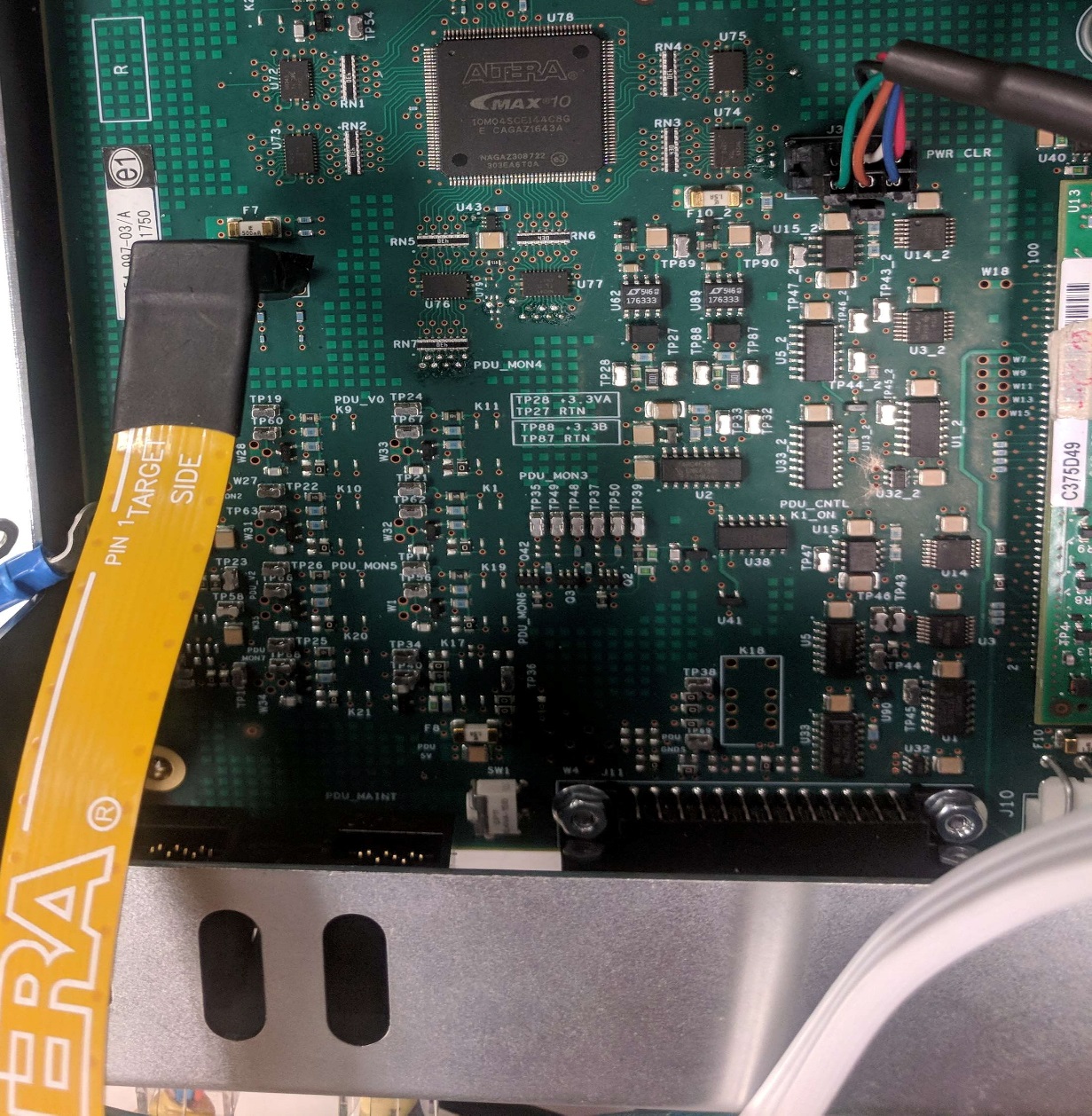
Now click the “Add File” button and select the .pof file from earlier. Now the screen should look like this



At this point, I am unsure which checkbox is used to program the internal flash of the FPGA (I think any of them will work) but for good measure, I select both CFM0 and UFM to program



Turn on the 48V power to the board

Place the usb blaster as shown in the pictures

The connection is not perfect since the pins are smaller than the holes. So to get a good contact you have to move the connector against the side and keep it with your hand. It is not too complicated to get it working but it might take a few tries.

Not you can press the “start” button to program it. In the output box in yellow you should start seeing the status of the programming. It should take at most a minute to do it and there should not be any errors. Check for errors highlighted in red. The most likely cause of errors is a flake connection, so try wiggling the connector to the other side and try again.

* Now you can start putting everything in. You should start by only plugging only the main rider card. Turn on the power and make sure no smoke comes out (you should start with a “TIB” rider card. Not necessary to pre-program the rider card at this point) (do not worry if you do not see any rider card LED activity at this point, although there might be)

Turn off the power. Now plug in the main Andromeda board (at this point you should make sure J23 jumper is in the default 1-2 position) turn on power and make sure no smoke comes out. (Also make sure you got the latest SD Card from Jim Hansen. Check the appendix on how to program an SD card)

If no smoke comes out, you should start seeing rider card LED activity along with Andromeda LED activity. The usual sequence of the Andromeda is: solid yellow LED -> Green blinking LED. The transition of LED sometimes takes a while, especially if there is no network cable connected to the Andromeda.

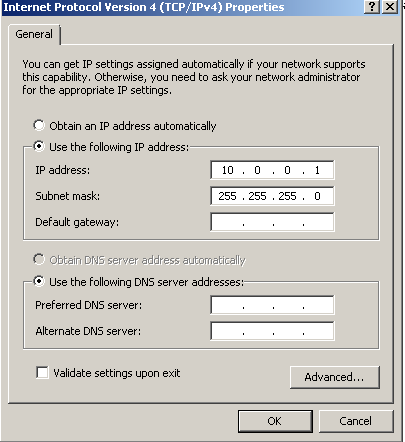
* At this point you should plug in a network switch to get the SMC network going (any switch should work. I have been working with the one on TB6-4, which I just plug in straight to the power strip. It takes a few minutes to start up this switch.)

Connect the main Andromeda network cable to the switch

Connect the host computer SMC port to the switch

Just as a reminder, the SMC port in the computer has to be configured like this on Windows:

Go to “network Connections” on the control panel. Double Click “Teradyne\_SMC”. Select “internet Protocol Version 4(TCP/IPv4)” and then click “Properties”



Now open a command prompt windows and type “telnet 10.0.0.2” To connect to the Andromeda.

At this point there should be a successful connection. Try typing “nodes” and check the output (should see node 126). If telnet cannot connect to the Andromeda something went wrong, some configuration is wrong, or something is not plugged in right. Look at the end of this document for a description of common problems and solutions.

* Now turn off the power. Plug in the secondary rider card. Turn on the power and make sure no smoke comes out (Due to the Rev B redesign, you should start with a “TIB” rider card. Not necessary to pre-program the rider card at this point) (do not worry if you do not see any rider card LED activity at this point, although there might be)

Turn off the power. Now plug in the secondary Andromeda board. Turn on power and make sure no smoke comes out.

If no smoke comes out, you should start seeing rider card LED activity along with Andromeda LED activity. The usual sequence of the Andromeda is: solid yellow LED -> Green blinking LED. The transition of LED sometimes takes a while, especially if there is no network cable connected to the Andromeda. Check the end of the document for troubleshooting if there is no LED activity.

Connect the secondary Andromeda network cable to the switch

Now open a command prompt windows and type “telnet 10.0.0.4” To connect to the secondary Andromeda.

At this point there should be a successful connection. Try typing “nodes” and check the output (there should be node 126). If telnet cannot connect to the Andromeda something went wrong, some configuration is wrong, or something is not plugged in right. Check the end of the document for troubleshooting

* At this point you are able to complete the rest of the RFUBO tab.
* After completing RFUBO you are ready to re-program the rider cards to the proper limit table and firmware. Please use the instructions located here: \\trail\public\munozdav\Telnet RC

The latest Gemini firmware and limit tables are located here: (Brian White is/was developing a unified firmware, check if he finished it and use that one for both)

\\trail\public\Dragon\SMC\Limit\_Tables\SMC\_Shelf\_Slot\_126\Gemini scn Tables

After updating reboot the SMC (it can be done by power cycling or pushing the SW1 button) Check the appendix for troubleshooting if there is no rider card activity after this update

* If everything goes well, at this point you are able to start with the Validation tab of the DQV. Here is how you turn on each testhead.

How to turn on the testheads depends if you have a remote PSU and CDU. First, I am going to outline **how to turn on the testheads without any PSU and/or CDU**.

Telnet to both andromedas (10.0.0.2 and 10.0.0.4)

Send these commands to both andromedas

* Ovron 119
* Ovronmon 126,31

Now to turn Testhead 1, you have to short to ground the point U32.1

To turn Testhead 2, you have to short to ground the point U32\_2.1

Sometimes for some reason, this does not work. Usually I have to send the command sndmsg 126,30

And if that doesn’t work I try sndmsg 126,25.

But after that it should not be necessary to do that again, unless you power cycle.

To turn off the test heads, you have to short to ground points U32.3 and/or U32\_2.3

**How to turn on the testheads with PSU and CDU**.

Telnet to both andromedas (10.0.0.2 and 10.0.0.4)

Send this commands to both andromedas

* Ovronmon 126,31

Now to turn Testhead 1, you have to short to ground the point U32.1

To turn Testhead 2, you have to short to ground the point U32\_2.1

That is it to get started! Get going on the Validation tab of the DQV

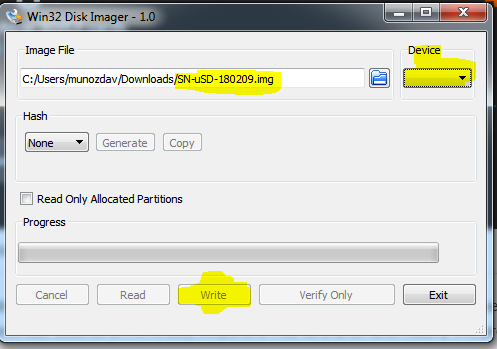
**Appendix:**

The master folder for Gemini related documents is [\\trail\public\Dragon\SMC\Gemini](file:///\\trail\public\Dragon\SMC\Gemini)

**To program an SD card for Gemini:**

You will need a micro sd, micro sd adapter to regular SD size, the .img file from Jim Hansen. The software win32-disk-imager (you can get the latest installer at <https://sourceforge.net/projects/win32diskimager/files/latest/download>, I also put a copy on this folder \\trail\public\munozdav\Gemini Software)

Open Win32 Disk Imager and select the Image file from Jim, select the Device Letter from the SD Card, and then click on “write”



**If there is no LED activity on the main Rider card**: The only problem we had on Rev A for the rider card was the communication from the internal ADUC to the Gemini FPGA. In gemini this logic path is controlled by the AD signals. The first step is to verify the rider card is not actually broken. Try plugging it in on a TIB to make sure the system can actually see the rider card.

The next thing to try is to verify the FPGA is actually programmed. For this make sure the Rider Card clock is reaching the FPGA (U78.29) The easiest way to check if making sure that the ADUC\_RD signal going to the FPGA (U78.62) is reaching it. And that the signal coming out AD\_READ\_CONTROL (U76.2) is producing an output.

Once you know the FPGA is programmed. Check that there are not strange signal irregularities in the lower bits of AD. I suggest looking at the signal AD01 (RN7.2)

The next thing to check if it is a problem with the AD data received. Connect the rider card from the TIB and program the Gemini Rider Card firmware that ignores the FPGA (look at the readme.txt at \\trail\public\Dragon\SMC\Limit\_Tables\SMC\_Shelf\_Slot\_126\Gemini scn Tables) and then try the Rider Card on Gemini. If it lights up and there is activity it means there is a problem with the AD data.

Since the rider card has activity with the firmware that ignores the FPGA. You can reprogram it from Gemini to the correct firmware to try the new FPGA code

If you program the rider card with the firmware that ignores the FPGA, and it still doesn’t work, it could mean that the path of the AD signals to the FPGA is the problem. In Rev A we had a problem with the translators. We used a part that had an active bus hold circuitry that kept the signal low when it needed to go high. Confusing the rider card. This should not happen in Rev B but it is a good piece of knowledge.

**If there is no LED activity on the secondary Rider card**: Here there could be 2 potential different problem. The Rider card LED does not turn on. Or it turn on but there is no activity.

If the LED does not turn on, it could mean that the Rider Card is broken. Or that the ADUC path in Gemini is somehow broken.

The first step is to verify the rider card is not actually broken. Try plugging it in on a TIB to make sure the system can actually see the rider card.

The next thing to try is to verify the FPGA is actually programmed. (if you got the main rider card working, this step is already done)

Make sure that the signal coming out of the FPGA that controls the AD\_READ\_CONTROL\_TH2 (U94.2) is producing an output.

Check that there are not strange signal irregularities in the lower bits of AD and comparing them to the AD bits of the main rider card. I suggest looking at the signal AD01\_TH2 (RN9.2)

At this point if you verified all of the above, the Rider Card should have at least activity LEDs. If it doesn’t, try disabling the Voltage Translators by stuffing a 0 ohm resistor to R470, R471, and R467. Also program the Rider Card with the experimental firmware that ignores the FPGA. This should get you to LED activity.

If all of the points of above look normal. Try checking with Adam Moriarty to see if his new code is not causing any problems.

But you can always compare how the signals should look like with the Main rider card signals equivalent signals.

**How to use serial to USB connector to communicate with the Andromeda in case network is not working:**

Use the cable located in the cabinet next to the lab door



Connect the usb to the computer you are going to use, and the other end to the Andromeda 6 pin header taking care of placing pin 1 in the correct location.

If it is the first time you plug it in, windows automatically will install the drivers. The device will appear as a COM device.

Set up a serial communication channel with the following settings:

Speed, Baud rate: 115200

Data Bits: 8

Stop bits: 1

Parity: none

Flow control: none

We usually use MobaTerm for this. Make sure you selected the right COM port when you set this up

**Major changes on Rev B:**

* Extra relay on AV400 12V
* Extra RC circuit on serial data to muxes
* Change of pin location on various signals including: ADUC, and PDU monitors
* Added extra direction control signal to lower AD bits voltage translator
* Added header for humidity sensor
* Added extra second rider card logic