February 18, 2020

Robert Johnson

pCT Phase-II Scanner Operation

# Installation and Cabling Setup

In the following, use cable ties to secure the various signal cables, high voltage cables, USB cables, Ethernet cable, etc. to the scanner structure, to avoid damage if somebody trips over a cable.

1. Set up the data acquisition computer, its monitor, its mouse, and its keyboard in the radiation-safe area. Two cables will run from it to the scanner. Probably they will have to run under the door.
2. Using the handles attached to the breadboard, two people must lift the scanner and set it either on top of the special jack cart (at LLUMC) or on top of the robotic positioning arm (e.g. at CPC). Clamp it securely in place if on the jack cart.
3. Install the energy detector from the rear onto its table under the electronics of the scanner, taking care to clear the various cables. When handling the detector take care to avoid touching the entrance window. The lemo plugs must be on the left (when looking toward the beamline) and pointing downward. Scoot the enclosure all the way forward and to the left, such that the ridge around the enclosure pushes up against the table. Check the left-right position to verify that the energy detector window matches the tracker window.
4. Plug in the 5 energy detector signal cables according to the channel labels on the detector and on the cables, and then plug in the high-voltage cable.
5. Position the scanner properly relative to the beam line, making sure that the height is correct to put the beam into the center of the aperture. Lock it into place if possible. If using a scattering foil, as at LLUMC, then measure and log the distance from the foil to the front of the tracker.
6. Plug in the long white AC-power extension cord and position its outlets close to the scanner. Plug the black extension cord into the white one or into another outlet.
7. Plug in the AC-to-12V power converter for the two tracker fans. It is identical to the converter for the event builder. Make sure that it is plugged into the same power strip as the tracker power, so that the fans will always be on when the tracker is on.
8. Plug the event builder 12V AC power adapter into the white extension cord and then into the event builder ML605 board (a white connector on the top of the board). Try to avoid bending the board when plugging it in. Making this connection first (and removing it last) is helpful in grounding all of the electronics and thus reducing susceptibility to electrostatic discharge.
9. Position the following auxiliary boxes close to the scanner and attach an AC power cord to each:
   1. Two black BKPRECISION switching power supplies.
   2. The Sorensen XEL supply for the tracker bias.
   3. The energy detector ISEG high-voltage supply.
   4. The stage controller.
10. Connect the energy detector high voltage cable to the ISEG power supply (making sure that its high voltage is turned off first). To avoid damage the two cable ends with high-voltage connectors must connect to the correct channels! The −365 volts goes to the left channel (when facing the front of the supply) while the −800 volts goes to the right channel. Don’t do this if you are unsure, as applying the −800 volts to the wrong cable will surely destroy all of the transistors and zener diodes on the PMT bias board inside of the energy detector. Be careful with this cable, also, as it is cobbled together from a mixture of high-voltage and non-high-voltage cables and may be fragile.
11. Plug the BKPRECISION power cords into the black extension cord. This is important to ensure that the tracker power cannot be turned on without turning on the tracker fans.
12. Plug the other boxes into the white extension cord.
13. Connect the cable from the stage controller to the stage.
14. Connect the tracker bias cable from the Sorensen XEL supply to the mezzanine board. On the supply end the cable has banana plugs, red to the positive high voltage and black to negative terminal.[[1]](#footnote-1) The plug on the mezzanine board is clearly labeled and marked with red tape. It is crucial not to plug this into the tracker or energy detector power plugs. The 80V bias would surely destroy the electronics! When you turn the Sorensen power on, *make sure to adjust it manually to no more than about 80V before the output is turned on*.
15. Connect the energy detector power cable (thinner 4-wire grey cable attached to the energy-detector BKPRECISION supply) to the crowbar box that is in turn plugged into the mezzanine board.
16. Connect the tracker power cable (thicker 4-wire grey cable attached to the tracker BKPRECISION supply) directly into the mezzanine board. (There was a crowbar box meant to be for the tracker, but in practice it does not work well enough for operation in beam tests. The 18-amp fuse will not blow when the crowbar trips, because the supply cannot source more than 22 amps, which apparently is not enough to blow the fuse. At the same time, the 16 or 17 amps drawn by the tracker heats up the fuse a lot and produces too much voltage drop across it.)
17. Set up the white USB hub close to the scanner and plug its power converter into the white extension cord. Plug its white input USB cable into the end of the 50-foot active USB cable and into the hub. Cable tie all of this into place to secure it.
18. Run a USB cable from the hub to the event builder. The event-builder end is a mini-connector. There are two jacks, one for programming and the other for the UART connection. Here we want the UART connection. There is a small label on the board, but it can also be recognized as the USB jack that is closest to the mezzanine interface board.
19. Run USB cables from the hub to the two BKPRECISION supplies, the Sorensen XEL supply, the ISEG high-voltage supply, and the stage controller.
20. Run a long Cat-5 Ethernet cable from the little Ethernet board mounted onto the event-builder mezzanine board to the data-acquisition computer. On the back of the computer it should plug into the lower Ethernet adapter (the upper one is for network connections). Looking at the back of the computer, plug it into the right-hand socket (it won’t work in the other).
21. Turn on the power strip of the white extension cable.
22. Make sure that the data acquisition computer has a connection to the Internet either via the top Ethernet board or via a wireless adapter plugged into USB. Otherwise the operating system will keep trying to find the Internet on the Ethernet adapter being used to acquire data, and it will very likely interfere with the data acquisition and/or cause the Ethernet link to the scanner to hang.

(I’m assuming that no beam-phase-lock will be used for the energy detector digitizer clocks. Otherwise the beam RF coax cable would have to be plugged into the lemo connector of the mezzanine board and a USB cable from a PC would have to be plugged into the same board in order to program the chip that does the phase lock and generates the digitizer clock.)

# System Startup

The password for the data acquisition computer: daqdaqg0

Turn on the system power as follows:

1. The power strip into which the tracker ventilation fans and tracker power supplies are plugged is turned on first. You should then hear the fans. Do not proceed to turn on the tracker power without the fans running. Check that you can feel air exiting the top of the tracker enclosures.
2. Turn on the event builder (Xilinx ML605 board). The switch is partially obscured by our mezzanine interface board, so a small stick may be needed in order push the switch.
3. Turn on the Tracker and Energy-Detector power supplies (two black BKPRECISION boxes with switches on the back), the Tracker bias power supply (Sorensen XEL supply with the power switch on the front), and the Energy Detector PMT high-voltage supply (ISEG supply with the switch on the back).
4. Turn on the controller for the stage (switch on the back).
5. Type “python /home/daq/powerSupplyGUI/powersupplyGUI.py”. This should open up a GUI window for operating the power supplies. It is supposed to connect automatically to all of the supplies over USB. If it bombs, you can try unplugging and plugging back in the main USB cable and then try the GUI again. If that does not work, the following command seems to bring it to its senses:

udevadm info –a –p $(udevadm info –q path –n /dev/device\_name)

Where ‘device\_name’ is replaced by the offending supply that does not want to communicate: sorensen-bias, bk-1003, bk-1046, or iseg-pmt.

1. Using the GUI pull-down menus, turn on the tracker and energy detector supplies. You should see a voltage of 5.5V for the energy detector and 5.0V for the tracker. If not, use the menus to set them so. The tracker should show a current of around 3.5 amps while the energy detector should show about 1.3 amps.
2. On the PMT HV supply, switch on the two front-panel switches to enable the HV output. Make sure that the voltage setting is negative and that the left channel reads about −365V while the right channel reads about −800V.
3. Use the GUI to set the tracker bias voltage to 80V and then turn on the output. Double-check all of the voltage settings.

Monitor the tracker and energy detector voltages and currents as follows:

1. Type “python /home/daq/powerSupplyGUI/powerMonitorGUI.py”. You should see a large GUI window open and display the voltages and currents on each of the 8 tracker boards and the two energy-detector digitizer boards.
2. Click on the checkbox at the bottom left to turn on the tracker ASIC power. You should then see voltage and current for every supply. You should also see on the power supply GUI that the tracker current increases to between 16 and 17 amps.
   1. Check that each supply on each board is close to the expected voltage (1.2V, 1.8V, 2.1V, 2.5V, or 3.3V). The analog 3.3V is fairly critical. If it reads below 3.2V on any of the T boards, then you may need to increase the tracker voltage from the nominal 5.0V up to no more than 5.5V. However, if you do so, keep in mind that this indicates an ohmic loss in a connection somewhere. Keep an eye on the connector for the tracker power into the mezzanine board, as it has been known to go bad and turn black and brown from excessive heat.
   2. Check that boards of the same type are drawing about the same amount of current for each supply. Note that you will not see current from the 1.8V supplies, as those only supply current during the FPGA programming.
   3. Check the bias current readings for the V boards, but ignore the T-board bias current readings (they are internally disconnected to solve a noise problem). You can expect a few hundred micro-amps per board, but you will see the current steadily rise as time progresses. Mainly you want to verify that the currents are not zero.
   4. Check that the temperatures look reasonable (near room temperature if you just recently turned the system on, but in any case well below 35C).
3. Click the ASIC current off.
4. Delete the monitor window. It cannot operate simultaneously with the data acquisition program.

Note that the tracker and energy detectors must always be powered on *after* the event builder is powered on. That is because the event builder FPGA must be operating before booting the FPGAs on the tracker and detector boards. So if you power cycle the event builder (using its switch) you must then power cycle the detector boards (using the power supply control GUI).

Also, keep tracker ASICs powered off (such that the tracker supply reads only around 3.5A of current) whenever a run is not in progress. That will help to avoid excessive heating inside the tracker enclosures.

It is a good idea to keep the tracker bias voltage off during any extended period between runs, because the bias current tends to creep upward during operation. Just be sure to turn it back on before starting a run!

# Configuration Setup

Open the file /home/daq/git/pct-acquire/pct\_config.yaml in an editor (e.g. gedit). Keep it open during the beam test, for convenience. Make sure that the following parameters are set properly (since these often get changed around for testing purposes):

* event\_builder:
  + timepix: false
  + spill\_synch: false (unless operating at LLU, in which case true may be desired)
  + tracker\_trigger\_mask: 0
  + energy\_trigger\_mask: 1
  + tracker: asic\_default: trigger\_delay: 2
  + buffer\_manager\_mode: new
  + energy\_detector:
    - data\_type: reduced
    - sample\_config: [6, 8]
    - trigger\_delay: 7
* scan\_config: (all of these may be set by hand from the GUI)
  + starting\_run\_number: (usually this is set by hand for each run; 0 is a good default to prevent overwriting data in case you forget to set the run number)
  + filename\_pattern: (set this to point to the SSD area for data acquisition)
  + max\_seconds: 20 (decrease or increase according to the maximum run time desired for each stage angle)
  + max\_bytes: 200000 (this is what normally stops the run at each angle)
  + start\_angle: 0
  + stop\_angle: 360 (for a full scan)
  + angle\_step: 4
* energy\_detector:
  + data\_type: reduced
  + pedestal\_output: true
  + sample\_config: [6, 8]
  + trigger\_delay: 7
  + read\_delay: 10
  + num\_samples: 16
  + trigger\_type: or
  + boards:
    - hardware\_label: E3
      * thresholds: [3., 3., 3.]
      * num\_channels: 3
      * trigger\_mask: 7
    - hardware\_label: E1
      * thresholds: [12., 12., 12.]
      * trigger\_mask: 0

You may want to change the “monitor\_supplies” and “monitor\_temperature” settings on the various boards. If set “true” they will result in the system reading the supply voltages and/or temperatures of the boards during the configuration stage prior to a run, with the results displayed on the pct-acquire GUI. The reading is rather slow, however, so that having all or many of them set “true” greatly slows down the configuration process.

# Executing a Scan

Make sure that everything is powered on, including the tracker bias! Start the pct-acquire from the folder “/home/daq/pct-acquire/” by the command

./pct.py log-file-name

where the log-file name and path should include the starting run number for the scan and ideally should be located in the SSD area where the data are sent. You should see the pct-acquire GUI window pop open. To start a scan going, do the following:

1. Click on the “Control Stage” button. If this is the first run of the session, click “Home” and wait for the stage to rotate one way and then the other until it says that it is homed. If the program cannot connect to the stage, you may have to power cycle the stage controller, restart the GUI, and try again.
2. Click on the “Configure Detectors” button. It will then take a while to go through the configuration sequence. While it is doing so you can work on the next step.
3. Set up the scan configuration:
   1. For a stepped scan:
      1. Set the “Starting run number” to a unique value for the beam test and write the number in the log book.
      2. Make sure that the “Filename Pattern” is correct. It should start with the SSD folder location for data from the current beam test followed by a descriptor of the run, such as “Emp” for an empty run, “Cal” for WEPL calibration, or the name of a phantom. Then should follow patterns for the run number and stage angle.
      3. “Max Seconds” and “Max Bytes” determine when the run will stop. 200,000 bytes will give around 4 million events. (Note that usually by the time the system notices that the byte count has exceeded Max Bytes and manages to halt the run the system has accumulated a few more tens of millions of events.)
      4. Make sure that the stage start, stop, and step values correspond to what is desired for the scan.
   2. For a continuous scan:
      1. Set the run number.
      2. Enter the entire filename and path (don’t rely on the automatic pattern completion).
      3. “Set the Max Seconds” to the full time of the scan, typically 360.
      4. Set the “Max Bytes” to a very large value, to avoid early termination of the scan.
      5. Click the box for continuous scan.
4. Once the configuration is complete, check the “Supply Monitor” readings, if there are any. You may need to click the “Reset Ethernet” button, especially if the previous run was aborted without closing everything down properly. If the system fails to link of with the event builder over Ethernet, then you will have to exit the GUI and start over. Hitting the reset doesn’t cost much time and can prevent connection problems.
5. Finally, click the “Start Scan” button.

During the run the GUI will display the bytes accumulated and bytes written to disk. When the run terminates for a given angle setting a display of the trigger and event rates during the first 10 seconds will pop up. Also, after each angle the run will be closed, run summary information printed to the log, the stage rotated, a new data file opened, and a new run started. All of that takes a few seconds. (Capability for doing runs with continuous stage rotation is being worked on.)

When the scan is finished, click the “Quit” button to close the GUI. It is best to restart it from scratch for the next scan. You may need to send the stage back to the home position before mounting the next phantom, in which case you should reopen the GUI and home the stage. Don’t try to home the stage before closing the GUI after the previous run, as it then tends to hang the entire process. In any case, be sure to home the stage before starting the next new scan.

If the system will not be operated for an extended time period, use the power control GUI to turn off the tracker power and tracker bias. This will help keep the system from overheating and keep the bias currents from getting too high and is especially important if the ASIC power does not switch off at the end of a run (which seems to be a typical problem after continuous scans). But don’t forget to turn the power back on, especially the bias, since the system will readily start a new run with the bias turned off (in which case you will get lots of noise hits in the tracker, resulting in a low event rate and high deadtime).

# Data Verification

See the end of this document for a checklist to be used to make sure that the setup is correct and is producing good data.

It is a good idea at the beginning of a beam test to take an “empty” run with data reduction turned off in the energy detector FPGAs. That is done in the pct\_config.yaml file according to the line

* energy\_detector:
  + data\_type: samples

Just be sure to set it back to “reduced” before making scans. To verify the basic instrument performance, run the single-event display program in the /home/daq/git/pct-acquire/ folder:

python display\_events.py file-name

It will open a window and display the first event. To go to the next event just delete the window. Use ctrl-c to exit the program. Make sure that you see tracker hits forming reasonably straight lines and energy detector pulses for each stage in which the entire pulse fits into the 16 available samples, with the peak around sample 4 or 5. The pulse position can be adjusted with the “energy\_detector: trigger\_delay” parameter in pct\_config.yaml, but normally no adjustment should be needed. The timing has been stable for years.

It is also a good idea to take one or more reasonably long (millions of events) “empty” run with data reduction turned on. When the run is finished, execute the monitoring program to verify the data. There are two possibilities:

1. The old monitoring program (no longer maintained, so not a great idea). In the /home/dat/git/pct-acquire/ folder execute the C++ program: “./DataMonitor file\_name num\_events” where the last argument (optional) is the number of events to analyze (you can analyze all the millions if you are patient, but set to 100000 for a quick look). The monitoring program will create a bunch of .png files or gnuplot .gp files that you should then look at one by one.
2. The new monitoring code, which runs within the preprocessing code. This is located in the /home/dat/git/Preprocessing/bin/ folder. Execute pCT\_preprocessing with the options “–u yes -l 2 -n 10000000 –o outputDirectoryName” and provide the filename as an argument. It will create a bunch of gnuplot .gp files that you should look at one by one. All of the options can be set on the command line or by editing the defaults in the file pCT\_confit.txt. The –o option is recommended to direct the output to a different folder so as not to fill the working directory with plots.

To view the gnuplot files, enter “gnuplot” to start that program, and then use the command ‘load “xxxx.gp”’ in order to view the plot “xxxx.gp”. For each tracker cassette you should see strip hit maps that reflect the beam profile and, hopefully, don’t have many dead or noisy strips. For the energy detector you should see a single-proton peak, at least, for each layer plus more-or-less uniform T and V profiles. For the tracker you should see good T and V residuals plus efficiencies in the 98%-plus range. The number of tracks per event should peak at 1, with not more than 10 or 20 percent of events having two tracks.

*Note that the new preprocessing program requires two calibration files usually named TVcorr.txt and Wcalib.txt.* Each has near the bottom a couple of cards specifying a date range, which you may have to adjust to get the preprocessing program to go. Also, if the calibration constants in those files are bad or too old, the energy plots from preprocessing can be systematically off.

# System Shut-Down

Use the power control GUI to power off the tracker bias and the tracker and energy detector supplies. Manually turn off the energy detector high voltage. Then power off the stage controller and the tracker and energy detector power supplies and bias voltages. Finally, turn off the event builder power.

# Problem Diagnosis

* The dead time is very high. For example, the trigger rate might be a megahertz whereas only 100 kHz of data are being logged.
  + The tracker bias voltage is not turned on, resulting in a huge number of noise hits that clog the datastream.
  + The energy detector is not set to reduce its data in the FPGA, resulting in a large data rate in order to read out the individual digitizations.
* There are no triggers
  + Check that the energy detector PMT bias is turned on and set to the correct voltages.
  + Check the trigger settings in the pct\_config.yaml file.
* The data acquisition will not start up or run.
  + Try cycling the power of the event builder, followed by cycling the power of the energy detector and tracker.
  + If the problem is with connecting to the Ethernet, check the Ethernet connections and click the reset-Ethernet button before starting the run.
* The detector configuration procedure hangs because a single tracker ASIC is not responding to queries from the event builder.
  + Disable the ASIC in the pct\_config.yaml file.

# Operating Checklist

Use the following checklist after the system is set up and powered on to be sure that the operating settings are correct and the data look good.

|  |  |
| --- | --- |
| **Check** | **Item (Check values read, not just the settings)** |
|  | Tracker bias voltage of 80V to 100V. Make sure it is always powered on during running. *If it is not on you will likely see a large dead time due to trying to read very large events full of noise.* It may be necessary to increase the supply current limit in order to reach the desired voltage. |
|  | PMT bias of −365V (left channel) and −800V (right channel). HV turned on! |
|  | Energy detector power supply voltage of 5.5V and current ~1.3 A |
|  | Tracker air blower connected and operating. |
|  | Tracker detector power supply voltage of 5.4V and current ~3.5 A (~16.5 A with ASICs on, but avoid leaving the ASICs on when not taking data, for risk of overheating). |
|  | Run powerMonitorGUI.py and check all boards:   1. Digital 1.2V supply (D1.2V) between 1150 and 1250 mV and current < 200 mA 2. Digital 1.8V supply (D1.8V) between 1750 and 1850 mV and current ~0 3. Digital 2.5V supply (D2.5V) between 2450 and 2550 mV and current < 500 mA 4. Analog 2.5V supply (A2.5V) between 2400 and 2500 mV and current < 1550 mA 5. Analog 3.3V supply (A3.3V) between 3150 and 3300 mV and current < 250 mA 6. Analog 2.1V supply (A2.1V) between 2000 and 2100 mV and current < 650 mA 7. V-board SSD Bias somewhere between 300 and 800 uA (increases gradually). Ignore the T-board bias reading, as they are not connected internally. 8. Energy detector 1.2V (E1.2V) between 1150 and 1250 mV and current < 100 mA 9. Energy detector 3.3V (E3.3V) between 3250 and 3350 mV and current < 250 mA 10. Energy detector 5.0V (E5.0V) between 4800 and 5000 mV and current < 300 mA 11. Energy detector 1.8V (E1.8V) between 1750 and 1850 mV and current ~0 12. Tracker board temperatures not higher than around 33 degrees C. Avoid leaving the tracker powered on with ASICs powered for long periods of time, and make sure that the blower is always on. |
|  | Double check the following configuration settings (in pct\_config.yaml)   1. Under event\_ builder:    1. time\_stamp\_out = on    2. tracker\_trigger\_mask = 0    3. energy\_trigger\_mask = 1    4. buffer\_manager\_mode = new 2. Under asic\_default:    1. buffer\_speed = 4    2. trigger\_delay = 2    3. trigger\_window = 1    4. max\_clusters = 10    5. output\_drive = 3    6. gain = 0 (or 1 if running with helium ions) 3. Under energy\_detector    1. data\_type = reduced    2. pedestal\_output = true    3. sample\_config = [6, 8]    4. trigger\_delay = 7    5. read\_delay = 10    6. num\_samples = 16    7. trigger\_type = or    8. Energy board E3 trigger\_mask = 7 |
|  | Take a short empty run with samples written and view the results on the single-event display.   1. The energy detector pulses are peaking at around the 4th or 5th sample. 2. Most of the events with good energy have 8 hits (4V and 4T) aligned in the tracker, and little noise. 3. Most of the events have the proton stopping in the last stage. |
|  | Set the energy detector mode back to “reduced”. |
|  | Take a 10s empty run and verify the following   1. Trigger rate not higher than about 1.2 MHz, with the dead time not more than about 10% 2. In the event display most protons stop in stage 5 and most events have all tracker hits 3. Run the preprocessing program on ~10 million events and look at the following plots. (Note that the last two plots require processing level 2, whereas for the others level 0 is sufficient.)    1. nTracks.gp: most events (>80%) should have exactly 1 track.    2. Tclusters.gp and Vclusters.gp: most events should have a single cluster per layer.    3. Cassette1.gp: verify the beam profiles in V and T.    4. Vefficiency.gp and Tefficiency.gp: efficiencies should be >98% and residual distributions should be Gaussian and centered on zero.    5. TrgBits.gp: the first 3 stages should all contribute to triggers, and the bit for the first stage should be on almost all the time (i.e. nearly equal to the number of events).    6. EnergyDetector.gp: you should see narrow pedestal peaks and high single-proton peaks located       1. around 3000 ADC counts for the first and second stages       2. around 2300 ADC counts for the third stage       3. around 5500 ADC counts for the fourth stage       4. around 4000 ADC counts for the fifth stage    7. Pedestals\_0.gp: each stage should have a sharply peaked distribution.    8. Energies\_0.gp: the sum-of-stage-energies should make a narrow peak just below 200 MeV. |

1. The negative terminal may or may not be strapped to the ground terminal. There is no need for it to be connected there to ground, as it is referenced to ground on the scanner. Connecting to the ground at the supply may create a ground loop but is probably harmless. In any case the bias current flows from the HV terminal and into the negative supply terminal, which *must* be connected. [↑](#footnote-ref-1)