

Stanford charge-readout Ubuntu DAQ

This page is about the computer running Ubuntu 14.04.4 LTS in Varian 022. The computer is on the bottom shelf of the grey rolling cart.

Contents

Plots

Connection info

- Logging in
- SSH connection
- VNC connection

Notes about latest SiPM-triggered testing, May 2016

- connecting to the digitizers
- DAQ GUI notes
- data-taking script

NGMDAQ code from Jason Newby at ORNL

- prerequisites
- notes about using offline with ROOT6 / NGM
- get code
- building code
- FFTW
- running code
- troubleshooting

Struck ROOT GUI

- prerequisites
- run the code
- build the code, if needed

Static IP addresses for connecting to SIS3316

- IP address assignments of ethernet cards in `/etc/network/interfaces`
- MAC Address assignments for ethernet cards in `/etc/udev/rules.d/70-persistent-net.rules`
- Check the settings with `ifconfig`
- Usual arp output
- checking UDP traffic

EXO-200 SW on this machine

~~nEXO MC SW on this machine — this was lost in hard drive failure, June 2016~~

- Using a geantino beam to explore the geometry
- nEXO MC for ssh user

Other SW

Plots

Plots are drawn by a python script in the Dropbox: `/home/teststand/Dropbox/ubuntuDaq/systemReport.py`

The 2 resulting plots are in the Dropbox and online:

- `/home/teststand/Dropbox/ubuntuDaq/monitoring_plots/`
- plots from the Ubuntu DAQ (<https://www.dropbox.com/sh/aktgmozolv3x6z9/AADuILYG3S-FLUovQuancXT5a?dl=0>)

This script is called every 5 minutes by a cron job. You can edit the crontab with:

```
crontab -e
```

This is the line that runs the script:

```
* /5 * * * * /home/teststand/Dropbox/ubuntuDaq/systemReport.py > /home/teststand/Dropbox/ubuntuDaq/monitoring_plots/cron_log.txt 2>&1
```

Connection info

Logging in

If you are in Varian 022, you can log on to the account teststand with a password that is our favorite isotope.

SSH connection

If you are connecting remotely, the teststand account only allows login with an ssh key.

```
ssh -Y teststand@171.64.56.173
```

The ssh-user account allows login with a password:

```
ssh -Y ssh-user@171.64.56.173
```

The password is the same as for the windows computer.

VNC connection

We use x11vnc as the VNC server, and we tunnel over ssh.

From your machine, ssh to the Ubuntu DAQ and bind the ports:

```
ssh -t -L 5900:localhost:5900 teststand@171.64.56.173 'x11vnc -rfbauth ~/.vnc/passwd -display :0 -localhost'
```

It seems like teststand must be logged on downstairs for this to work:

Run this command on the Ubuntu DAQ:

```
teststand@022-Ubuntu:~$ startupvnc
```

(It is an alias for this command:)

```
teststand@022-Ubuntu:~$ x11vnc -rfbauth ~/.vnc/passwd -display :0 -localhost
```

On your machine, connect your VNC viewer to localhost:0. The Real VNC viewer works for Alexis. The VNC password is the same as the Windows computer login: u*****.

Notes about latest SiPM-triggered testing, May 2016

connecting to the digitizers

- edit and run `--/configure_udp_localNet_eth` (https://github.com/Charge-readout-test-stand/charge-readout-scripts/blob/master/struck/ngm/configure_udp_localNet_eth) (should be set up for SN 97, 98 and DT unit (SN 155) plugged in to new ethernet switch, which is plugged in to eth2):

```
sudo --/configure_udp_localNet_eth
```

- some notes about the ethernet switch are in logbook post: "ethernet switch for SIS-DAQ"
 - switch port 1 is connected to eth2. Switch should be accessible from the DAQ GUI in the web browser at: <http://192.168.2.105>
 - switch ports 2, 3, 4 are the digitizers. We may want to test this — having one digitizer in eth1 may be faster
 - login: xenon1 password: our favorite isotope
- helpful command: `arp -a`
- test the connection with the DAQ gui

DAQ GUI notes

- need to edit the gui when switching between the DT and VME units. Comment this line for the VME units, uncomment for the DT unit in `struckRootGUI/rootGUI/src/sis3316_ethernet_access_class.h` (https://github.com/Charge-readout-test-stand/struckRootGUI/blame/master/rootGUI/src/sis3316_ethernet_access_class.h#L32):

```
#define VME_FPGA_VERSION_IS_0000_OR_HIGHER
```

- run `make clean`; make after editing

data-taking script

- seems to work ok with pulser trigger, but not working yet with DT unit: `takeData_3digi.py` (https://github.com/Charge-readout-test-stand/charge-readout-scripts/blob/master/struck/ngm/takeData_3digi.py)

- there are 2 or three locations where the number of cards/slots is hard-coded

NGMDAQ code from Jason Newby at ORNL

prerequisites

These things were done in June/July 2016 to install dependencies after hard drive failure -- probably don't need to do this again soon..

root 6.04.18:

```
teststand0022-Ubuntu: /software/root/build_root6_04_18 sudo apt-get install git dpkg-dev make g++ gcc binutils libx11-dev libxpm-dev libxft-dev libxext-dev
teststand0022-Ubuntu: /software/root/build_root6_04_18$ cmake ../root-6.04.18/
teststand0022-Ubuntu: /software/root/build_root6_04_18$ cmake --build --
```

Had trouble building with make -j, so using cmake --build. Had to install python-dev to fix an issue where cmake didn't find python:

```
sudo apt-get install python-dev
```

before this, cmake was reporting:

```
Could NOT find PythonLibs (missing: PYTHON_LIBRARIES PYTHON_INCLUDE_DIRS)
```

Added this to .bashrc:

```
# root 6.04.18+
source /software/root/build_root6_04_18/bin/thisroot.sh
```

Intel TBB: Downloaded and installed Intel threaded building blocks from here: tbb (<https://www.threadingbuildingblocks.org/download>) In .bashrc:

```
# TBB
source /software/tbb44_20160526oss/bin/tbbvars.sh intel64 linux auto_tbbroot
```

```
export NGM=/home/teststand/software/ngm_sw/install_ngm_daq
export LD_LIBRARY_PATH=$NGM/lib:$LD_LIBRARY_PATH
```

sqlite3 installed:

```
sudo apt-get install sqlite3 libsqlite3-dev
```

notes about using offline with ROOT6 / NGM

NGM compiles against root 6. I didn't have any luck compiling EXO offline against root 6, even after modifying the make files. I think there is a difference between root 5 and 6 in the way the root dictionaries are generated:

As a workaround, I moved some of the EXO wfm classes into NGMDaq/EXOUTilities. To do this, I had to do several steps. An example with EXOBaselineRemover:

- copy EXOBaselineRemover.cc and .hh files from offline/utilities/misc to NGMDaq/EXOUTilities
- modify #include statements in .cc and .hh files to remove EXOUTilities from paths, e.g. #include "EXOUTilities/EXOWaveformTransformer.hh" -> #include "EXOWaveformTransformer.hh"
- add EXOBaselineRemover to EXOUTilities_LinkDef.h
- add EXOBaselineRemover.cc and EXOBaselineRemover.hh to CMakeLists.txt

Needed to delete contents of build directory, reconfigure & recompile NGM after this:

get code

```
svn co svn+ssh://user@svn.slac.stanford.edu/svn/slac/g/exo/subversion/collaborators/trunk/alexis/NGMDaq
```

building code

Used cmake to build:

```
teststand0022-Ubuntu: /software/ngm_sw$ mkdir build_ngm_daq
teststand0022-Ubuntu: /software/ngm_sw$ cd build_ngm_daq/
```

```
teststand@022-Ubuntu: /software/ngm_sw/build_ngm_daq$ cmake --DCMAKE_INSTALL_PREFIX=/software/ngm_sw/install_ngm_daq /software/ngm_sw/NGMdaq
teststand@022-Ubuntu: /software/ngm_sw/build_ngm_daq$ make --j4 install
```

Usually just make clean and make to implement changes:

```
teststand@022-Ubuntu: /software/ngm_sw/build_ngm_daq$ make clean; make --j4 install
```

If you add a new module to EXOUtilities you need to edit the EXOUtilities/CMakeLists.txt and the EXOUtilities_LinkDef.h file with the added .cc and .hh files. Also you need to rerun cmake and make.

When we added in the fftw dependent things (Matched Filter and FastFFT) from EXOUtils we had to add in at the top of both EXOFastFourierTransformFFT.cc and EXOMatchedFilter.cc the following lines

```
#define UGB_ROOT_FFTW 1
#define HAVE_FFTW 1
```

both where needed to get make to compile and recognize fftw functions such as fftw_free(). If make complains about fftw functions check for these lines:

Also to check if you have fftw in the root install run at the command line

```
root-config --features
```

which should return something like

```
asimage astiff builtin_afterimage builtin_ftgl builtin_glew builtin_pcre builtin_lzma builtin_llvm cxx11 cling exceptions explicitlink fftw3 fortran genvector-
memstat pch python shadowpw shared sqlite thread tmva vdt xft xil
```

Instructions are included on how to get and install FFTW from source but I actually don't think that is needed since ROOT already has fftw as part of its installation. Also checking to see if fftw is enabled can also be done using

```
root-config --has-fftw3
```

FFTW

Not sure if this is needed:

Need fftw now if we end up using Matched Filter. Brian setup fftw at NERSC for offline so using 3.3.4 even though 3.3.5 seems to be the most recent version as of 10/25/2016:

```
wget ftp://ftp.fftw.org/pub/fftw/fftw-3.3.4.tar.gz
tar -zxvf fftw-3.3.4.tar.gz
mkdir fftw-3.3.4_install
cd fftw-3.3.4
./configure --prefix=/path/to/install/fftw-3.3.4_install --enable-shared --with-pic
make --j4
make install
```

running code

```
root -b -q /software/ngm_sw/NGMdaq/test.c
```

troubleshooting

- check sisreadthread.log for temperature exceeded messages. A fan should always be blowing on the DT unit; it has overheating issues
- check siswritethread.log for messages. If the data rate is too high, every single buffer spill will have a message that says "writing skipped":
 - Begin-writing next spill Total Previous Bytes(831064) Writing Skipped
- We once saw very noisy traces & unreliable behavior when one of the VME digitizers wasn't properly seated in the VME pizza box.
- Either of these messages at the end of the output from takeData*.py indicates that something is wrong (unit is powered off, wrong ethernet interface is being used, LVDS cable isn't installed properly, a VME module isn't in the crate correctly, or an ethernet optical connector isn't seated properly):
 - Channel 1:0 did not receive the expected number of words 65536(10066330
 - Max tries exceeded for channel: 0 32000000, Previous Bank: 1

Struck ROOT GUI

For data collection with one SIS3316, we can use the code from Struck that Peihao modified: [struckRootGUI](https://github.com/Charge-readout-test-stand/struckRootGUI) (<https://github.com/Charge-readout-test-stand/struckRootGUI>)

As of June 2016, this was working to talk to one card with a regular ethernet cable (not a patch/crossover cable) between the added-on PCIe card and the Struck card in the VME pizza-box DAQ, S/N 98:

There is a line that must be commented out to communicate with cards with older firmware, in the VME crate: [struckRootGUI/rootGUI/src/sis3316_ethernet_access_class.h](https://github.com/Charge-readout-test-stand/struckRootGUI/blame/master/rootGUI/src/sis3316_ethernet_access_class.h#L32) (https://github.com/Charge-readout-test-stand/struckRootGUI/blame/master/rootGUI/src/sis3316_ethernet_access_class.h#L32)

```
/**define VME_FPGA_VERSION_IS_0000_OR_HIGHER
```

Notes about using the code are here: [Stanford charge-readout test stand Struck SIS3316 notes](#)

prerequisites

It's working with older root, 5.34:

```
source root5
```

this is an alias for: `source /home/teststand/software/root/root_v5_34_36/bin/thisroot.sh`

Installed both of these packages:

```
sudo apt-get install fftw3-dev fftw-dev
```

run the code

To run the code:

```
source /home/teststand/software/root/root_v5_34_36/bin/thisroot.sh
sudo /home/teststand/software/charge-readout-scripts/struck/ngm/configure_udp_localnet_eth
cd /software/struckRootGUI/rootGUI/Release/
./sis3316_root_gui
```

this should open the familiar Struck x11 window on your computer

terminal output looks like this...

```
sis3316_eth::set_UdpSocket0103316_IPAddress()
IP: 192.168.2.100
port: 57344
return code: 0
...
```

build the code, if needed

To build the code, we always need to call make clean, then make:

```
teststand022-Ubuntu: $ cd /struckRootGUI/rootGUI/Release/
teststand022-Ubuntu: /struckRootGUI/rootGUI/Release$ make clean; make
```

Static IP addresses for connecting to SIS3316

As of June 2016, the Ubuntu DAQ has one built-in ethernet card, and two PCI-express cards that we added:

IP address assignments of ethernet cards in /etc/network/interfaces

The only way I have been able to get the computer to reliably assign & keep a static IP address for the PCIe ethernet cards is by adding info for eth1 and eth2 in /etc/network/interfaces:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
# AGS added 17 June 2016:
auto eth1
iface eth1 inet static
    address 192.168.1.99
    netmask 255.255.255.0
auto eth2
iface eth2 inet static
```

```

address 192.168.2.99
netmask 255.255.255.0

```

After changing this, I restarted the computer and the settings have persisted.

MAC Address assignments for ethernet cards in /etc/udev/rules.d/70-persistent-net.rules

After one reboot, the names eth0, eth1, eth2 were shuffled around and the computer could no longer connect to the internet, since one of the 192.168.X.99 addresses was assigned to the ethernet card on the Stanford network. To fix this, modified /etc/udev/rules.d/70-persistent-net.rules

```

# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME key.

# PCI device 0x0006:0x15b0 (e1000e)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="64:00:6a:72:8d:cb", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x10ec:0x0160 (r0169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="60:e3:27:01:a6:0b", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"

# PCI device 0x10ec:0x0160 (r0169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="60:e3:27:01:43:2c", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth2"

```

Check the settings with ifconfig

```

teststand022-Ubuntu: /software/struckRootGUI/rootGUI/Release$ ifconfig | grep cast
inet addr:171.64.56.173 Bcast:171.64.59.255 Mask:255.255.252.0
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:192.168.2.99 Bcast:192.168.2.255 Mask:255.255.255.0

```

Usual arp-output

```

teststand022-Ubuntu: /software/struckRootGUI/rootGUI$ arp -a
? (192.168.1.100) at 00:00:56:31:60:61 [ether] PERM on eth1
waa-rtz v1644-GUNet (171.64.56.1) at 00:1b:17:00:05:30 [ether] on eth0
? (192.168.2.100) at 00:00:56:31:60:62 [ether] PERM on eth2

```

```

teststand022-Ubuntu: /software/struckRootGUI/rootGUI$ arp -n
Address HWtype HWaddress Flags Mask Iface
192.168.1.100 ether 00:00:56:31:60:61 CM eth1
171.64.56.1 ether 00:1b:17:00:05:30 C eth0
192.168.2.100 ether 00:00:56:31:60:62 CM eth2

```

checking UDP-traffic

tcpdump is the best tool I've found for convincing myself that things are working.

Here 192.168.2.99 is the ethernet card eth2 and 192.168.2.100 is the Struck card with S/N 98.

```

sudo tcpdump -i eth2 'host 192.168.2.100 and udp and port 57344'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes

```

When the "Validate IP address" button is pressed on the Struck root gui, this kind of output appears:

```

11:26:52.226095 IP 192.168.2.99.57344 > 192.168.2.100.57344: UDP, length 1
11:26:52.227407 IP 192.168.2.99.57344 > 192.168.2.100.57344: UDP, length 5
11:26:52.227441 IP 192.168.2.100.57344 > 192.168.2.99.57344: UDP, length 9
11:26:52.227506 IP 192.168.2.99.57344 > 192.168.2.100.57344: UDP, length 9
11:26:52.227512 IP 192.168.2.99.57344 > 192.168.2.100.57344: UDP, length 9

```

EXO-200-SW on this machine

With root 5:

```

teststand022-Ubuntu: /software/offline/make$ ./configure --disable-viewer3d --disable-display --without-mysql --without-geant4 --without-exobinary

```

FAIL: trying to build offline w root6:

```

teststand022-Ubuntu: /software/offline/make$ ./configure --disable-viewer3d --disable-display --without-mysql --without-geant4 --without-exobinary CXXFLAGS=-std=c++11

```

~~nEXO MC SW on this machine -- this was lost in hard drive failure, June 2016~~

June 20 2016:

Geant4:

```
teststand@022-Ubuntu: /software/geant4/build_geant4.10.01$ cmake -DCMAKE_INSTALL_PREFIX=/software/geant4/install_geant4.10.01/ ../geant4.10.01
-DGEANT4_INSTALL_DATA=ON -DGEANT4_USE_XM=ON -DGEANT4_USE_OPENGL_X11=ON -DGEANT4_USE_RAYTRACER_X11=ON
```

VGM:

```
teststand@022-Ubuntu: /software/vgm/build$ cmake -DCMAKE_INSTALL_PREFIX=/software/vgm/install ../src-4.2
teststand@022-Ubuntu: /software/vgm/build$ make -j12 install
```

nEXO MC branch:

```
teststand@022-Ubuntu: /software/nEXO/build_LXe_Test_Stand_MC$ cmake -DVGM_DIR=/software/vgm/install/lib/VGM-4.2.0/ -DGeant4_DIR=
/software/geant4/install_geant4.10.01/lib/Geant4-10.1.0 -DCMAKE_INSTALL_PREFIX=/software/nEXO/build_LXe_Test_Stand_MC/ ../LXe_Test_Stand_MC/
teststand@022-Ubuntu: /software/nEXO/build_LXe_Test_Stand_MC$ make -j4
teststand@022-Ubuntu: /software/nEXO/build_LXe_Test_Stand_MC$ make install
```

Using a geantino beam to explore the geometry

Modify line 5 of TestStand_VRML.in to set tracking verbosity to 1 (this will add ~70,000 lines of output!)

```
/tracking/verbose 1 # change from 0 to 1 to see geantino stepping info
```

Run the software and pipe output to a file:

```
teststand@022-Ubuntu: /software/nEXO/LXe_Test_Stand_MC$ /software/nEXO/build_LXe_Test_Stand_MC/nEXO -m
/software/nEXO/LXe_Test_Stand_MC/Batch_Call_Scripts/stanfordLXeSetup/TestStand_VRML.in >& log.out
```

print the "transportation" steps where the geantino entered a new volume:

```
cat log.out | grep Transportation
```

The output should look something like this:

1	0	60	29.1	1	0	121	121 /nEXO/TPCVessel Transportation
2	0	60	27.1	1	0	2	123 /nEXO/TPCInternals Transportation
17043	0	60	10.2	1	0	2.00e-11	132 /nEXO/TPCInternals/LXe/ActiveRegion Transportation
54163	0	60	4.41e-16	1	0	0.0005	150 /nEXO/TPCInternals Transportation
72004	0	60	-8.92	1	0	0	159 /nEXO/TPCVessel Transportation
72005	0	60	-10.9	1	0	2	161 /nEXO/TPCExternals/HPE Transportation
72006	0	60	-631	1	0	620	701 /nEXO/TPCExternals/inCyro Transportation
72007	0	60	-633	1	0	2	703 /nEXO/TPCExternals/Vacuum Transportation
72008	0	60	-659	1	0	26	809 /nEXO/TPCExternals/OuterCryo Transportation
72009	0	60	-661	1	0	2	811 /nEXO/TPCExternals Transportation
72010	0	60	-4.99e+03	1	0	4.33e+03	5.14e+03 /nEXO Transportation
72011	0	60	-5e+03	1	0	9.00	5.15e+03 OutOfWorld Transportation

These are the column headings:

```
Step# X(mm) Y(mm) Z(mm) KinE(MeV) dE(MeV) StepLeng TrackLeng NextVolume ProcName
```

The /gps/pos/centre command controls the starting position of the beam in x,y, and z:

```
/gps/pos/centre 0. 6.0 15.0 cm
```

nEXO MC for ssh-user

To build from scratch:

```
ssh-user@022-Ubuntu: $ cd software/nEXO/build_LXe_Test_Stand_MC/
ssh-user@022-Ubuntu: /software/nEXO/build_LXe_Test_Stand_MC$ cmake -DVGM_DIR=/home/teststand/software/vgm/install/lib/VGM-4.2.0/ -
DGeant4_DIR=/home/teststand/software/geant4/install_geant4.10.01/lib/Geant4-10.1.0 -DCMAKE_INSTALL_PREFIX=/software/nEXO/build_LXe_Test_Stand_MC/
../LXe_Test_Stand_MC/
ssh-user@022-Ubuntu: /software/nEXO/build_LXe_Test_Stand_MC$ make -j4 install
```

To run the visualization:

```
ssh-user@022-Ubuntu: /software/nEXO$ /software/nEXO/build_LXe_Test_Stand_MC/nEXO-m-  
/software/nEXO/LXe_Test_Stand_MC/Batch_Call_Scripts/stanfordLXeSetup/TestStand_Vis.in
```

To build after modifying some code:

```
ssh-user@022-Ubuntu: $ cd software/nEXO/build_LXe_Test_Stand_MC/  
ssh-user@022-Ubuntu: /software/nEXO/build_LXe_Test_Stand_MC$ make -j4
```

Alternate visualization:

```
ssh-user@022-Ubuntu: /software/nEXO$ /software/nEXO/build_LXe_Test_Stand_MC/nEXO-m-  
/software/nEXO/LXe_Test_Stand_MC/Batch_Call_Scripts/stanfordLXeSetup/TestStand_VRML.in
```

This saves at .wrl file, which you can view on your own computer:

Here is one application for viewing: <http://freewrl.sourceforge.net/download.html> (<http://freewrl.sourceforge.net/download.html>)

Or Mitchell recommends this one: <http://www.cortona3d.com/cortona3d-viewers> (<http://www.cortona3d.com/cortona3d-viewers>)

Other SW

For tier3 generation:

```
sudo apt-get install python-numpy python-scipy python-matplotlib
```

Retrieved from "https://nexowiki.llnl.gov/index.php?title=Stanford_charge-readout_Ubuntu_DAQ&oldid=11785"

This page was last edited on 19 May 2017, at 03:24.