

# Photon Detector Calibration System and Plans for ProtoDUNE-HD II

Zelimir Djurcic et. al.

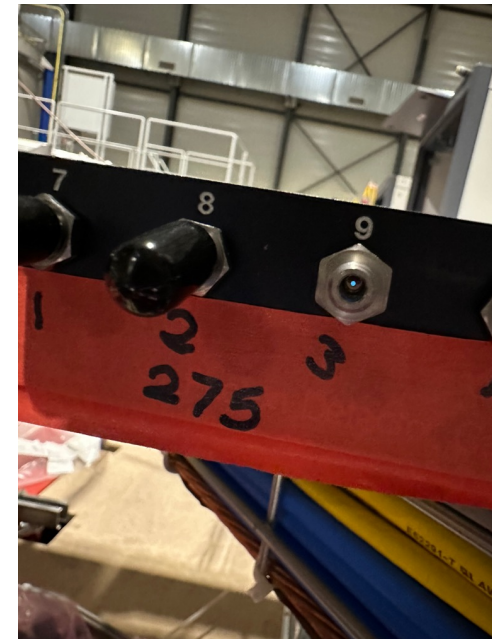
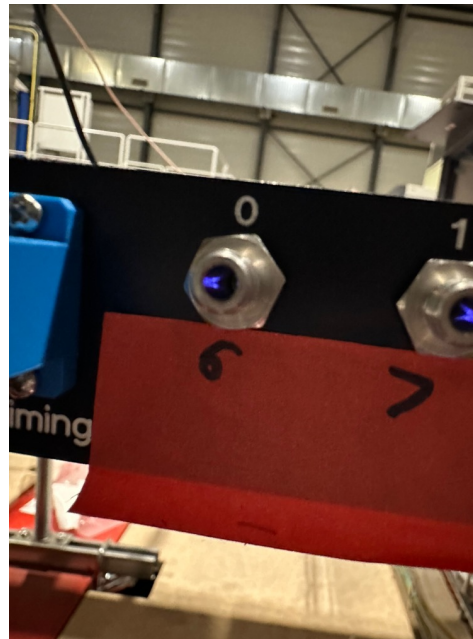
➤ See the Calibration System description at the DUNE DUNE collaboration meeting:

[https://indico.fnal.gov/event/60987/contributions/282788/attachments/174444/236498/DUNE\\_PDS\\_Calibration\\_Slides\\_Jan23\\_2024.pdf](https://indico.fnal.gov/event/60987/contributions/282788/attachments/174444/236498/DUNE_PDS_Calibration_Slides_Jan23_2024.pdf)

➤ Installed and integrated two calibration modules, please see the following figures:



➤ Installed and integrated two calibration modules, please see the following figures (cont.):



(light output verified with a cell phone, for all channels)

➤ Baseline Operation: DAQ-based

- Implemented the following operation based on requirements developed over past several year (DAQ discussions, completed reviews)
  - Have run the prototype system in ProtoDUNE-I with readout in the self-trigger mode at 1 pe level.
  - DAQ operation: run the calibration system on DAQ timing command
  - Currently running at the timing sync command (ID: 0)
  - Have tested the following operation: run double pulses at a 1kHz rate after the timing “pps” command, and emit 1000 pulses per command. May run “indefinitely” (\*).
  - Light observed at the back-end (see previous slides), test repeated several times.
- Many thanks to people who helped this quickly reviewed, clarified, reorganized, tested: Adam, Roland, M. Kirby, Wes, Dennis, D. Cussans ,Xavier, Giovanna, M. Oberling et. al.
  - See the next slide with screenshot of DAQ command line.
- Timing system group will implement trigger command ID7 and run trigger at 1KHz by sending the trigger to both Calibration and Daphne systems
  - Calibration system will be able to take it.

---

(\* ) Note: did not implement “Stop Run” script yet to shutdown the pulser, to be done.

```
(dbt) [zedjunci@np04-srv-011 CONFIG] nano ssp_conf/ zelinir-test
# aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
# Shonky NanoRC
# This is an admittedly shonky nano RC to control DUNE-DHQ applications.
# Give it a command and it will do your biddings.
# but trust it and it will betray you!
# Use it with care, user!
# aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
[09:59:38] INFO nano-conf-svc Flask lives on PID: 22030
FSM available states: ['none', 'booted', 'initial', 'configured', 'ready', 'running', 'paused', 'dataflow_drained', 'trigger_sources_stopped', 'error']
FSM available transitions: ['stop', 'scrap', 'start', 'abort', 'boot', 'disable_triggers', 'stop_trigger_sources', 'enable_triggers', 'drain_dataflow', 'terminate', 'conf']
Extra commands are []
INFO Using filelogbook
Running on the apparatus ssp_conf:
# aaaaaaaaaaaaaaaaaaaaaa
# ssp_conf
# aaaa ssp_conf
# aaaaaaaaaaaaaaaaaaaaaa
user@rc> boot
[09:59:51] INFO Subsystem ssp_conf is booting partition zelinir-test
INFO creating pm enqueuer
INFO booting task starting db://np04-srv-011:557/configuration?name=ssp_conf
Looking for services
ssp logs are in /np04-srv-011:/nfs/home/zedjunci/dev/CONFIG/log/ssp_5555.txt
# apps started
# ssp
[09:59:53] INFO booting task ending
INFO ResponseListener Flask lives on PID: 22016
INFO Application ssp booted
node.py:212
node.py:218
node.py:229
node.py:237
appctrl.py:71
node.py:38

# ssp_conf applications in partition zelinir-test
# aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
# a name a state a host a pings a last cmd a last succ. cmd
# a ssp_conf a initial a a a a a a
# aaaa ssp_conf a initial a a a a a a
# aaaa ssp a initial - alive a np04-srv-011 a True a None a None a
# a ssp_conf
# aaaa ssp_conf
# aaaa ssp
# aaaaaaaaaaaaaaaaaaaaaa
user@rc> conf
# acks received
# responses received
# ssp
# a ssp_conf
# aaaa ssp_conf
# aaaa ssp
# aaaaaaaaaaaaaaaaaaaaaa
user@rc> start 99
# acks received
# responses received
# ssp
```

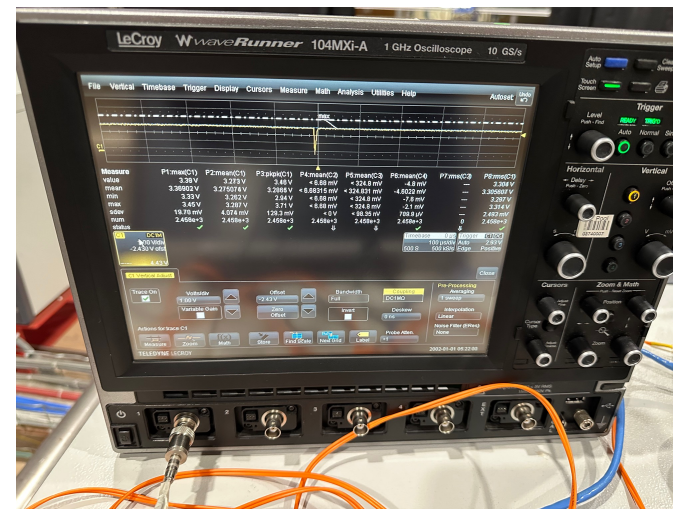
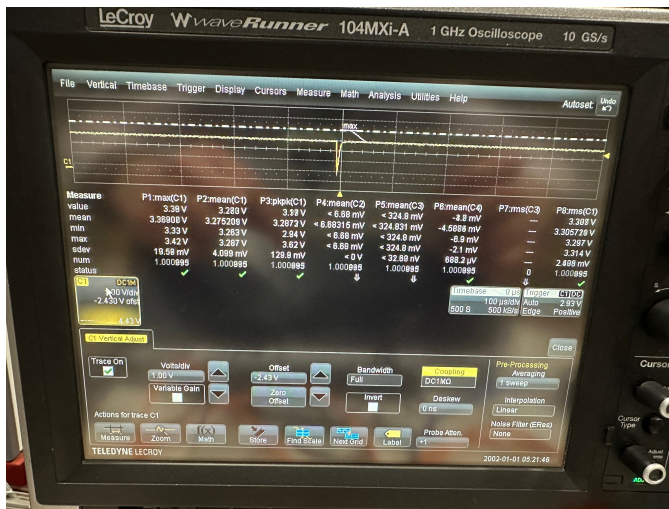
```
user@rc> start 99
# acks received aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 100%
# responses received aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 100% 0:00:00 0:00:00
# ssp aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa 100% 0:00:00 0:00:00

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa Started run #99, saving run data in /~/RunConf/99/ aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Run #99 ongoing
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa Type a TEST a
a Start time a 29/01/2024 10:00:57 a
a Duration a 0:00:00.003751 a
a Data storage enabled a True a
a Trigger rate a default from config (1Hz?) a
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa ssp_conf applications in partition zelinir-test
# a name a state a host a pings a last cmd a last succ. cmd
# a ssp_conf a ready a a a a a a
# aaaa ssp_conf a ready a a a a a a
# aaaa ssp a ready - alive a np04-srv-011 a True a start a start a
# a ssp_conf
# aaaa ssp_conf
# aaaa ssp
# aaaaaaaaaaaaaaaaaaaaaa
```



- Alternative Operation: still DAQ-based, but with electrical trigger output for current status of DAPHNE (if need be)
  - Tested the trigger electrical output from the front-end along with the DAQ-based tests described above
  - Signal observed by the scope (thanks to Manual!); may need to condition the trigger signal (see the pictures below for both modules)



➤ Optical fibers installed

-From the calibration module to optical feedthroughs, please see the figures below

-Zelimir installed these, but these are fabricated by David and Jairo (many thanks!); thanks to ANL who supplied feedthroughs.





➤ Next steps:

-Basic operation demonstrated: operation possible for readout in a self-trigger mode, with a DAQ trigger, and likely with the electrical trigger output.

-Need to document details and provide instructions on above elements.

-Need a student onsite to do more tests and operate the system and polish the DAQ: Talked to several people about this great opportunity (Jairo et al.)