First steps on accessing ProtoDUNE HD (light) data

J. Soto

Outline

This is a collection of useful links and basic commands towards accesing/analysing ProtoDUNE-HD data.

This is not a comprehensive guide, for that we have the wiki and the computing tutorials.

1. How to access data (samweb, metacat/rucio)
2. How to read data (using LArSoft/ROOT, and python)
3. How to understand data (channel maps).

How to access data (from lxplus)

```
$ source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
$ kinit $yourFNALuser@FNAL.GOV
```

Samweb (being migrated to metacat/rucio):

```
$ setup sam_web_client
$ samweb list-files run_type hd-protodune and file_type detector and "(run_number 22629 or run_number 22628)"
```

Metacat (list files):

```
$ setup metacat
$ export METACAT_SERVER_URL=https://metacat.fnal.gov:9443/dune_meta_prod/app
$ export METACAT_AUTH_SERVER_URL=https://metacat.fnal.gov:8143/auth/dune
$ metacat query "files from dune:all where core.run_type=hd-protodune and core.file_type=detector and core.runs=23387"
```

Rucio (locate files, need authorization, open ticket using your FNAL account):

```
$ setup rucio
$ setup kx509
$ kx509
$ export RUCIO_ACCOUNT=$yourFNALuser
$ rucio list-file-replicas hd-protodune:np04hd_raw_run023387_0183_dataflow8_datawriter_0_20231121T120241.hdf5
```

Last command will provide the location of the file. Using this path you can directly do:

```
$ ls /eos/experiment/neutplatform/protodune/dune/hd-protodune/17/38/np04hd_raw_run023387_0183_dataflow8_datawriter_0_20231121T120241.hdf5
```

Useful links:

How to access DUNE Computing: https://github.com/DUNE/FAQ/issues/5
https://dune.github.io/computing-training-basics/03-data-management/index.html
https://github.com/DUNE/data-mgmt-ops/wiki/Using-Rucio-to-find-Protodune-files-at-CERN
https://github.com/DUNE/data-mgmt-ops/wiki/Rucio-for-Beginners

(Thanks Jake!)

How to read data (LArSoft/ROOT)

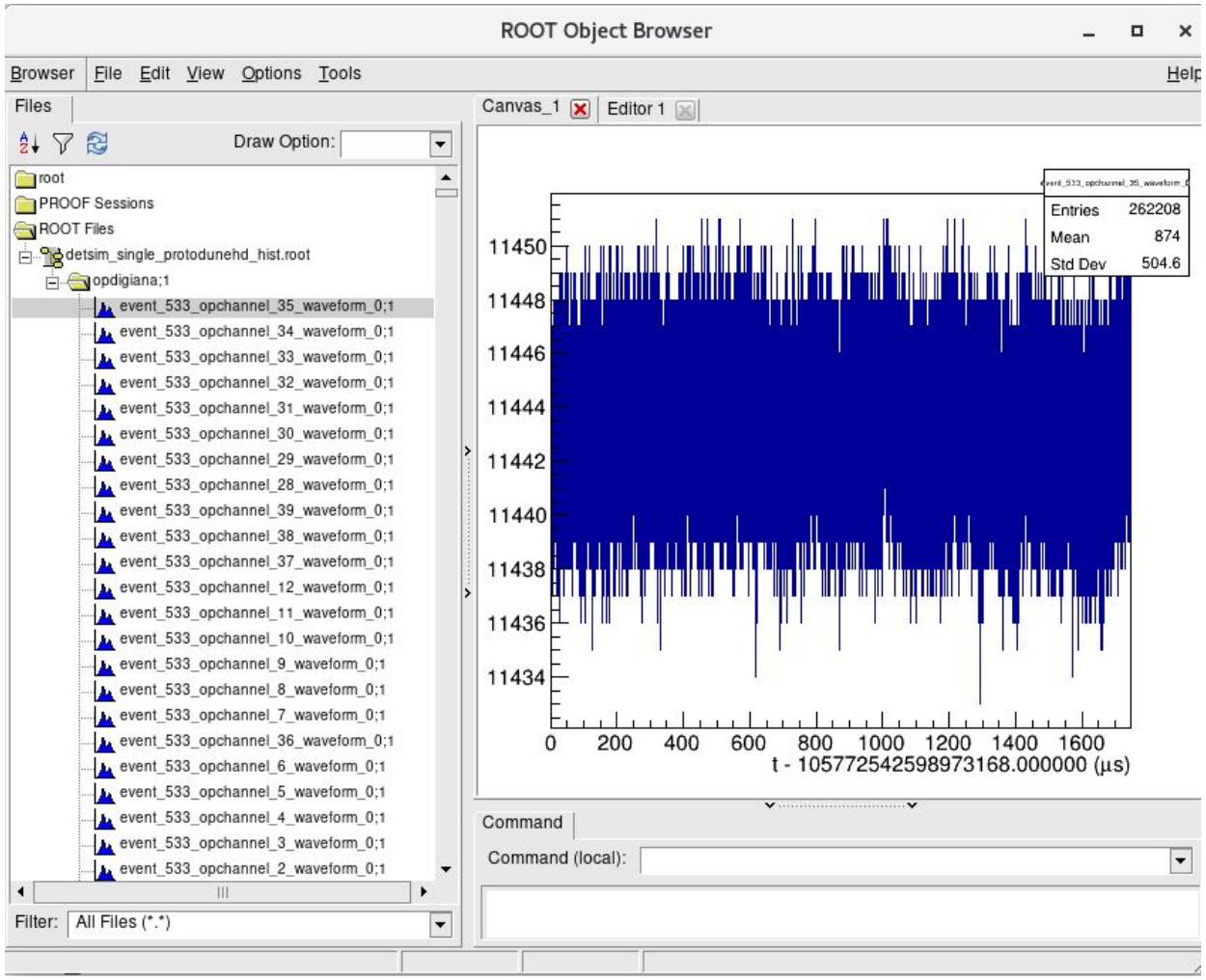
There is a LArSoft module that converts hdf5 into ART:

$ lar -c **run_pdhd_tpc_decoder.fcl**
/eos/experiment/neutplatform/protodune/rawdata/np04/hd-protodune/raw/2023/detector/test/None/00/02/26/29/np04_hd_run022629_0007_dataflow0_datawriter_0_20230818T121309.hdf5 -o example.root

It uses a temporary test channel mapping.

The ART file can be dumped it into ROOT histograms:

$ lar -c dump.fcl example.root -T hist.root (dump.fcl in the backup)



How to read data (python script)

There is a set of scripts created by Vitaly to read the PDS data directly from the HDF5 files.

To run it you need to set the DAQ software first:

```
$ source /cvmfs/dunedaq.opensciencegrid.org/products/setup
$ setup python
$ source /cvmfs/dunedaq.opensciencegrid.org/setup_dunedaq.sh
$ setup_dbt v7.5.0 OR setup_dbt latest
$ dbt-create -c fddaq-v4.2.1-c8 workarea # (it doesn't work in lxplus right now!, reported)
$ cd workarea
$ source dbt-env.sh
$ dbt-workarea-env
$ dbt-build
$ git clone https://github.com/DUNE-DAQ/dqmtools.git
$ cd dqmtools
$ pip install -r requirements.txt
```

Then use it with:

```
$ dqm_analyzer.py /data1/np04_hd_run022752_0000_dataflow0_datawriter_0_20230925T084543.hdf5
```

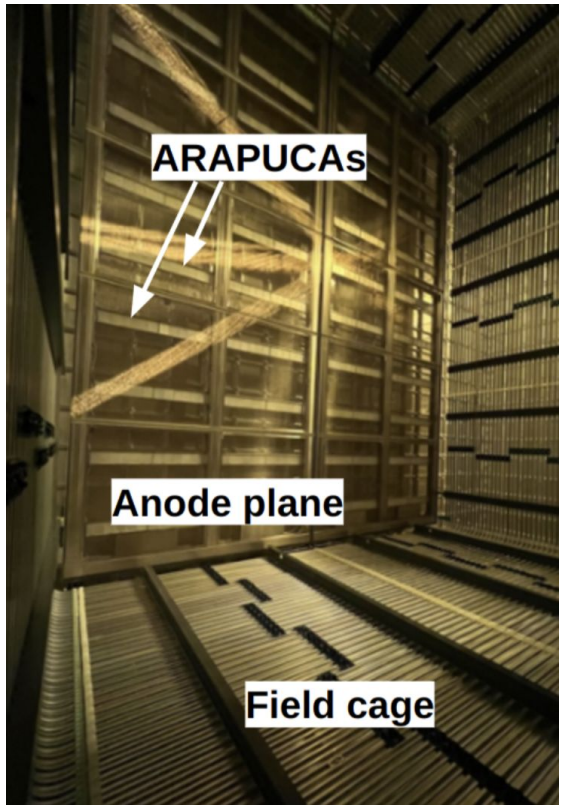
Useful links:

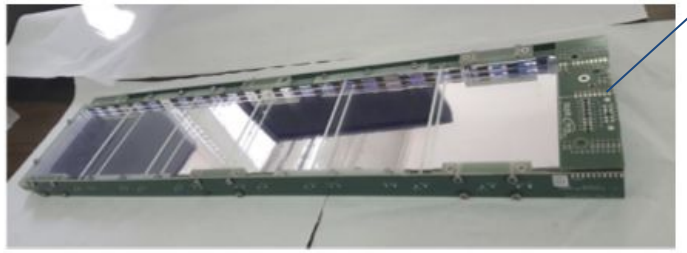
https://dune-daq-sw.readthedocs.io/en/v2.11.1/packages/daq-buildtools/#creating-a-work-area

https://github.com/DUNE-DAQ/dqmtools/tree/vipopov/waveform_pds_ana

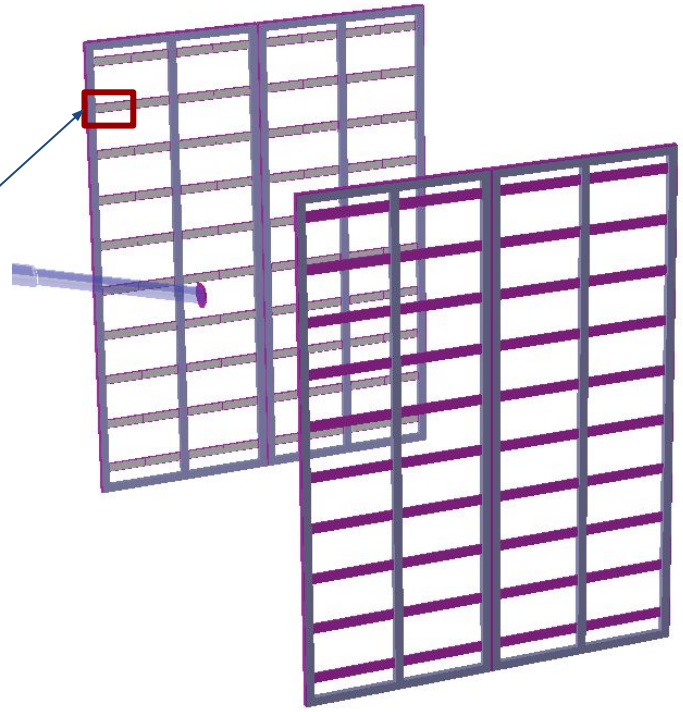
(Thanks Vitaly!)

ProtoDUNE-HD PDS





- X-Arapucas are integrated in the anode, behind the wire plane.
- 160 X-ARAPUCA supercells (40 x APA), that corresponds to 160 channels in the DAQ.
- 4 configurations are tested: 2 WLS bars and 2 SiPM models.
- https://wiki.dunescience.org/wiki/ProtoDUNE-HD_Geometry



LArSoft Channel mapping

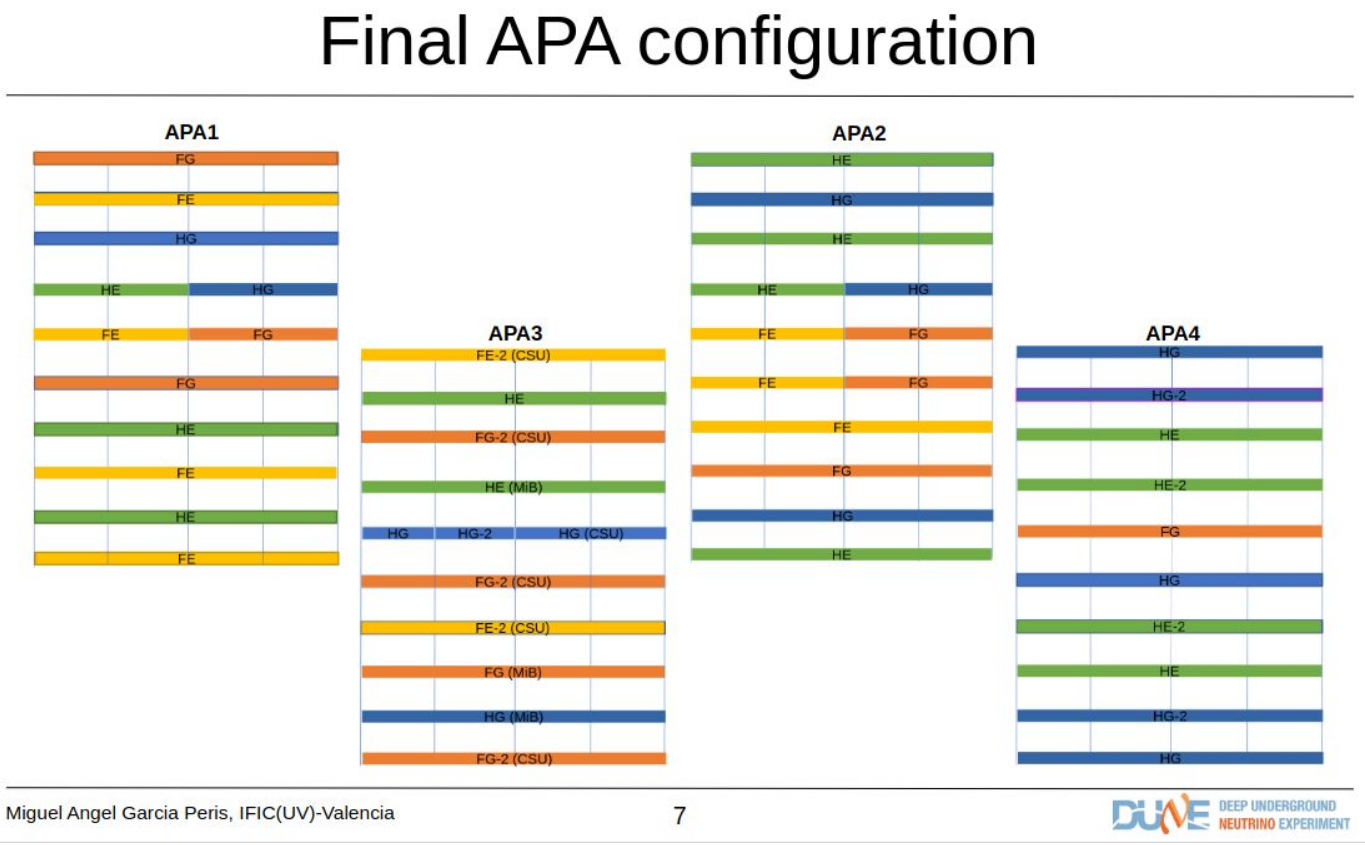
	APA 4				APA 3			
	70	60	50	40	30	20	10	0
	71	61	51	41	31	21	11	1
	72	62	52	42	32	22	12	2
X > 0	73	63	53	43	33	23	13	3
Beam Left	74	64	54	44	34	24	14	4
Jura Side	75	65	55	45	35	25	15	5
	76	66	56	46	36	26	16	6
	77	67	57	47	37	27	17	7
	78	68	58	48	38	28	18	8
	79	69	59	49	39	29	19	9

	APA 2				APA 1			
	150	140	130	120	110	100	90	80
	151	141	131	121	111	101	91	81
	152	142	132	122	112	102	92	82
	153	143	133	123	113	103	93	83
X < 0	154	144	134	124	114	104	94	84
Beam Right	155	145	135	125	115	105	95	85
Saleve Side	156	146	136	126	116	106	96	86
	157	147	137	127	117	107	97	87
	158	148	138	128	118	108	98	88
	159	149	139	129	119	109	99	89

(By Jake)

Hardware channel mapping

Miguel's:
https://indico.fnal.gov/event/53965/contributions/257977/attachments/163218/215959/2023_01_24_DUNE_cm_PDS_installation_PD2_migue.pdf



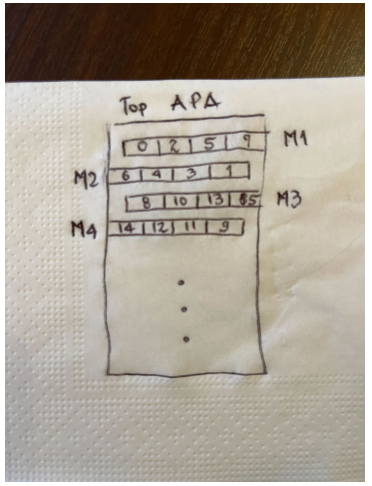
Hardware channel mapping

Daphne cabling is defined here:

Manuel's slides:
https://indico.fnal.gov/event/60987/contributions/283201/attachments/174542/236665/DAPHNE_NP04.pdf

Database):
https://docs.google.com/spreadsheets/d/1zzlUnk-Bd3HPtZjVei6dwMzZfJfmq4GauykcVqXREKw/edit#gid=1944195526

The WLS correspondence is not complete.



Channel numbering logic

		Saleve side			
		APA 1		APA 2	
		SiPM	WLS	SiPM	WLS
	M1	FBK	G	HPK	E
beam ->	M2	FBK	E	HPK	G
	M3	HPK	G	HPK	E
	M4	HPK	E/G	HPK	G/E
	M5	FBK	G/E	FBK	G/E
	M6	FBK	G	FBK	E/G
	M7	HPK	E	FBK	E
	M8	FBK	E	FBK	G
	M9	HPK	E	HPK	G
	M10	FBK	E	HPK	E

		Jura side			
		APA 3		APA 4	
		SiPM	WLS	SiPM	WLS
	M1	FBK		HPK	G
beam->	M2	HPK		HPK	
	M3	FBK		HPK	E
	M4	HPK		HPK	
	M5	HPK		FBK	G
	M6	FBK		HPK	G
	M7	FBK		HPK	
	M8	FBK	G	HPK	E
	M9	HPK	G	HPK	
	M10	FBK		HPK	G

Backup

Opdigiana example

```
#include "services_refactored_pdune.fcl"
#include "tools_dune.fcl"
process_name: Ana
services:
{
  TFileService: { fileName: "histograms_file.root" }
  TimeTracker:       @local::dune_time_tracker
  MemoryTracker:   @local::dune_memory_tracker
  RandomNumberGenerator: {} #ART native random number generator
  FileCatalogMetadata:  @local::art_file_catalog_mc
  @table::protodunehd_refactored_simulation_services
}
source:
{
  module_type: RootInput
  maxEvents: 30000
  fileNames: ["input.root"]
}
physics:
{
 analyzers:
 {
              opdigiana: {
              InputModule: "pdhddaphne"
              InstanceName: "daq"
              SSP_LED_AnaTree: false
              module_type: "OpDetDigiAnaDUNE"
              }
 }
 ana: [ opdigiana ]
 trigger_paths: [simulate]
 end_paths:          [ana]
}
```