DUNE CDR Review - ProtoDUNE-SP Lessons Learned

Karol Hennessy

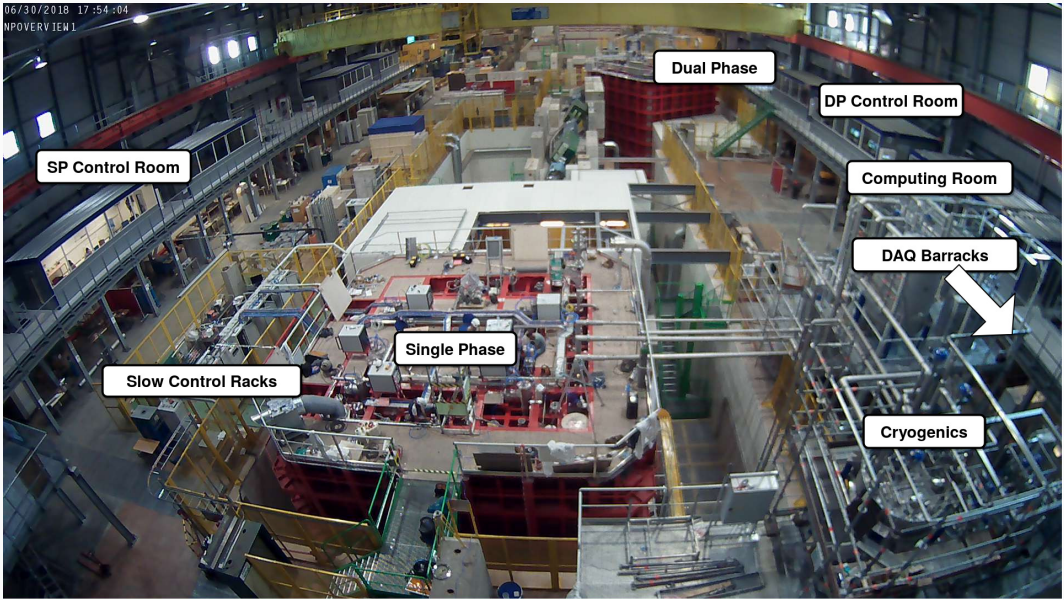
on behalf of ProtoDUNE-SP DAQ

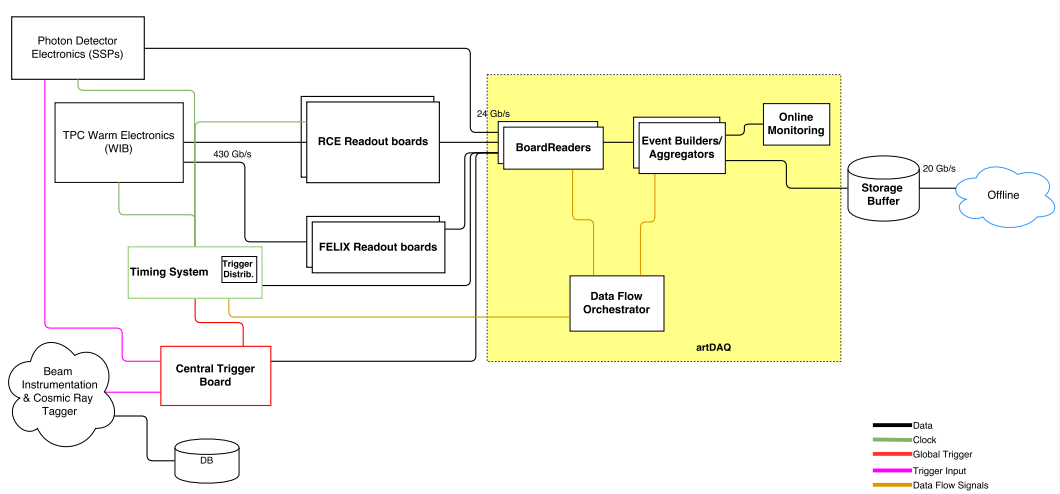
December 3, 2018

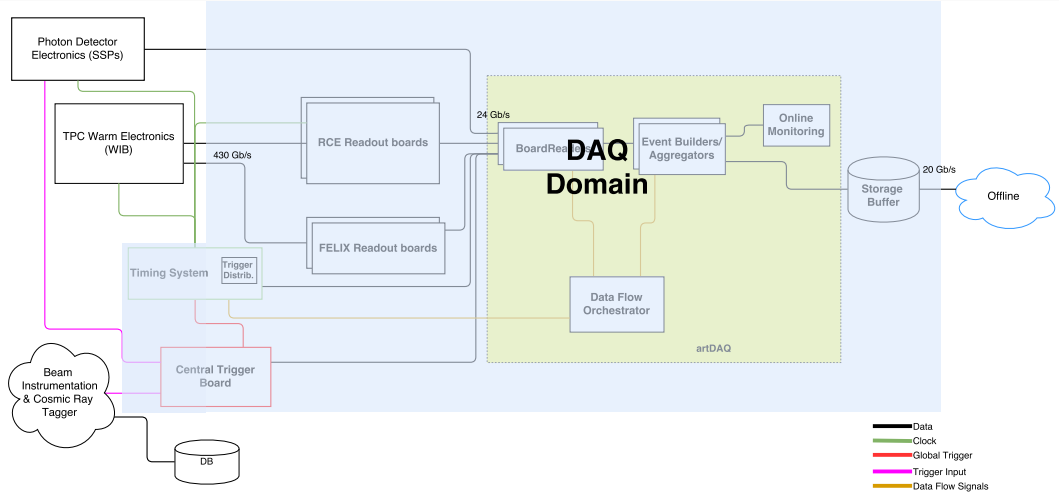
University of Liverpool

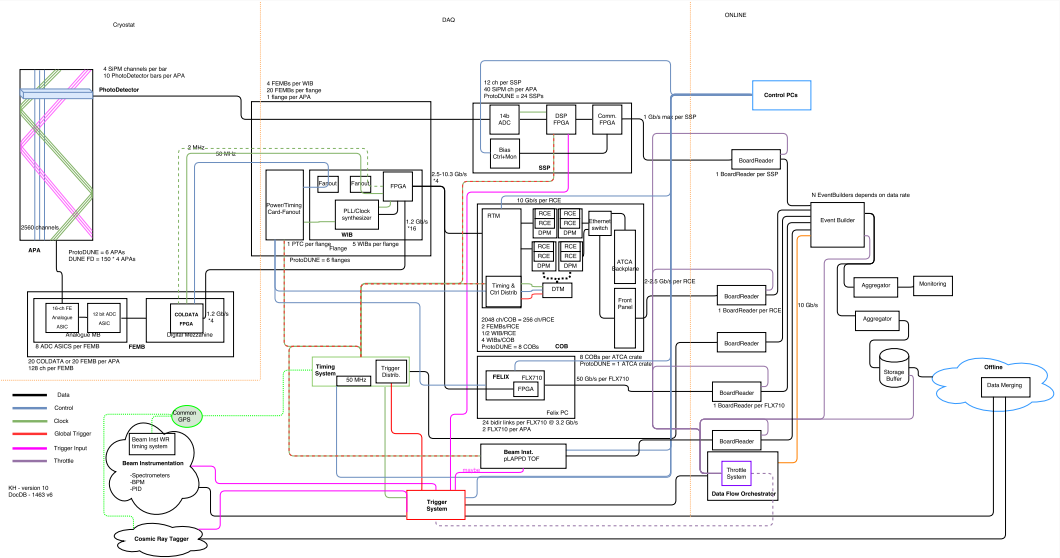
Intro - ProtoDUNE-SP

- Started Q3 2016 (in terms of DAQ)
- One of the two prototypes for the DUNE far detector
- running on beam in Q4 2018
- running with cosmics longer...
- The TPC wire planes and photon detectors are supported by beam instrumentation and a cosmic ray tagger.
- Externally triggered (no self triggers)
- No zero suppression
- At surface


06/30/2018 17:54:04
NPOVERVIEW1
Dual Phase
DP Control Room
SP Control Room
Computing Room
DAQ Barracks
Single Phase
Slow Control Racks
Cryogenics


Photon Detector Electronics (SSPs)
TPC Warm Electronics (WIB)
430 Gb/s
RCE Readout boards
FELIX Readout boards
24 Gb/s
BoardReaders
Event Builders/ Aggregators
Online Monitoring
Storage Buffer
20 Gb/s
Offline
Timing System
Trigger Distrib.
Data Flow Orchestrator
artDAQ
Central Trigger Board
Beam Instrumentation & Cosmic Ray Tagger
DB
Data
Clock
Global Trigger
Trigger Input
Data Flow Signals


Photon Detector Electronics (SSPs)
TPC Warm Electronics (WIB)
430 Gb/s
RCE Readout boards
FELIX Readout boards
24 Gb/s
BoardReader
DAQ Domain
Event Builders/ Aggregators
Online Monitoring
Storage Buffer
20 Gb/s
Offline
Timing System
Trigger Distrib.
Data Flow Orchestrator
artDAQ
Central Trigger Board
Beam Instrumentation & Cosmic Ray Tagger
DB
Data
Clock
Global Trigger
Trigger Input
Data Flow Signals


Cryostat
DAQ
ONLINE
4 SiPM channels per bar
10 PhotoDetector bars per APA
PhotoDetector
2560 channels
APA
ProtoDUNE = 6 APAs
DUNE FD = 150 * 4 APAs
2 MHz
50 MHz
4 FEMBs per WIB
20 FEMBs per flange
1 flange per APA
Fanout
Fanout
FPGA
PowerTiming Card-Fanout
PLL/Clock synthesizer
WIB
1.2 Gb/s *16
5 WIBs per flange
1 PTC per flange
Flange
ProtoDUNE = 6 flanges
2.5-10.3 Gb/s *4
12 ch per SSP
40 SiPM ch per APA
ProtoDUNE = 24 SSPs
14b ADC
DSP FPGA
Comm. FPGA
Bias Ctrl+Mon
SSP
1 Gb/s max per SSP
Control PCs
BoardReader
1 BoardReader per SSP
N EventBuilders depends on data rate
Event Builder
10 Gb/s per RCE
RTM
RCE
DPM
Ethernet switch
ATCA Backplane
Timing & Ctrl Distrib
DTM
Front Panel
2-2.5 Gb/s per RCE
BoardReader
1 BoardReader per RCE
10 Gb/s
Aggregator
Monitoring
Aggregator
Storage Buffer
Offline
Data Merging
16-ch FE Analogue ASIC
12 bit ADC ASIC
Analogue MB
COLDATA FPGA
Digital MB/Cable Interface
1.2 Gb/s *4
8 ADC ASICS per FEMB
FEMB
20 COLDATA or 20 FEMB per APA
128 ch per FEMB
2048 ch/COB = 256 ch/RCE
2 FEMBs/RCE
1/2 WIB/RCE
4 WIBs/COB
ProtoDUNE = 8 COBs
COB
8 COBs per ATCA crate
ProtoDUNE = 1 ATCA crate
50 Gb/s per FLX710
Timing System
50 MHz
Trigger Distrib.
FELIX
FLX710
FPGA
Felix PC
24 bidir links per FLX710 @ 3.2 Gb/s
2 FLX710 per APA
BoardReader
1 BoardReader per FLX710
Beam Inst. µLAPPD TOF
Throttle System
Data Flow Orchestrator
Trigger System
Data
Control
Clock
Global Trigger
Trigger Input
Throttle
Common GPS
Beam Inst WR timing system
Beam Instrumentation
-Spectrometers
-SPM
-PID
Cosmic Ray Tagger
KH - version 10
DocDB - 1463 v6

General comments

- We got some things right! (It worked after all...)
- Possibly the most important was the delineation of interfaces
  - We made clear early on exactly where DAQ starts and ends and what are the interfaces to FE electronics and offline.
    - but we failed on some of the internal interfaces
- We weren't too ambitious; time was against us
  - We used off the shelf solutions where possible
  - We leveraged CERN IT substantially (and many thanks to them for their help and support)
- We had "milestone weeks" in the early days
  - Travel is expensive
  - BUT having people in the same room for significant periods is essential
  - Huge progress was made in these weeks
- We had a great team! (And we had fun)

Timing

- We “lacked” the LHC clock !
  - That meant everyone and their grandmother arrived with a different clock frequency (50 MHz, 150 MHz, 66 MHz...)
  - PICK A FREQUENCY !
  - We had two semi-independent systems - the Bristol Timing Board for NP04 and White Rabbit for Beam instrumentation
- We used too much copper
  - Way too time wasted with stupid HDMI converter boxes and special cables $\Rightarrow$ Oscilloscope debugging
    - One of them didn’t work in the end (we had redundancy thankfully)
  - Just use fibre
- Too busy core firmware developer
  - who then became very important to DUNE DAQ :-|
  - Any firmware this crucial needs more than one core developer
  - Should have a bus factor > 1

Timing 2

- Integration needs to be done early and regularly (new firmwares etc)
  - tied to availability of hardware and experts
- Calibration systems must connect to the timing too! (done but too late)
- Suggest a mini-DAQ system deployable to electronics sites

Trigger

- Tighter coupling with timing
  - Trigger HW doesn't know which fragments get vetoed by timing system
- Monitoring Metrics - essential for trigger to determine rates
  - Inherent problem - they don't really know when to evaluate (i.e. once every 2 secs is not the same as once every N events)
  - More permanent metrics - stats per run, storage etc. Should be automatic.
- Trigger definitions
  - The firmware was very versatile but needed more feedback from offline (physics,computing...)
- Beam instrumentation/Online
  - Need more commissioning time for the trigger - it's very hard to do anything without real hardware and/or beam irrespective of planning.

TPC Readout - RCE

- Full working slice essential
- Simulation of firmware not good enough (particularly for compression)
  - preferred to have hardware in own lab
  - remote possible but time difference is an issue
- Some RCEs failed regularly - but hard to disassociate from problems upstream
- “Reboot” is not the most efficient recovery method
- Need to expose more “expert info” to the run control - unclear which parts failed when it failed
  - at data input? compression/processing? output buffering?
- Similarly for DQM - need informative error banks being monitored

TPC Readout - FELIX

- First fully working version was pretty late
- Need project specific firmware experts (currently borrowed from ATLAS)
- Documentation and better monitoring solutions
- Similar to RCE
  - difficult to disassociate from problems upstream
  - Need to expose more “expert info” to the run control
- Problems were difficult to debug for non-experts

artDAQ

- Overall, artDAQ felt a little too monolithic
  - concepts of HW config and readout are mixed up - suggest to split the two
  - e.g., Dispatcher did not work but hard to determine if it was dispatcher proper, upstream data source, or art interface
- Fault Tolerance
  - One crash = all stop
  - Need to be able to bring processes up and down without killing the running system
- Configuration book-keeping
  - too simple - easy to clobber “good configurations” from say Photons with a new Trigger config
  - needs splitting up per sub-config; more steps

artDAQ 2

- Need better monitoring - both host level and in artDAQ proper
  - Hardware issues were often confused with artDAQ - implement ganglia etc.
- Separate packages that can be released independently would be preferable to “one big release”
- Achieved perfomance quite late and often by scaling horizontally (16 EBs instead of 4)
- We need a BoardReader starter training. This is beyond just artDAQ (because there are examples!) but also “good practices”. Like, your data format should always have a version number...

Run Control

- Need a development environment; most code was developed on the production system
- Alarms were rudimentary - but they're essential for shifts! (Significant task)
- Need communication between slow control and DAQ
  - Shifters copying info about slow control to a checklist is a recipe for mistakes.
  - Should have been put in place but constantly sidelined for other priorities. Main roadblock was CERN network isolation.
- Scale is a possible issue - WinCC can be a resource hog
  - Is WinCC the right tool for RC?

Run Control 2

- Resource management/Partitioning was messy with special cases - e.g., trigger
  - trigger was in Partition 1 but trigger primitives (PDS) were in another partition
- Metric monitoring
  - overall WinCC viewer is not great (neither for viewing, nor extracting data)
  - rates were not reported accurately due to averaging - needed 2 Hz fake triggers to avoid false alarms
- Needs more user friendliness

Online Monitoring

- Monet (OM webviewer) was too slow for LArTPC datasets. Eventually it performed much better but hard to say it's good enough for DUNE scales.
  - Not clear if slowness is due to the "web" part of the viewer, but the on-demand server-side image conversions don't help
  - Need a clear evaluation of any future tool with MC
- OM needs to be more than "data viewing" - it should know about slow control and run control states, and beam info
- OM was restricted to files for data transfer so we didn't learn much about the performance of the software itself, being limited by reading files instead of the shared memory

Online Monitoring 2

- students coped quite well with the maintenance and implementing improvements,
  - but a more stable group would have made things easier; a closer collaboration with the nearline developers would have been preferable, but the version dependency clashes made it quite difficult
- Some users found the OM based on LarSoft very tricky to maintain and inefficient - some fresh ideas are needed here.
- We begged many times for people on this - best effort doesn't work. It needs official status and groups responsible.

Misc

- Events don't come in order - don't expect it. Reason: many event builders in parallel
- Define interfaces as well as possible early - you shouldn't change the interface and if you have to there better be a good reason
- Availability of experts - often spotty. On site subsystem experts are essential.
- The connection between the hardware and the analysis was not the best. This affects both the practical aspects of data formatting, and even of people understanding what they are looking at.
- Metadata is important to offline analyses - needs input
- Timezones are an issue - the milestone weeks were a good way to progress

Integration/Installation

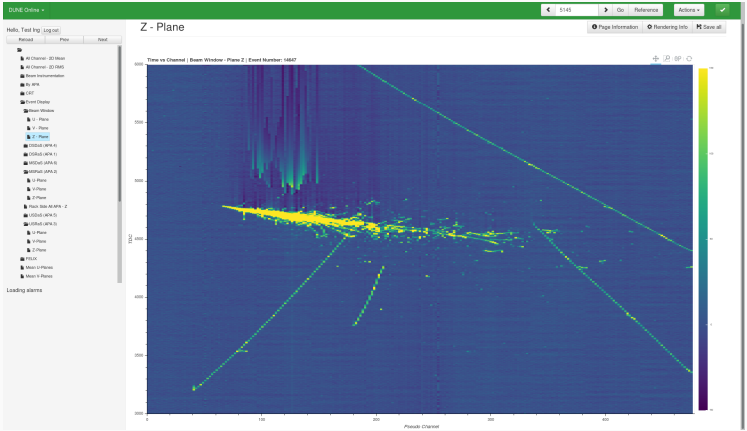
- Sysadmins - “you need professionals” -from the mouth of *our* sysadmin Geoff.
- Remote access - there needs to be a standard solution (be it RDP or whatever)
  - problems from remote - viewing live logs; RC; metrics
- Scale is the real issue here - homebrew NFS servers won’t cut it for DUNE
- Consistency makes life easier - reduce special cases/setups to a minimum
- Use Ansible - automation tool
- We never really got the right mix of “special users” vs. permissions - the generic user “np04daq” can effectively “rm *”
  - Nothing really bad happened and we did have a working back up!
  - Need proper group permissions

Integration/Installation 2

- Assume there will be too much dust. And assume there's going to be a leak somewhere above your favourite electronics.
- Label everything early and leave the spaghetti to the chef. Doesn't matter if it's only going to be there for one day. (also you're wrong, it'll be there forever). Double check the labelling.
- Cold electronics required their own DAQ - independent from the experiment DAQ. They also required DAQ services - disk space, analysis CPU. This was unexpected.
- Better Control room tools - some were selected based on urgency not quality (logbook)

Final Remarks

- I'd like to re-iterate my starting point that ProtoDUNE-SP/NP04 was a great success.



Final Remarks

- I'd like to re-iterate my starting point that ProtoDUNE-SP/NP04 was a great success.
  - High uptime - runs up to 8 hours
- ProtoDUNE-SP had a great team and that success is due to their hard work and dedication.
- These lessons are useful only to a point
- For DUNE to be a success we should keep as much of the knowledge gained by that NP04 team as possible.
- Reminder - We have to do it all again in 2021 when the beam comes back
  - and we have to do it with dual-phase too

Backup

Firmware

- Forward progress was made when firmware developers were actively engaged
  - Again interfaces - a clear interface between software and firmware should be defined (this tended to be better when the dev wrote the firmware and the software)
  - Clear register map
  - Clear API
- Changing firmware over network is essential
- If we need a “blaster cable” - we need USB, it quickly becomes a lot of USB connections
- Don’t have “special firmware” for calibration/testing. Have one firmware do everything and turn features on/off as you need them. Buy a bigger chip if necessary.

Misc

- No salvage! Don't be cheap! If someone says you should re-use hardware X from the past, you're going to have a bad time.
- Spares, tools - have a proper inventory and have people check things out.
- All SFPs are not created equally - there are 1G SFPs, 10G SFPs, 40G QSFPs, timing SFPs of various wavelengths and colours.
- A tighter link between offline (DQM and reco) and online monitoring would be beneficial to both.

Offline

- Data formats should be versioned, always. (should be in the data header)
- Earlier and more complete design for handling of error conditions. These are visible offline as corrupted or truncated data.
- “best practices” document.
- Data challenges essential test of online and offline
- Slow controls info needed offline
- Good runs list was extremely useful. Must be done live. Hard to determine later.
- All code should be committed so not lost after NP04
- More “build” machines, general web service machines
- More offline slices

Cold Electronics/Photon Detectors

- Unified firmware - for both calibration and DAQ
- Need to allocate enough resources for “not just DAQ”
- More debugging tools
- More extensive alarms
- Infrastructure for automated scans - scans were done by hand in individual runs
- An extra configuration layer on top of low level firmware register accesses