

ProtoDUNE SP DAQ Overview

Edited by G. Lehmann Miotto (CERN EP/DT-DI)

1 Introduction

The design of the ProtoDUNE SP DAQ relies on the main principles of the DUNE DAQ design and on experience gained while taking data at the 35 t prototype. It is nevertheless also tailored for operation at a test-beam (cyclic spill structure, external systems to provide a trigger system). The overview of the DAQ system is briefly described in the ProtoDUNE TDR [1]. In this document we will add information regarding the external interfaces of the system, as well as describing in more detail how the DAQ components interact with each other.

The main performance requirements for the DAQ system are summarised in Table 1 and further described in the TDR.

Table 1 Main requirements for the ProtoDUNE DAQ.

Parameter	Value
Trigger rate	25 Hz
Spill duration	4.8 s
SPS Cycle	22.5 s
Readout time window	5 ms
# of APAs to be read out	6
Single readout size (per trigger)	230.4 MB
Lossless compression factor	4
Instantaneous data rate (in-spill)	1440 MB s ⁻¹
Average data rate	576 MB s ⁻¹
3-Day buffer depth	300 TB

2 External Interfaces

The main external interfaces of the DAQ system are depicted in Figure 1.

The DAQ receives a Spill signal from the SPS [2][3], in order to know that a particle burst is about to arrive.

In addition the DAQ receives input signals from the Beam Instrumentation, from the Cosmic Rays Tagger (CRT) and possibly from the Photon Detectors (via the Silicon Photomultiplier Signal Processor (SSP) boards) that allow the system to form a trigger decision.

The DAQ distributes timing and trigger information to the Time Projection Chamber (TPC), in particular to the Warm Interface Boards (WIBs) and to the Photon Detectors, in particular to the SSPs).

The DAQ collects data from the WIBs and SSPs, builds complete events and stores them into data files.

Those data files together with the required metadata are sent to the Offline Computing (EOS) using the protocols defined by the latter. In order to provide sufficient decoupling between the

resources of the CERN computer centre (EOS) and the DAQ system a temporary storage worth 3 days of raw data will be installed on the DAQ side.

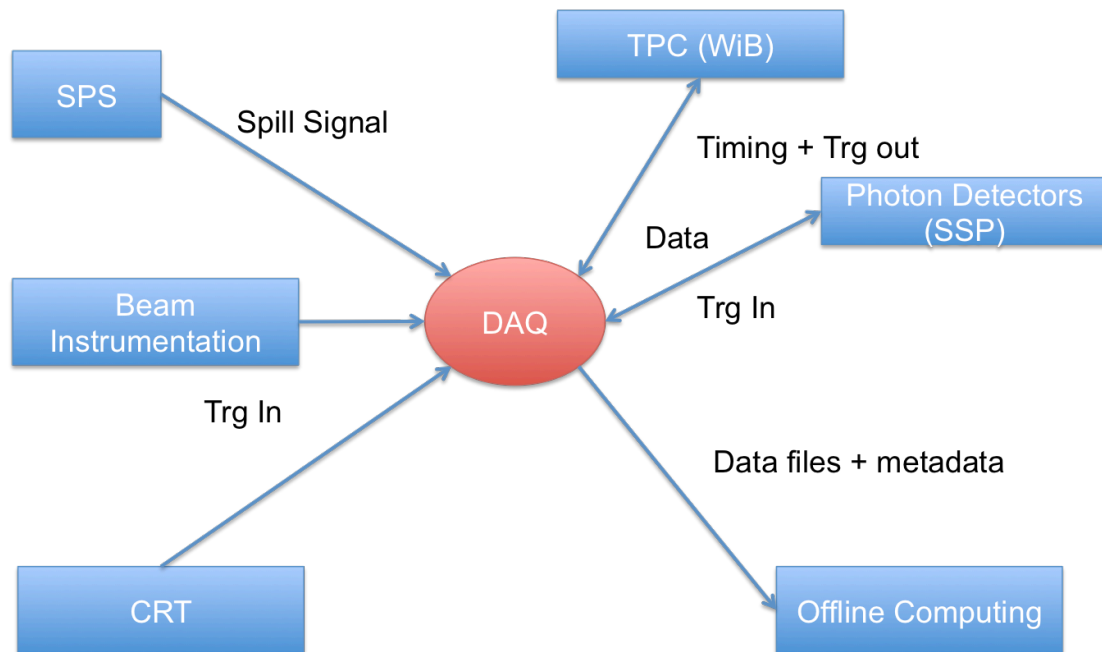


Figure 1: The external systems interfacing with the DAQ system.

Figure 2 Shows another important view of the interfaces: the physical connections between DAQ elements and external system with respect to their grounding. ProtoDUNE SP has chosen to isolate the detector and therefore its ground will potentially differ from the building's ground level. In order to avoid any ground loops, all connections between the DAQ and the systems that will be on detector ground are going to be optical.

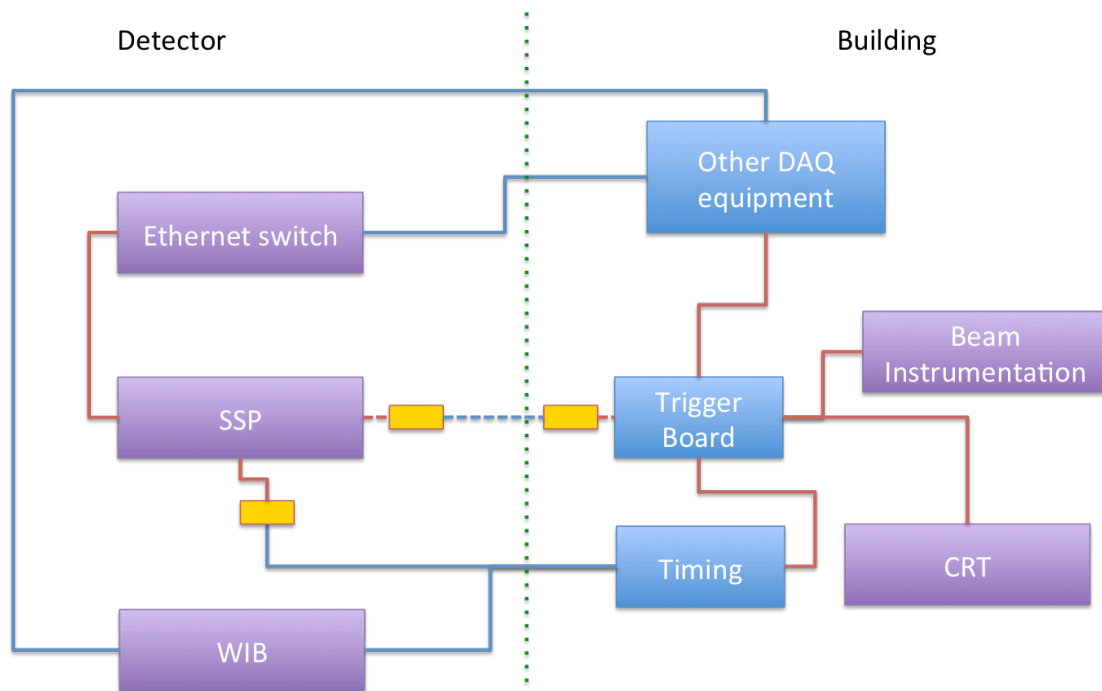


Figure 2 View of the connections between components on detector ground and the DAQ elements of building ground. Blue lines are optical and red lines are copper connections. All lines crossing the dotted vertical line are optical, in order to avoid any ground loops.

3 The DAQ Architecture

Figure 3 shows a high level view of the DAQ architecture for ProtoDUNE.

The Beam Instrumentation, CRT and Photon Detectors provide input signal to a trigger board that combines them, applies simple (no overlapping readout windows for triggers) and complex dead-time logics (e.g. not more than N events in M seconds). The trigger decision is distributed via the timing distribution system. The target trigger rate for ProtoDUNE is of 25 Hz during the SPS spills (making use of the inter-spill time to complete the acquisition).

The TPC sends data continuously to the readout system: 15360 channels are sampled at a rate of 2 MHz, causing a continuous flow of data at ~430 Gb/s. The Photon Detectors process data on detector and stream out self triggered data summaries (headers) as well as short waveform data when an external trigger arrives.

Upon distribution of a positive trigger decision, the readout system gathers all time slices corresponding to a readout window of 5 ms around the time of the trigger, compresses the data (for the TPC) and sends them to the Event Building farm.

Once the Event Builder has completed forming a complete event, data are sent for aggregation and storage into files. The estimated event size (assuming a factor x4 compression at the readout system) will be of ~60 MB.

In parallel to the main DAQ, also the CRT and Beam Instrumentation will record information, usable for offline analysis. The association of the information from these three data sources will be done using a common timestamp, since all systems will take their clock from a common GPS source.

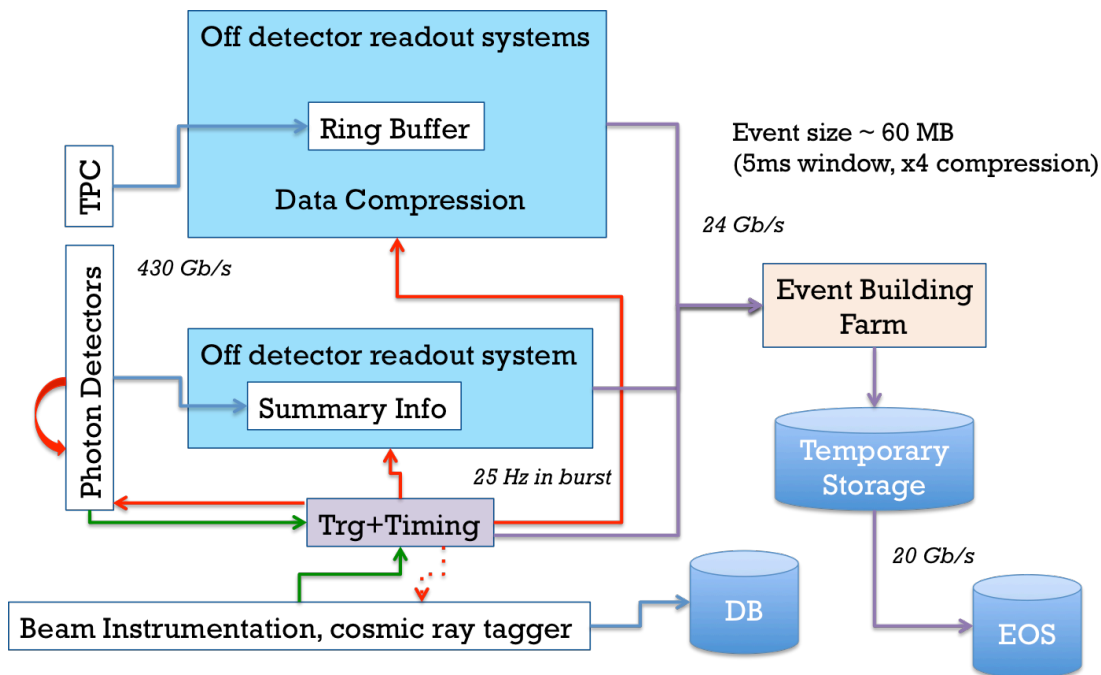


Figure 3 High level view of the DAQ architecture.

Trigger and Timing

The timing and trigger systems will be described in detail in the talk by Dave Newbold and its supporting documentation.

Readout Systems

There will be three flavours of off-detector readout systems in ProtoDUNE:

- software only: the readout systems for Photon detectors as well as for the Trigger and Timing system are fully software based. Their internal logics will be described in the respective talks (D. Newbold, M. Haigh)
- Software + dedicated hardware: for the TPC readout two solutions are being implemented, to be evaluated and compared; the ATCA based SLAC solution (RCE) and the PCIe based FELIX solution. These two systems will be described by M. Graham and F. Filthaut respectively.

For all readout systems the software application that interfaces to the event building farm is the so-called Board Reader. As the name suggests, this application type is tailored to read data from specific detector electronics, prepare event fragments and send them to the event building system.

Data Flow

The data flow software is based on the artDAQ **Error! Reference source not found.** framework and will be described in detail in the talk by K. Biery. The underlying hardware will be fully based on COTS servers running linux CentOS7 and on a 10 Gb/s network (single device). Overall, 10-15 servers will be sufficient to implement the complete dataflow for protoDUNE (board readers and data flow orchestrator, event builder, storage).

Figure 4 shows a simplified sketch of the data flow for a single event. This design is not fully implemented in artDAQ yet, but its salient aspects are already supported and the software will be upgraded well before installation of the DAQ in EHN1.

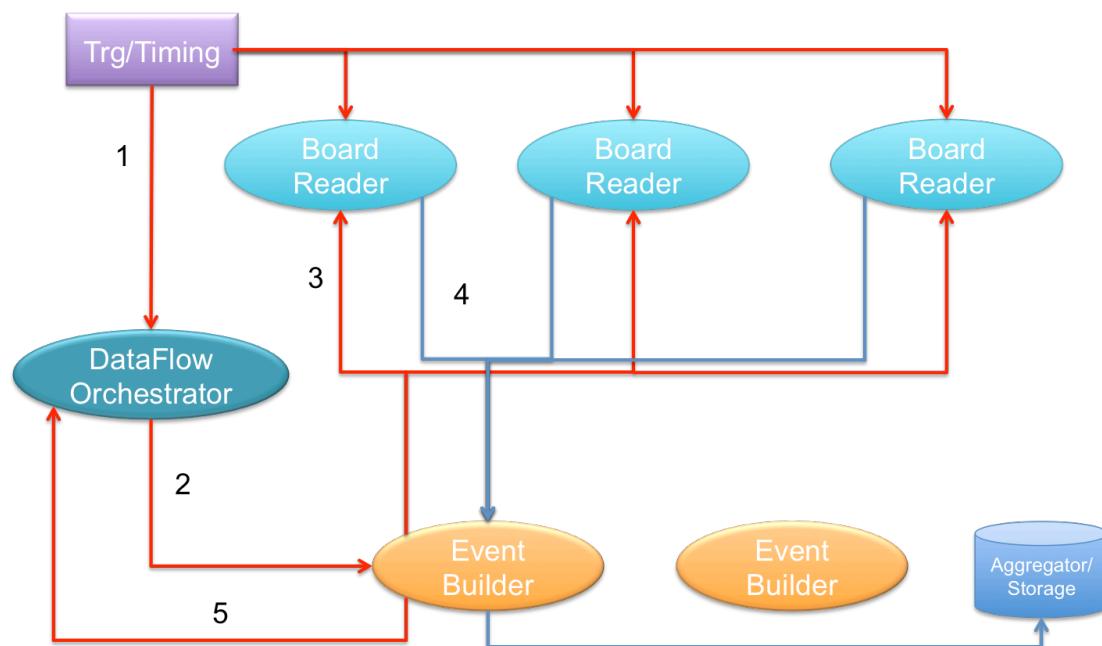


Figure 4 A simple diagram illustrating the flow of data for a single event. 1) A trigger is fired and the trigger information (timestamp) is distributed to the Board Readers and the DFO. 2) The DFO generates a sequence number and assigns the event to an EB node by sending a message to it. 3) The EB requests data from the Board Readers. 4) The Board Readers respond with the data associated to the timestamp of the trigger. The event builder sends the event to the aggregator for storage. 5) The EB node notifies to the DFO that the event has been completely built.

Backpressure

The DAQ system is dimensioned to be able to sustain the requested rate of 25 Hz during an SPS spill and is capable of making use of the inter-spill time to absorb any backlog that may have built up during the spill.

Four different sources of backpressure have been identified for protoDUNE. In this section they are described, together with an indication of the time scales at which every type of backpressure builds up.

Backpressure is not expected at the level of the on/near detector readout electronics, since the system has been dimensioned such that for the TPC (and for the SSP), data will be read-out continuously (based on self triggering).

1) Trigger rate spikes

Triggering with beam is subject to variations in rate that may cause for an over-subscription of DAQ resources. In order to prevent this from happening, artificial dead-time will be managed inside the trigger board logics. As an example, simple dead-time logics will be introduced, excluding the firing of the trigger at an interval closer than a defined fraction of the readout window (~5 ms); additional complex dead-time logic in the trigger board may forbid accepting more than a fixed number of triggers during a time interval (e.g. max 200 triggers during spill, max 50 triggers during inter-spill; at 25Hz over a 5s spill, 125 events are expected on average per spill).

An example where the oversubscription of DAQ resources may occur is at the buffering in the RCE which is used to accept data mainly during the spill and push it downstream continuously (de-randomised) more slowly using also the inter-spill gap). A similar de-randomisation happens in the board readers for the SSPs.

Trigger rate spikes will thus be smoothed out directly by the trigger and do not need any feedback to the trigger from other systems.

2) Transient noise bursts

Transient noise bursts may blow-up the size of data corresponding to a trigger. The limited internal bandwidth in the RCE based readout system may cause a backlog of data to build up. In order to prevent data loss, a BUSY software message may be sent towards the trigger, when buffer occupancy reaches a certain threshold. When reaching a comfortable buffer occupancy, the BUSY shall be cleared.

This type of problem builds-up progressively, on a timescale of hundreds of milliseconds or even seconds.

When the SSPs are running in self-triggered mode, the hits will be passed to the board readers for packaging as event fragments. The SSP hit-thresholds and waveform-length settings must be set to not normally overload the Ethernet links from the SSP to the board readers. There is the possibility of a 'FIFO-full' condition if there is an exceptionally large number of hits due to a fluctuation. This exceptional condition must be managed from within the SSP, preferably by dropping hits and by having a status bit to indicate that some hits have been dropped (to be read once at the end of each spill). When the SSPs are running in triggered mode, (or in a superposition mode, where triggered hits are given a large waveform window, while self-triggered hits are just headers), the trigger rate limits proposed in section 1) and the waveform-length configuration settings should be used to remove the possibility of overwhelming the SSP-board reader link with data.

It shall be noted that a backpressure mechanism shall only be activated for exceptional transient noise bursts. In general, the trigger rate shall be adapted to the detector conditions and shall not cause any regular backpressure to be generated by the readout elements.

3) Event building/aggregation limitations

Temporary failure of one or more dataflow components (a network link, an event builder node/process, an aggregator, a pool of disks, etc...) may cause the data acquisition performance to be degraded. This will be noticed by the data flow aggregator (lacking EB nodes to assign event to). Similarly to the previous case, a BUSY message shall be issued

once queue lengths exceed a defined value. When reaching a comfortable situation, the BUSY shall be cleared.

This type of problem builds-up progressively, on a timescale of hundreds of milliseconds or even seconds.

A backpressure mechanism should only be used for transient issues of this type that may be rapidly fixed. Any long-term performance degradation of the DAQ shall be followed up with a reconfiguration of the system, defining a sustainable trigger rate.

4) Storage limitations

The DAQ writes raw data to a temporary storage area, waiting for the data to be transferred to EOS. A long-term outage of the connection to EOS or its performance degradation may cause the storage area to progressively fill up. This type of issue builds up in hours/days. No transient BUSY mechanism can be applied in this case.

From the analysis of back-pressure sources it is apparent that in ProtoDUNE there is no requirement for a fast feed-back to the trigger system to stop the generation of triggers. Therefore a software based messaging system is well suited for throttling the trigger. Nevertheless, provisions are made at the timing master level to be able to introduce a hardware signal indicating busy conditions, in case it turned out to be needed.

Control, Configuration, Monitoring

The control, configuration and monitoring systems are described in the talk by W. Ketchum.

Hardware Layout

The DAQ for protoDUNE is a fairly small system. It is expected that it will be completely implemented within 3 standard 19 inch racks. Figure 5 shows a draft layout of the DAQ racks.

4 Summary and Outlook

A complete design of the DAQ system exists:

- The external interfaces of the DAQ have been defined.
- Interactions between the DAQ and its neighbouring systems have been understood.
- The DAQ has been organized into components with well defined functionality as well as institutes taking the responsibility for delivering them.
- The flow of data and signalling between DAQ components have been designed.
- All hardware components required for the DAQ have been identified
 - COTs servers, switches, interconnects (not a large system!)
 - COTs storage solution
 - ATCA based SLAC solution
 - FELIX PCIe cards
 - Timing units
 - Central Trigger board

- We are confident that the proposed DAQ design can be implemented on the timescales required by ProtoDUNE, that it satisfies the requirements that have been put forward by the experiment and that it has sufficient modularity to allow for flexibility, in case of additional performance needs.

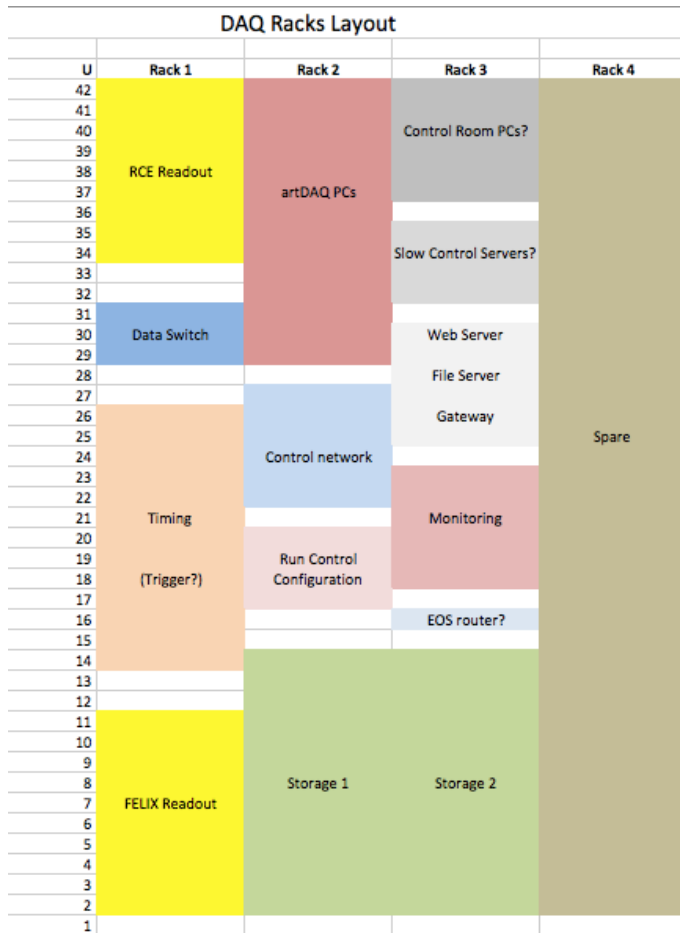


Figure 5 Draft rack layout for the ProtoDUNE DAQ system.

5 References

- [1] ProtoDUNE TDR,
<http://docs.dunescience.org:8080/cgi-bin/ShowDocument?docid=1794>
- [2] <https://home.cern/about/accelerators/super-proton-synchrotron>
- [3] <http://sba.web.cern.ch/sba/Documentations/How2controlNAbeams.htm>
- [4] Biery, K. et al. artdaq: An Event-Building, Filtering, and Processing Framework. *IEEE TNS* **60** 3764-3711 (2013)