Mu2e slow control and online DAQ interface development

Antonio Gioiosa
Università di Pisa, INFN Pisa

New Perspectives 2021
August 19, 2021
The signal we are looking for is a delayed monoenergetic electron with an energy of just under 105 MeV (muon mass)

The Mu2e Experiment at Fermilab

![Diagram of Mu2e experiment setup]

- Production Solenoid
- Transport Solenoid
- Detector Solenoid
- Production Target
- Collimator
- Tracker
- Calorimeter

![Graph showing proton pulse and signal]

- POT pulse
- $\pi^-$ arrival/decay time ($\times 1M$)
- $\mu^-$ arrival time ($\times 400$)
- $\mu^-$ decay/capture time ($\times 400$)

Live Window

Prompt background
Mu2e TDAQ and Slow Control integration

Summary:

• Mu2e TDAQ components Diagram
• Mu2e Timing Distribution
• Mu2e TDAQ Readout scheme
• Online DAQ (otsdaq) overview
• Slow control and its integration in otsdaq
  – Monitoring and Slow Controls GUI
  – Slow Controls Integration with otsdaq State Machine and Alarm handling
• Conclusions
Mu2e TDAQ components Diagram
Mu2e Timing Distribution

Requirement is to process 200K events/s

- Mu2e Runs are broken up into contiguous Event Windows
- Experiment defined Run Plan is coordinated by the Command Fan-Out Card (CFO)
- The System Clock (40MHz) and Event Window markers originate at the CFO ...and are distributed to ROCs:
  1. CFO distributes System Clock and Event Windows to DTCs with fixed latency
  2. DTCs distribute System Clock and Event Windows to ROCs with fixed latency
  3. ROCs respond to Data Requests

\[\text{DTC} \leftrightarrow \text{ROC Heartbeat packet (16 bytes) to specify the detail of each Event Window}\]
TDAQ Readout scheme

• 396 ROCs 69 DTCs (Kintex-7) for data readout and event building
• Large front end buffers to average over long off-spill time
• 800 threads on 40 nodes for HLT → ~5 ms per event
• ~40 GB/s data read out to storage decision layer, ~280 MB/s written to disk

High Level Trigger Software
The Test Stand
Mu2e Online DAQ solution: otsdaq

**otsdaq overview**

Acronym for “off-the-shelf data acquisition.”

- **otsdaq** is a Ready-to-Use data-acquisition (DAQ) solution aimed at test-beam, detector development, and other rapid-deployment scenarios
- It uses the **artdaq** DAQ framework under-the-hood, providing flexibility and scalability to meet evolving DAQ needs
- **otsdaq** provides a library of supported front-end boards and firmware modules which implement a custom UDP protocol
- Developments are in two directions: server side and web side.
- An integrated Run Control GUI and readout software are provided, preconfigured to communicate with **otsdaq** firmware
**otsdaq overview**

More info at **otsdaq** web page [https://otsdaq.fnal.gov/](https://otsdaq.fnal.gov/)

**otsdaq** is a Ready-to-Use data-acquisition (DAQ) solution aimed at test-beam, detector development, and other rapid-deployment scenarios. **otsdaq** uses the **ardaq** DAQ framework under-the-hood, providing flexibility and scalability to meet evolving DAQ needs. **otsdaq** provides a library of supported front-end boards and firmware modules which implement a custom UDP protocol. Additionally, an integrated Run Control GUI and readout software are provided, preconfigured to communicate with **otsdaq** firmware.
otsdaq overview

Data Flow Block Diagram

Server side is C++. User code is added through plugins (C++ classes inheriting from the appropriate class)

Web side is HTML and JavaScript. User code is added in the form of web-apps through .html files (including the appropriate .js and .css files)
Data processing: Data Quality monitor GUI example

- Mu2e’s event window data will be processed through artdaq modules

- Data processor and Data Quality Monitor DQM plugins are provided by otsdaq core

- **DQM** generates data products that are sent to an artdaq Dispatcher, which aggregates DQM metrics and presents them to a visualizer application
Slow Controls connection and EPICS plugin development in otsdaq

Experimental Physics and Industrial Control System

Each arrow on this diagram is a network connection.
Channel subscription to EPICS uses Input Output Controller (IOC)

• integration of slow control in the online daq uses the same Interface plugin for:
  a. Monitoring of all mu2e slow control channels
  b. Sending Process Variables (PVs) of DAQ hardware info as EPICS channels and PVs settings into EPICS databases

• The Interface plugin:
  a. Performs channel subscription to EPICS using Channel Access EPICS C++ libraries to send and retrieve slow control data information like: Value, Alarm (Status, Severity), Settings
  b. Uses Postgres database C++ libraries to set channels and retrieve channels and alarms histories from EPICS databases
Slow Controls Monitoring in otsdaq

Slow Controls GUI Hierarchy

EpicsInterface.h

Interface #2

Interface #3

Base Class for required functionality

Controls VInterface.h

Supervisors

Controls Dashboard Supervisor.h

Network Communication

SlowControls Dashboard.js

SlowControls Dashboard FileTree.js

SlowControls Dashboard FileTree.html

SlowControls Dashboard Edit.js

SlowControls Dashboard Edit.html

SlowControls Dashboard.html

Widget

Widget

Widget

Sending updates

Widget functionality, Polling for updates
Examples

Example of loaded page

Calorimeter monitor in the slow control GUI
Integration with State Machine

- **State Machine** Configuration and data subscription to EPICS
- Alarm propagation (from EPICS) and *otsdaq* State Machine handling

DAQ HW, artdaq and DQM metrics configuration

[Image of State Machine configuration

[Image of Alarm Configuration

**artdaq** EPICS metrics Plugin

**otsdaq** EPICS Plugin

EPICS
Integration with State Machine

- otsdaq FE (DTC/ROC/CFO) / artdaq metric new channel or new slow control setting → configuring State Machine → EPICS DBs and IOC configuration
- otsdaq Interface → otsdaq CA subscription and DBs select → Monitoring

otsdaq EPICS interface: Channel Access subscription for PVs Values, Settings, Status, Severity

retrieving:
- PVs List and history
- Last Alarms and Alarms Log

EPICS ARCHIVER DATABASE

OTS IOC

EPICS ARCHIVER • ALARM • LOG DATABASES

EPICS SERVER

Monitoring
Conclusions

- Mu2e Experiment is under construction at Fermilab and will be ready for data taking in two/three years
- Mu2e TDAQ and slow control are in large part developed according to the requirements (200K events/s for data taking) and hardware tests are going on
- Slow control integration in the online DAQ system, otsdaq, provides an advanced slow controls monitoring, an interface to send otsdaq front-end DAQ hardware, data processing, and DQM slow controls information to EPICS, and a real configuration and Integration with the otsdaq State Machine