H4DAQ:
a modern and versatile data-acquisition
package for calorimeter prototype test-beams

BTTB2019: 7th Beam Telescopes and Test Beams Workshop,
14-18 Jan 2019, CERN,
Geneva (Switzerland)

Giacomo Cucciati, on behalf of the CMS collaboration
Outline

- H4DAQ
- General schema HW/SW
  - Communication: ZMQ
  - Control flow
  - Trigger flow
  - Data flow
  - GUI + Database
- Data acquisition example
- H4DAQ performance
- Future plans
H4DAQ

- **Simplified DAQ system** to test new detectors and electronics in an controlled environment as a test beam facility.

- Created and optimized for **H4 test beam line** at the North Area of the **CERN SPS** for the CMS electromagnetic calorimeter (ECAL) set up.

- Thanks to its modularity, easily adaptable to other test areas like:
  - EA-T9 CERN PS
  - H2 at CERN SPS
  - INFN Frascati Beam Test Facility

- But also used by several groups for prototype testing from 2014 and 2018:
  - New scintillator crystals (W-CeF3 and W-LYSO Shashlik)
  - R&D studies for CMS MTD
  - New ECAL VFE electronics for **HL-LHC Phase 2**
GENERAL HW/SW schema*

Hardware

- 3 plastic scintillators
- Scintillator crystal matrix
- BEAM
- Flag BUSY
- Trigger

Software

- SPS status signals
- Flask Server
- GUI
- DQM
- EB
- DR
- DR
- DR
- Trigger AND not BUSY = ACQ signal

* Setup used during 2018 campaign for ECAL VFE prototype testing.
COMMUNICATION

- H4DAQ software: **multi-applications** distributed system.
- **ZeroMQ** as communication library.
- Advantages:
  - Fast communication, asynchronous I/O
  - Light library available for most languages (C++, Python, "LabView"…)
  - Using of normal TCP over Ethernet connections

**Interface with different machines/OS/Software**

In H4DAQ, used mainly for DAQ applications communication, but also: controlling of the moving table, monitoring of the chiller and the temperature/humidity sensors

- One-way or **SUBSCRIBE** method applied in H4DAQ:
  - An application creates a socket (host+port) and publishes messages
  - The other application subscribes to that socket and cyclically checks for new information published

http://zeromq.org/
CONTROL FLOW

- **Flask server** (Python framework):
  - Interface with the operator (via GUI)
  - Interface with the configuration database (MongoDB)
  - Instantiate DAQ applications (RC, DR, EB)

- **Run Control** (RC, C++) follows the evolution of the data acquisition:
  - Receives commands from the user
  - Configure the run
  - Send commands to sub-applications (DR, EB) such as “enable acquisition” at beam extraction (i.e. spill signals from SPS)
  - Monitor sub-application status
TRIGGER FLOW

- Timing optimized for CERN-North Area delivery: ~5s spill every ~30s

- **During a spill:**
  - Triggers built from coincidence of three plastic scintillators along beam line
  - Readout hardware BUSY flags joined by OR logic
  - Acquisition allowed if trigger AND no BUSY flags (logic built via NIM modules)

- **Between two spills:**
  - Slow communication and data transmission
  - Prompt reco + DQM + EOS transfer

- **Clock:**
  - VFEs use a common clock from a simplified FE for synchronization
DATA FLOW

- Each **readout board** (VFE) controlled by a **Data Readout** application (C++/python):
  - Following the commands from the Run Control
  - Extracting each event from readout unit mem just after its acquisition
  - Sending the event fragments to the Event Builder
  - **Custom hardware code** running inside pre-define FSM functions

- **Event Builder** (C++):
  - Creating events from fragments matching them by timestamp between two spills

- **DQM**:
  - Reading data stream and preparing a preview of data
GUI and DB

- **GUI:** a web application (Vue.js library) to control and configure the run.
- **DB:** MongoDB to store hardware and application configurations and run info.
- **Inspired by** CMS-ECAL run control system.

**Main feature with respect to previous versions:**
the user has full control of the run and of the configuration remotely
GENERAL HW/SW schema*

Hardware

SPS status signals

3 plastic scintillators
Scintillator crystal matrix

Trigger AND not BUSY = ACQ signal

Flag BUSY

Software

RC

DR

EB

DQM

Flask Server

GUI

* Setup used during 2018 campaign for ECAL VFE prototype testing
GENERAL HW/SW schema

**Hardware**

- SPS status signals
- 3 plastic scintillators
- 2 x MCP
- BEAM
- Trigger
- v1742 digitizer
- Scintillator crystal matrix
- Flag BUSY
- ACQ signal
- VFE

**Software**

- RC
- DR
- EB
- DQM
- Flask Server
- GUI
- FSM

* Setup used during 2018 campaign for ECAL VFE prototype testing*
GENERAL HW/SW schema*

Hardware

- SPS status signals
- 3 plastic scintillators
- 2 x MCP
- Trigger
- v1742 digitizer
- Flag BUSY
- Scintillator crystal matrix
- ACQ signal
- Flag BUSY

Software

- RC
- DR
- EB
- FSM
- DQM
- Flask Server
- GUI

* Setup used during 2018 campaign for ECAL VFE prototype testing
**GENERAL HW/SW schema**

* Setup used during 2018 campaign for ECAL VFE prototype testing
EXAMPLES OF DATA ACQUIRED

2018 ECAL upgrade campaign for HL-LHC Phase 2*:
- Energy resolution obtained with the first prototype of HL-LHC ASICs amplification chip connected to 25 PbWO$_4$ crystal matrix (ECAL tower) and 160 MHz commercial ADC.

Time resolution of prototype for CMS-MTD:
- Time resolution obtained with LYSO + SiPM tiles acquisition using mainly CAEN v1742 digitizer.
- Prototype custom electronics readout by DR app in python.

* More info in "The CMS ECAL Upgrade for Precision Crystal Calorimetry and Timing at the HL-LHC" presentation, Simone Pigazzini
H4DAQ PERFORMANCE

- Max rate considering interspill “deadtime”: 170 Hz
- Max rate during a spill (live time only): 1012 Hz (4k events x spill)
- Event size up to 100-150 Mb, 1Gb per spill
- Longest run: ~10 hours

Part of the scientific contribution from the recent data-taking:

H4DAQ PERFORMANCE

W/CeF\textsubscript{3} test beam results R&D:

- doi:10.1109/NSSMIC.2015.7581770 "High-energy electron test results of a calorimeter prototype based on CeF3 for HL-LHC applications".
- doi:10.1016/j.nima.2015.09.052 "Test beam results with a sampling calorimeter of cerium fluoride scintillating crystals and tungsten absorber plates for calorimetry at the HL-LHC".
- doi:10.1088/1748-0221/10/07/P07002 "Beam test results for a tungsten-cerium fluoride sampling calorimeter with wavelength-shifting fiber readout".

LYSO+SiPM:

- doi:10.1016/j.nima.2016.05.030 "Detection of high energy muons with sub-20 ps timing resolution using L(Y)SO crystals and SiPM readout".

IMCP test beam results:

- doi:10.1088/1748-0221/12/08/C08014 "Micro-channel plates in ionization mode as a fast timing device for future hadron colliders".
- arXiv:1707.08503 "Response of microchannel plates in ionization mode to single particles and electromagnetic showers".
- doi:10.1088/1748-0221/12/03/C03019 "A fast timing calorimetric layer using micro-channel plates in ionisation mode".
FUTURE PLANS

- **Software:**
  - Generalization of the DAQ Software to increase hardware compatibility
  - Error recognition and auto-recovery of the run (partially done)

- **Hardware:**
  - **Remove** readout unit BUSY signals from trigger decision
  - In place, RC opens acquisition windows when trigger from plastic scintillators and allows channels which are ready to acquire data, receiving an acknowledged when done
  - Main advantage:
    - Decoupling readout units with different rate/deadtime
    - Keep trace easily of the detectors/RU available at each trigger
  - Example:
    - Used pixel detector for high precision positioning of the beam which acquires at **50 Hz** (1 us acquisition + data transfer)
    - VFEs still acquiring at ~**1000 Hz**
CONCLUSIONS

• DAQ system created inside the CMS ECAL group to test electronics prototypes and new detectors in the CERN H4 test beam line

• Modular and easily adaptable to other test environments:
  • EA-T9 CERN PS
  • H2 at CERN SPS
  • INFN Frascati Beam Test Facility

• It has demonstrated to be robust and reliable

• Acquisition rate enough for test campaigns, but strongly dependent on detector applied
  • Main limitation due to data transfer from readout units to PC

• Space for improvements on the software and the triggering system

• Software available on GitLab repository:
  https://gitlab.cern.ch/ecal-daq-upgrade/H4DAQ