



*... for a brighter future*

## *DAQ Software Developments*

*J.L.Schlereth  
February 22, 2008*



U.S. Department  
of Energy

UChicago ►  
Argonne LLC

A U.S. Department of Energy laboratory  
managed by UChicago Argonne, LLC

## *Overview*

- Modules which interact with hardware have been tested
- New modules developed to analyze readout data
- Charge injection scan run type has been implemented
- Configuration reader object has been developed
- Integration with CALICE DAQ

## DHCAL HAL Modules

- Modules use HAL libraries and same tables as existing DAQ
- DhcVmeDevice
  - controls DataCollector (DCOL)
  - one instance for each DCOL module
  - handles configuration of DCOL and DCAL chips
  - handles block transfers from DCOL
- TtmVmeDevice
  - controls Trigger Timing Module (TTM)
  - configures TTM for various trigger types
    - *handles actual trigger for charge injection runs*
    - *cosmic/random trigger handled by polling time stamp FIFO*
    - *noise trigger implemented as wait time interval*
    - *beam spill operation needs development work*

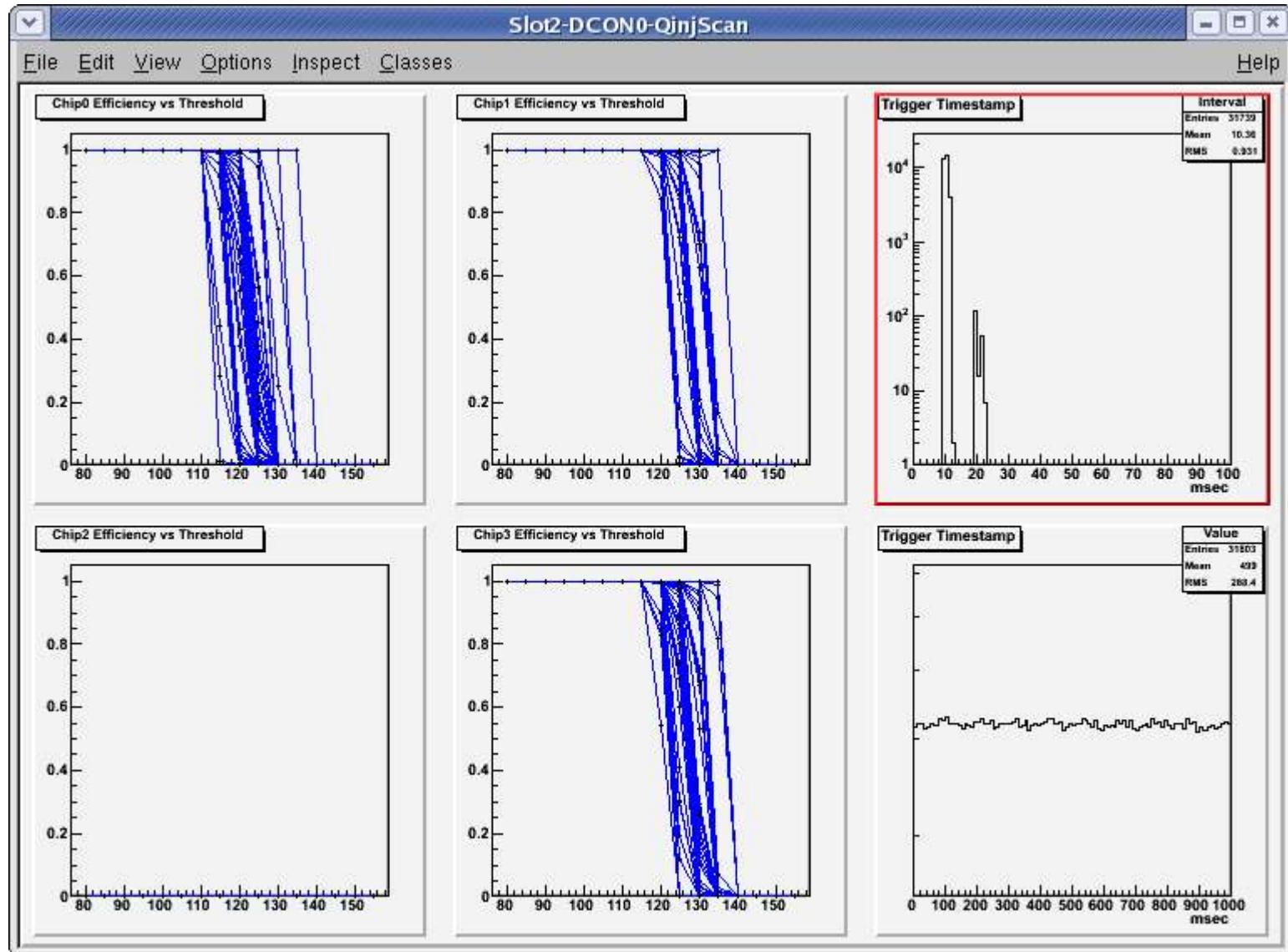
## *Development of analysis modules*

- Needed to verify operation of HAL modules.
- DhcFeHitData
  - unpacks address, status, time stamp and hit data
  - optionally verifies check sum byte
  - basis for converting data files to other formats
- HstDhcQscan
  - extracts configuration and data records from charge injection scan run
  - analyzes all data from selected Data Collector
  - uses DhcFeHitData to unpack raw data
  - accumulates data for average efficiency vs threshold setting.
  - presents data via ROOT interface

## ***Charge Injection Scan Run Type (QinjScan)***

- Modification of charge injection run using multiple configurations
  - vary threshold setting for selected number of steps of selected step size
  - for each threshold setting, run 4 configurations with charge injected into  $\frac{1}{4}$  of the channels.
- Tested on single front end board
  - 64 configurations per run (16 threshold settings)
  - run time 5-6 mins at 100 Hz trigger rate
- Data checked with 'qscan' analysis program offline

# Sample Display from 'qscan' Application



# *Configuration Development*

- Driven by need to be able to change small number of parameters from run to run
- Most parameters applied globally, but some components may need unique settings
- DhcConfigReader
  - provides interface between external file and internal data objects
  - parses XML format file, creates and fills objects compatible with Dhc configuration modules
  - relies on external library, TinyXML package.
- During a run, configuration data is typically written to same file as detector hit data in configuration data sub records

# Sample XML Configuration File

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- @version: -->
<DhcConfiguration>
  <ReadoutConfiguration>
    <dhcal crate="0xdc">
      <trigger slot="8" type="internal"/>
      <pollInterval sec="0" usec="8000"/>
      <enable slot="2" feslot="0xff"/>
    </dhcal>
  </ReadoutConfiguration>

  <BeConfiguration>
    <dcol id="dcol0"
      csr="0x0b"
      dcon_enable="0xff">
      <location crate="0xdc" slot="broadcast" />
    </dcol>
  </BeConfiguration>
<FeConfiguration>
  <dcon id="broadcast"
    con="febroadcast">
    <location crate="0xdc" slot="broadcast" />
    <dcal id="wild"
      chip="21"
      plsr="200"
      intd="139"
      shp2="121"
      shp1="116"
      blrd="81"
      vtnd="0"
      vtpd="80"
      dcr="0x1a"
      inj="0x0000000000000000"
      kill="0x0000000000000000"
      />
  </dcon>
</FeConfiguration>
</DhcConfiguration>
```

## *Integration with CALICE DAQ*

- Paul Dauncey will look into merging modifications and additions for DHCAL
- Remaining issues are clear, but complex
  - triggering, different hardware implemented
  - offline integration, not really part of DAQ except for data records