Silicon tracker integration

Mengqing Wu
on behalf of the LYCORIS group

Bologna, 24/04/2018
AIDA2020 3rd Annual Meeting
WP15.3 task: Silicon strip reference tracker

LYCORIS: a Large Area X-Y Coverage Readout Integrated Strip Telescope

- DESY testbeam infrastructure:
  3 test beam areas w/ 1-6 GeV e-.
- T24/1 area: 1T solenoid to put DUT in magnetic field
  - Movable, internal diameter 85 cm;
  - Possible to install EUDET-style pixel telescope,

Some use cases require:
- **Large activate area**: bent trajectories, smeared electron momenta after magnet wall (~25% $X_0$);
- **Limited space** (~3.5 cm) left for each telescope arm, for large user device, e.g. TPC …

$\rightarrow$ Build a new telescope to fulfill the above requirements in a **complete package**
- DAQ: EUDAQ2
- TLU: AIDA2020-style TLU
EUDAQ2 integration: status

Github repo: https://github.com/Lycoris2017/EUDAQ-Lycoris

- Sensor readout chip KPiX: has its own DAQ
- EUDAQ2 Modules:
  - dedicated RunControl module with its GUI customized with more info
  - dedicated Producer/DataCollector modules with KPiX DAQ soft as dynamic lib
  - corresponding DataConverters in progress
EUDAQ2 integration: status

Store & convert data

- EUDAQ2 output data in format: `.raw` and KPiX binaries;
- Data converted offline for analyses in progress:
  - customized CliConverter executable
  - from eudaq raw to histograms in ROOT
    - for telescope development (time distribution, ADC per channel …)
  - Status: done, but in upgrading with ongoing sensor tests.
- from eudaq raw to LCIO event
  - for full telescope data analysis
  - Status: constructed, pending for event definition as input for reconstruction software.
EUDAQ2 integration: RunControl & Configurables

Status

- KPiX specialized configurables controllable in EUDAQ config file —>

Done:

- Diff. config files prepared: pedestal/source/calibration
- Can control how many KPiX cycles
- Multiple output files available

```system
#*****************************************************************************
## ------LYCORIS-KPiX------
#*****************************************************************************

[RunControl]
#KPiX_STOP_RUN_AFTER_N_SECONDS=10  # stop kpix when Timeout(s)
PRODUCER_TO_CONTROL="Producer.lycorisBkg"
PRODUCER_TO_CONTROL="Producer.lycoris"

###--> pedestal run:
[DataCollector.lycorisBkgDC]
DISABLE_PRINT = 1
EUDAQ_FW_PATTERN="/scratch/data/tracker_test/eudaq/pedestal/kpix_12D_run6R$X"

[Producer.lycorisBkg]
## Choose a data collector by its name:
EUDAQ_DC="lycorisBkgDC"
DISABLE_EUDAQ_BIN=0
EUDAQ_DataBase="/scratch/data/tracker_test/eudaq/pedestal"
DISABLE_KPiX_BIN=0
KPiX_DEBUG="False"
KPiX_CONFIG_FILE="/home/lycoris-dev/kpixDaq/kpix.master.xml/tracker_pedestal.xml"
KPiX_DataBase = "/scratch/data/tracker_test/eudaq/pedestal"
## DataAuto on: auto-generated DataFile name based on DataBase;
## DataFile no use in this case;
KPiX_DataAuto = "True"
KPiX_DataFile = "/scratch/data/daq-dev/eudaq_data.bin"

------- Run Control:
# temporarily choose run state here: Running, Running Calibration, Evr Running
KPiX_RunState = "Running"
KPiX_RunCount = 3000  # not work unless you break kpix data streaming @2018-Feb-13
```
EUDAQ2 integration: RunControl & Configurables

Todos

- KPiX specialized configurables controllable in EUDAQ config file —>

Done:

- Diff. config files prepared: pedestal/source/calibration
- Can control how many KPiX cycles
- Multiple output files available

Todo:

- add/rm configurables in EUDAQ —> need more commissioning experience;
- document all the LYCORIS relevant configurables.
EUDAQ2 integration: data buffering

Validation

- EUDAQ2 using FIFO to stream data from KPiX DAQ
  - many other ways tried, failed (time shift…)
- Validate data collected by EUDAQ2:
  - compare w/ data from KPiX DAQ.
- **Target**: sanity check w/ diff. run conditions (external trigger/internal trigger, sync/unsync to beam)
  - data output from both KPiX DAQ side and EUDAQ2 side in the same run.
- **Status**: validated
  - internal trigger: perfect agreed
  - external trigger: agreed w/ understood delay issue (not affect using EUDAQ2)
TLU integration: KPiX sync to DESY beam

Status

- **KPiX** performs w/ power cycle (chip developed for an ILC environment);
- **DESY-II** has its own cycle —> synchronization needed to maximize particle events when KPiX is active to take data;
- KPiX synced to **TLU** via a common clock;

—> **Synchronization** needed, implemented via a shutter signal generated by TLU to start the KPiX acq. Cycle

- **KPiX** can **timestamp** event using **TLU global clock**;
- flexible to accommodate different DUTs (next slide)

**Status:**

- idea **tested/validated** by feeding a manipulated shutter signal to KPiX
- —> to test with an AIDA TLU.
TLU integration: KPix sync to DUTs

DUT syn to TLU

In general, DUTs can be categorized: sync to TLU or not

1. a DUT sync to the TLU:
   - The TLU common **global clock** to **sync** all devices;
   - Activation (shutter) issued by TLU
   - Busy signals (TLU state idle) to TLU: either global or local
     - **Global**: no trigger when any device is busy;
     - **Local**: trigger continuously issuing though some device busy
TLU integration: KPiX sync to DUTs

DUT syn to TLU

In general, DUTs can be categorized: sync to TLU or not;
1. a DUT sync to the TLU:
2. a DUT unsync to the TLU:
   - Synchronization by trigger counting;
   - Global busy used: no trigger sent, when either device is busy;
   - Add-on TLU func: configurable delayed TLU active period

Status:
- new TLU will be issued by end of this month;
- a first use case needed to test.

Shutter: activation signal
T0_sync: 1 per run, common start signal from David Cussans
Status summary & Todo-list

Summary

**EUDAQ2 integration:**
- Can do RunControl with KPiX needed configurables!
- Validated: The current FIFO data buffering in EUDAQ (w/o changing the KPiX data polling sequence) from KPiX to EUDAQ is validted under different run conditions ()
- Add-on func: customized CliConverter executable —> output in LCIO or ROOT (histograms).

**AIDA2020-style TLU integration:**
- a new TLU will be sent by end of this month;
- a new DAQ board exp. in May: to use TLU clock with common run start T0_sync;
- Sync to DESY-II beam: idea tested;
- Sync to different DUTs: to test.

Todo

- Particle event matching from various devices at reco level —> Event definition?
- To test new TLU with telescope, and with telescope and a DUT.
Status summary & Todo-list

Summary

- **EUDAQ2 integration:**
  - Can do RunControl with KPiX needed configurable
  - Validated: The current FIFO data buffering in EUDAQ (w/o changing the KPiX data polling sequence) from KPiX to EUDAQ is validated under different run conditions ():
  - Add-on func: customized CliConverter executable —> output in LCIO or ROOT (histograms).

- **AIDA2020-style TLU integration:**
  - a new TLU will be sent by end of this month;
  - a new DAQ board exp. in May: to use TLU clock with common run start T0_sync;
  - Sync to DESY-II beam: idea tested;
  - Sync to different DUTs: to test.

Todo

- Particle event matching from various devices at reco level —> Event definition?
- To test new TLU with telescope, and with telescope and a DUT.

Thank you for your attention!
Back-ups
Opening with brief review

- AIDA2020-WP15: To build a new large area strip telescope within solenoid in T24
  - **Demands:**
    - Large curvature coverage
    - Small space (~3.5 cm) for large DUT (e.g. TPC)
    - Good spatial resolution along the bending direction ($\sigma_y \approx 10 \mu m$)
    - To use EUDAQ as other telescopes at DESY
  - **Review:**
    - SiD strip sensor designed by SLAC,
      - 10x10 cm$^2$ active area, **pitch** of 25 $\mu$m, **thickness** of 320 $\mu$m, **material budge** of 0.3% $X_0$;
      - produced by Hamamatsu, delivered end 07/2017
      - bump-bonded by IZM, arrived on 23/11/2017
    - KPiX studied with ECal pixel sensors at DESY
      - well prepared to move to strip sensor.
KPiX sync with DUT unsync to TLU

- ALTRO electronics run independently of other devices without sending any time information.
- Matching will be done by exclusion of unmatched events using busy signals.
Software status update: lab test succeeded

- KPiX core code not changed anymore
- KPiX data can be outstreaming at same time:
  - from KPiX side (as if running KPiX DAQ);
  - from EUDAQ side: in **both KPiX binary** format and **EUDAQ raw** format.

- event mapping on the ECAl sensor for KPiX tests at DESY
  - same sensor, same bucket