Status of the Software for the LP

Martin Killenberg

universitätbonn

Software for the Large Prototype

Goal: Compare different readout technologies

Common framework for all readout technologies:
→ ILCSoft / Marlin

Common data format:
→ LCIO

Common geometry description:
→ GEAR

DAQ

- Goal: Common reconstruction of all sub-detectors
(pads with altro/after, TDCs, TimePix, silicon strip hodoscope, pixel beam hodoscope...)
- Minimum requirements:
  - All DAQs have to be syncronised (Trigger Logic Unit)
  - All DAQs have to write LCIO
- Ideal:
All DAQs are implemented as „Eudaq Producers“
http://projects.hepforge.org/eudaq/
  - Run control
  - Eudaq Data Collector currently collects raw data events from different producers and combines them to one data stream
=> Have to discuss with Eudaq and LCIO developers to be able to combine LCIO events from different subdetectors (ADCs, TDCs, TimePix, Silicon)

Pad Geometry: GEAR

Two pad Layouts available:

- FixedPadSizeDiskLayout
  - Circular geometry, pad rows on circles
  - All rows heights and pad pitches are the same
  - “first” pad in row is aligned with x-axis
- RectangularPadRowLayout
  - Pad height and pitch can vary row by row
  - Number of pads can vary row by row
  - rows can be staggered

ToDo:

- Currently only one pad plane per TPC. Has to be extended (e.g. one “pad plane” per module)
- Extend the disk layout (e. g. staggering angle per row)

MarlinTPC - Overview

Currently 51 Processors in different sections:

- Simulation
  - Primary ionisation for detailed digitisation
- Digitisation
  - Detailed digitisation and fast “Mokka to RawData”

- **Calibration**
- **Reconstruction**
- **Analysis**

- Validation
  for debugging the code
- Tools

TPCCondData: Conditions Data Objects for the TPC

MarinTPC - Calibration

Nothing implemented yet

What do we need?
- Pedestal calculator
- Gain calculator
- Alignment calculator

Per channel:
- Channel correction calculator
- Time shift calculator

MarlinTPC - Reconstruction Overview 1

TrackerRawDataToDataConverterProcessor

TimeShiftCorrectorProcessor

PedestalSubtractorProcessor

GainCorrectorProcessor

ChannelMapperProcessor

CountsToPrimaryElectronsProcessor

PulseFinderProcessor

HitTrackFinderTopoProcessor

HitFinderProcessor

TrackFinderRectangularProcessor

TrackFinderHoughTrafoProcessor

TrackSeederProcessor

TrackFitterLikelihoodProcessor

LinearRegressionProcessor

MarlinTPC Reconstruction Overview 2

TimePixZeroSuppressionProcessor

TimePixHighTOTAnalyserProcessor

TimePixMapHandlerProcessor

TimePixPixelInterpolationProcessor

TimePixClusterFinderProcessor

TimePixClusterProjectionSeparatorProcessor

TimePixHitCenterCalculatorProcessor

TimePixHitSorterProcessor

MarlinTPC Reconstruction 1

TrackerRawDataToDataConverterProcessor

Simply convert to TrackerData (all correctors have TrackerData as in and output, so they can be exchanged/ left out)

PedestalSubtractorProcessor

Correct pedestal per channel from conditions data

TimeShiftCorrectorProcessor

Correct time shift according to conditions data

GainCorrectorProcessor

Correct per channel gain fluctuations from conditions data

ChannelMapperProcessor

Map hardware channels to GEAR pad indices

CountsToPrimaryElectronsProcessor

Apply the global gain calibration factor from conditions data

MarlinTPC Reconstruction 2

PulseFinderProcessor

Search for pulses in ADC data, negative or positive polarity

Calculate time (using readout frequency from GEAR) and charge

HitFinderProcessor

Combine neighbouring pads in one row to hits

Calculate 3D coordinate from GEAR (all pad layouts) and drift velocity from conditions data

ToDo: Take into account broken and noisy channels

TrackFinderRectangularProcessor

Search for straight tracks in RectangularPadRowlayout

TrackFinderHoughTrafoProcessor

Search for straight tracks independently from geometry

MarlinTPC Reconstruction 3

HitTrackFinderTopoProcessor

Search for contiguous area on pad plane (incl. z cut)
Calculate hits on the rows
Works for all GEAR geometries
ToDo: Take into account broken and noisy channels

TrackSeederProcessor

Analytically calculate estimated values for track parameter as seed for a track fitter (both helix and straight line)

LinearRegressionProcessor

Analytically calculate track parameters for straight line (identical to χ2 minimisation)

TrackFitterLikelihoodProcessor

Calculate track parameters (straight line and helix by maximising likelihood of measured signals on pads (all GEAR geometries).

TrackFitterChiSquare (in preparation)

Calculate track parameters independently from readout geometry

MarlinTPC - Analysis

MarlinTPCAnalyser Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

marlintpc::BiasedResidualsProcessor	*Fill the residuals of the hits into an **AIDA** histogram*
marlintpc::HitAndTrackChargeProcessor	*Fill charge per hit and per track length into **AIDA** histograms*
marlintpc::HitAndTrackCounterProcessor	*Count hits in events and tracks, and tracks per event*
marlintpc::LinearGeometricMeanResolutionProcessor	*Resoltuion calculator for straight lines using the gemetric mean method*
marlintpc::LinearThreePointResolutionProcessor	*Resoltuion calculator for straight tracks using the three point method*
Processor	
marlintpc::TimePixClusterSizeProcessor	*Fill the number of pixels in a cluster and the cluster radius in a histogram*
marlintpc::TimePixOccupancyProcessor	*Count how many times a pixel has been hit on the TimePix chip*
marlintpc::TimePixTOTDistributionProcessor	*Fill the TOT values of all pixels in a histogram*
marlintpc::TrackParametersDistributionProcessor	*Fill **AIDA** histograms with all track parameters: phi, lambda, tan(lambda), d0, z0*
marlintpc::XYZDistributionProcessor	*Fill the x, y, and z positions of all hits into an **AIDA** histogram*
marlintpc::XYZDistributionTracksProcessor	*Fill the x, y, and z positions of the hits on tracks into an **AIDA** histogram*
marlintpc::ZBinTemplateProcessor< N1DHISTOS, N2DHISTOS >	*A template to create processors wich create 1D or 2D distribution histograms, one histogram per z bin*
marlintpc::ZBinTemplateProcessor< N1DHISTOS, N2DHISTOS >::HistoNBins2D	*A nested convenience struct which contains the number of bins for both directions of a 2D histogram*
marlintpc::ZBinTemplateProcessor< N1DHISTOS, N2DHISTOS >::HistoRange	*A nested convenience struct which contains the minimal and maximal value of a (1D) histogram rage*
marlintpc::ZBinTemplateProcessor< N1DHISTOS, N2DHISTOS >::HistoRange2D	*A nested convenience struct which contains two 1D **HistoRange** objects*

MarlinTPC Analysis (2)

What else do we need?

- Pad occupancy
- Geometric Mean resolution for helix tracks

Will start discussion thread in the forum.

Conditions Data

- LCIO data objects (see next slides)
- Need data base server which can be accessed from everywhere
- Should be written automatically / user has to provide it for evey run

What data do we need?

Who will set up the data base? Where do we host it?

Discussion thread in forum?

TPCCondData

ADCChannelMapping

Mapping of H/W channels to GEAR pad indices

- ChannelID
- PadID
- Type

ChannelCorrection

Per channel calibration

- Quality flags (broken, noisy)
- Calibration factors
- Time offset

Pedestal

per channel

- Value
- Width

TPCConditions

Calibrated TPC Parameters

- DriftVelocity
- Diffusion (trans/long)
- "Defocussing"
- Amplification

GasConditions

- Mixture
- Pressure
- Temperature
- OxygenContent
- WaterContent

FieldSettings

- Nominal drift field
- Nominal B-Field

especially for GEMs:

- GEM voltages
- Transfer fields

WeatherConditions

- float Temperature
- float Humidity
- float Pressure

TimePixPixelMode

- Mode
- Status (broken/noisy)

What else is missing?

- Event Display