

# BT 2014 Telescope Data Reconstruction. Telescope and LumiCal Tracks Matching.

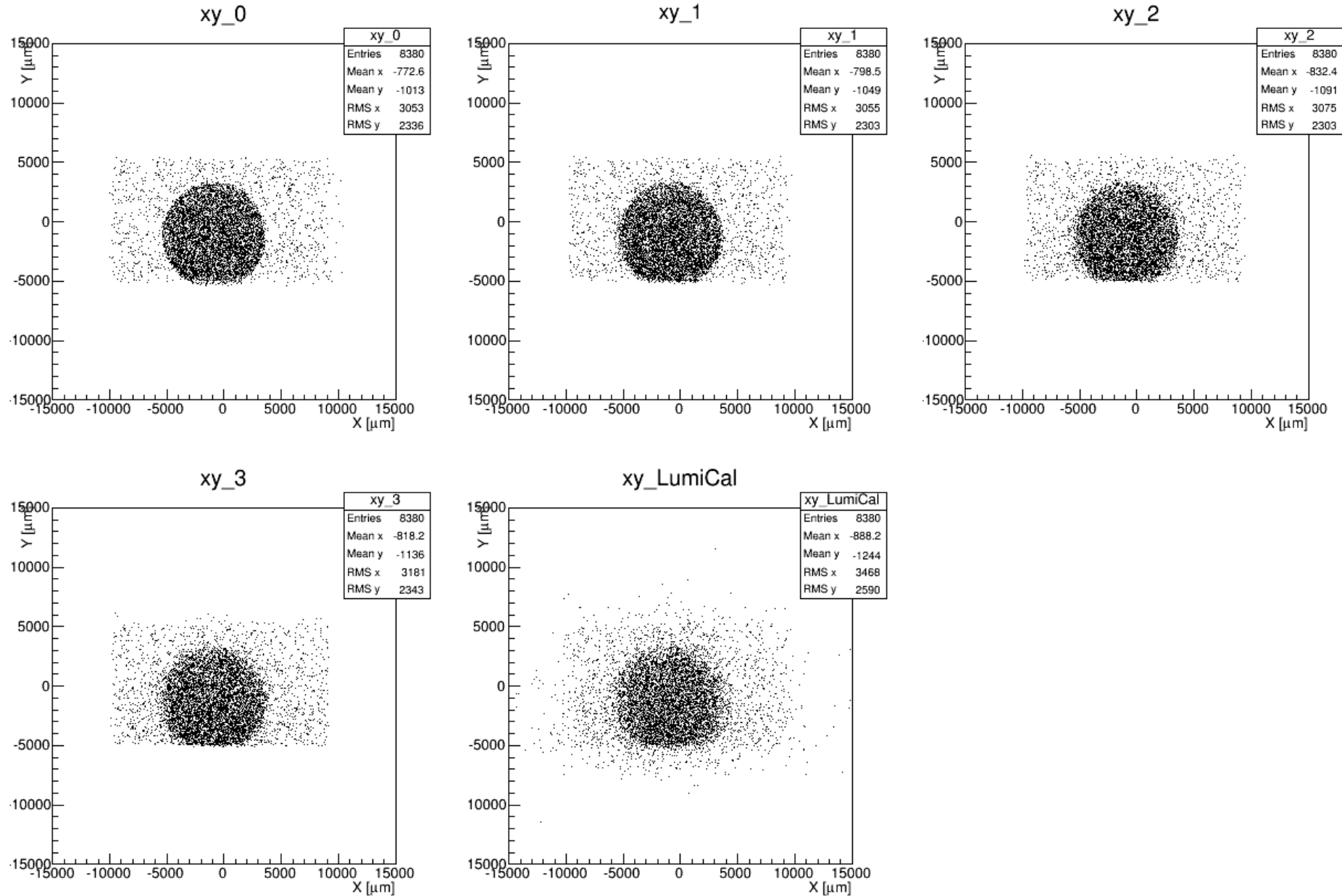
I.Levy, O.Borysov

TAU. December 15, 2014

# Software for Telescope Data Reconstruction

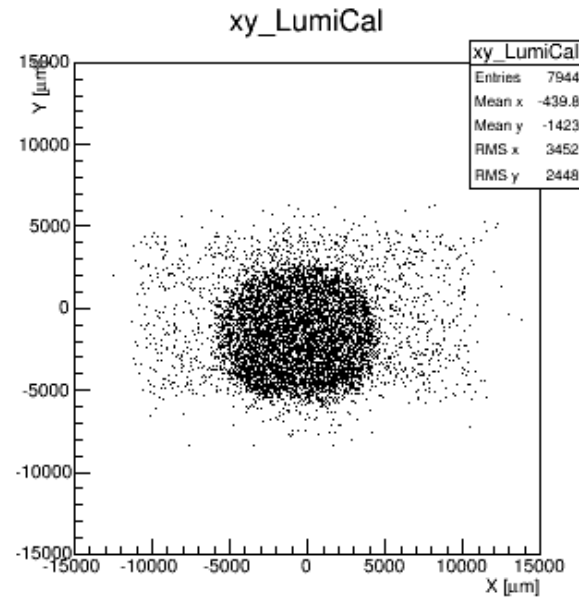
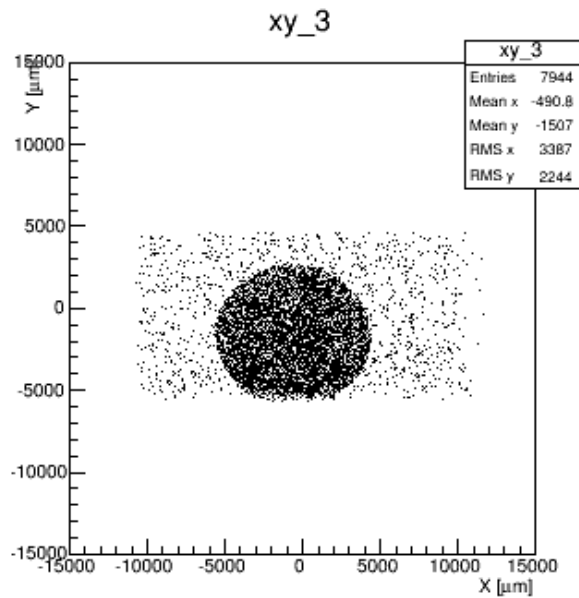
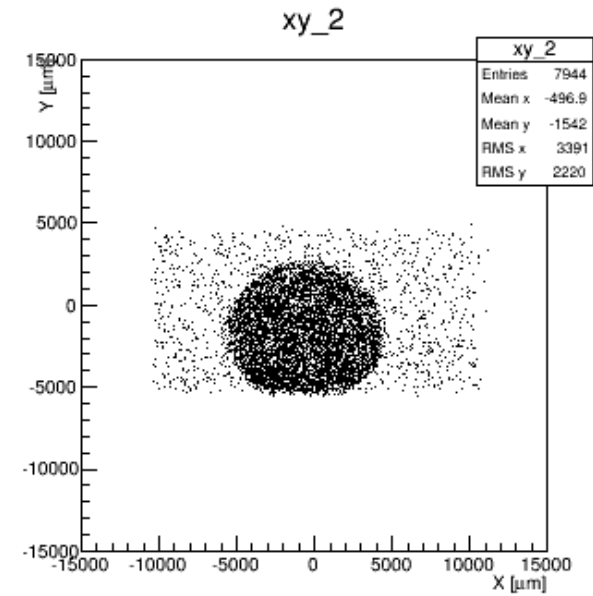
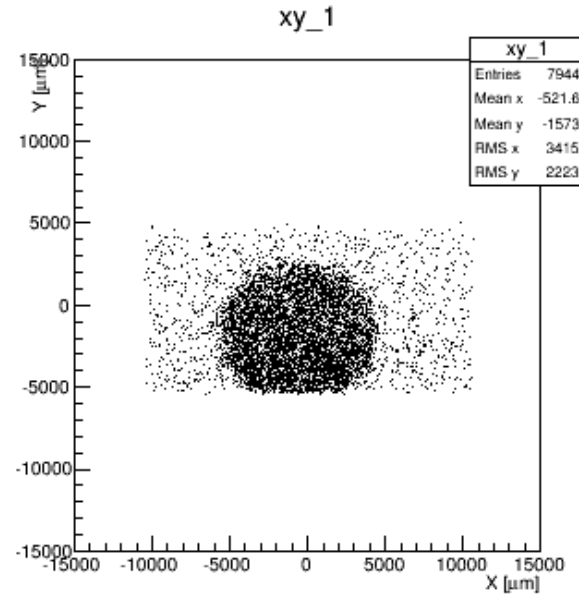
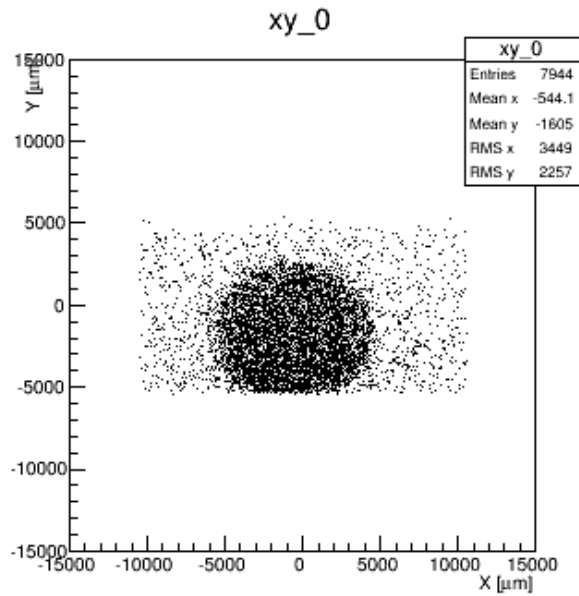
- TAF is used for reading raw data files from Mimosa sensors and producing hits;
  - developed in IPHC (Strasbourg) using MIMOSA Analysis Framework (MAF).
  - <http://www.iphc.cnrs.fr/Public-documentation.html>;
  - requires root and configuration file for data analysis.
- Aarhus telescope reconstruction software:
  - alignment;
  - tracks reconstruction;
  - requires: root, armadillo c++ linear algebra library (depends on LAPACK).
- TAF has been modified to save the hardware frame number in output tree.  
(gTAF->DSFProcudtion(...) output);
- Aarhus telescope reconstruction software has been modified:
  - to produce tree with tracks and particles position in first sensor of LumiCal.
  - to copy frame number from TAF to output tree;
  - Small changes in track finding procedure.
- pxie\_reader\_test.C macro produces additional tree which is used by telescope reconstruction software to combine frames for the same trigger.

# Telescope Run 48 (w/o LumiCal). Hits. “Wrong” Beam Direction.

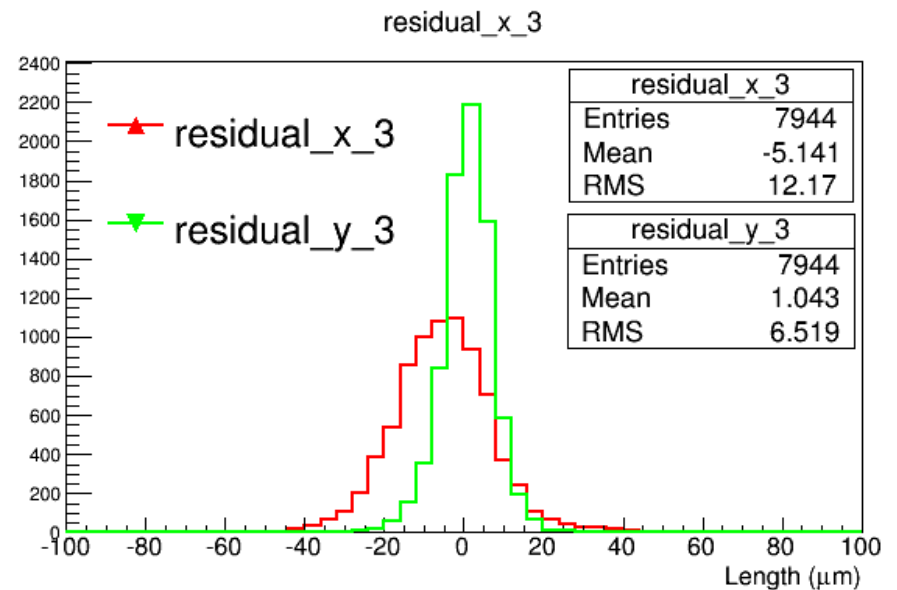
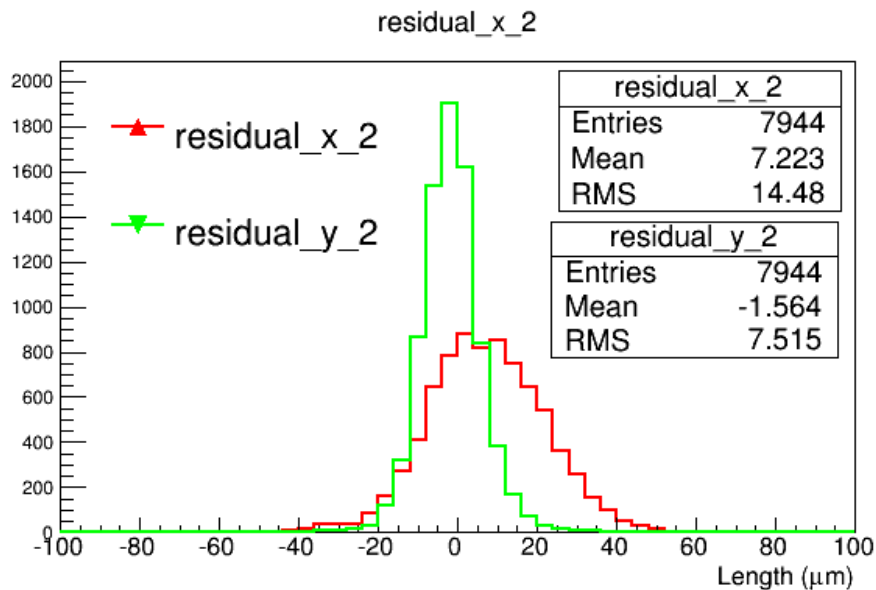
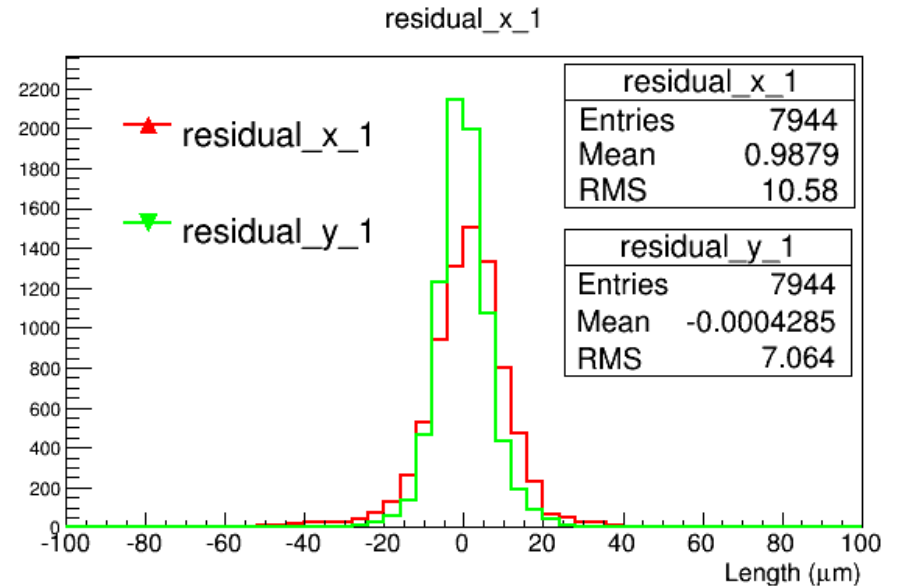
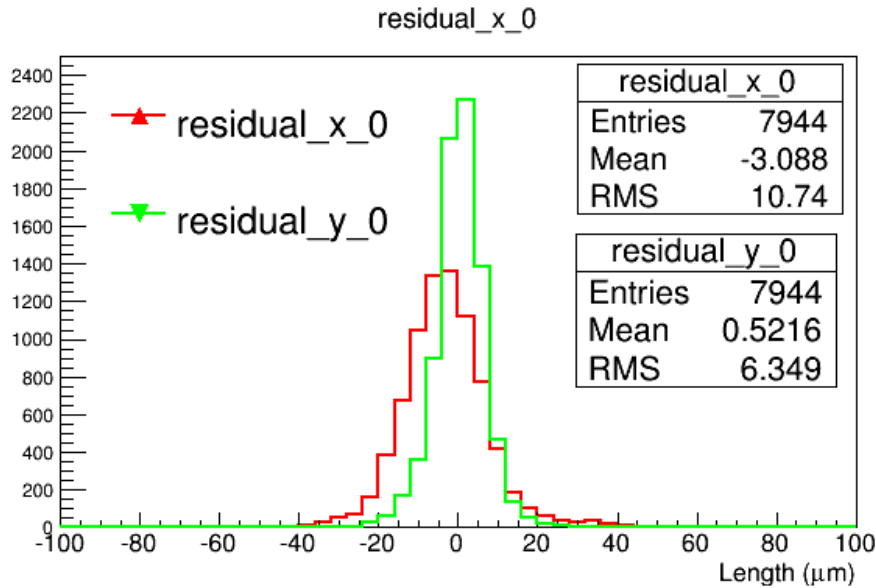


# Telescope Run 48 (w/o LumiCal). Hits.

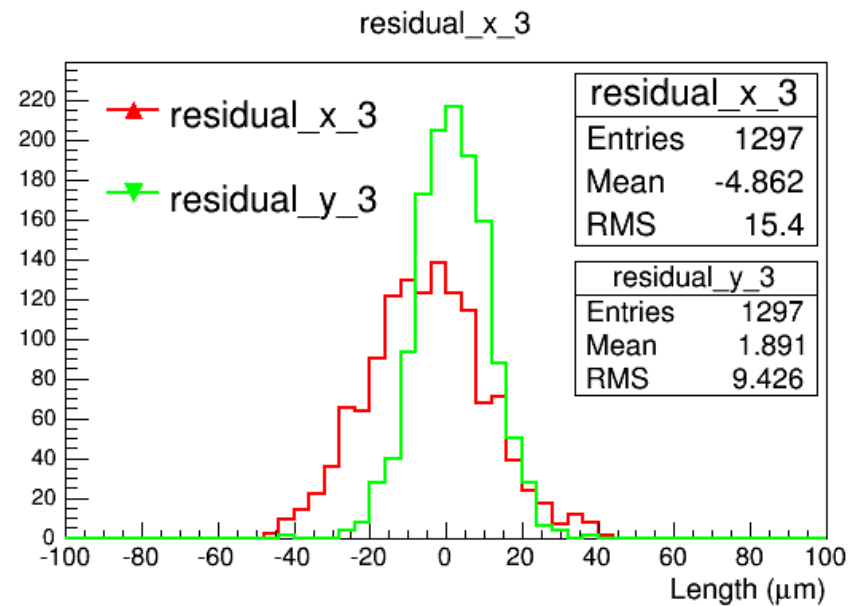
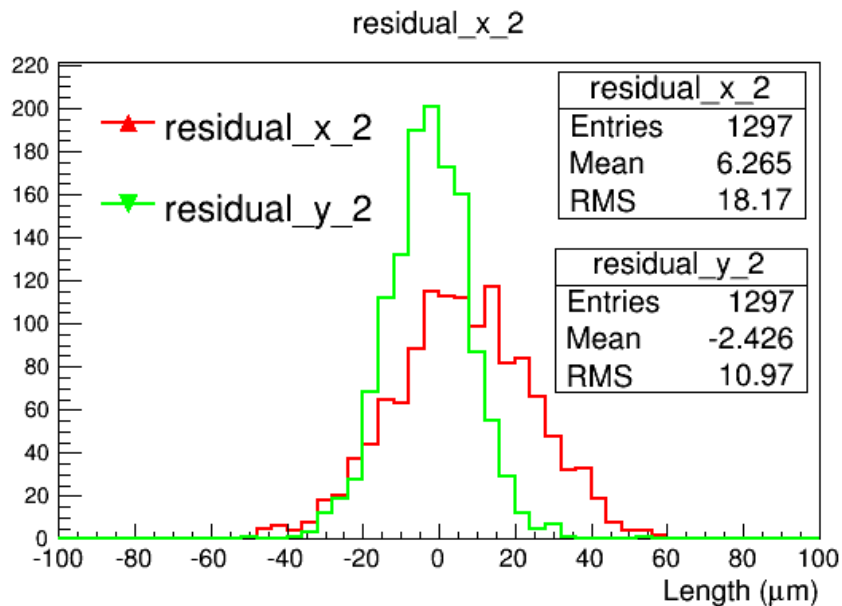
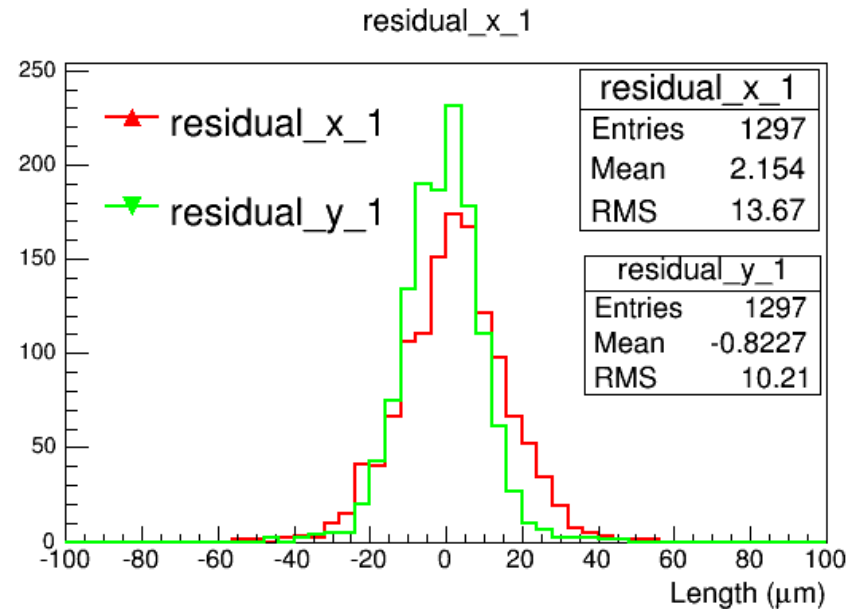
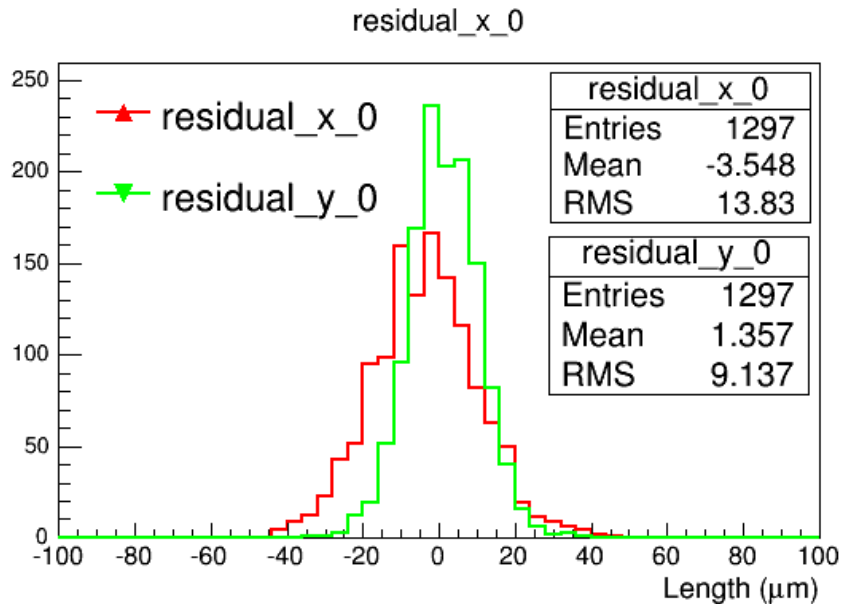
## Correct beam direction.



# Telescope Run 48. Track Residuals.

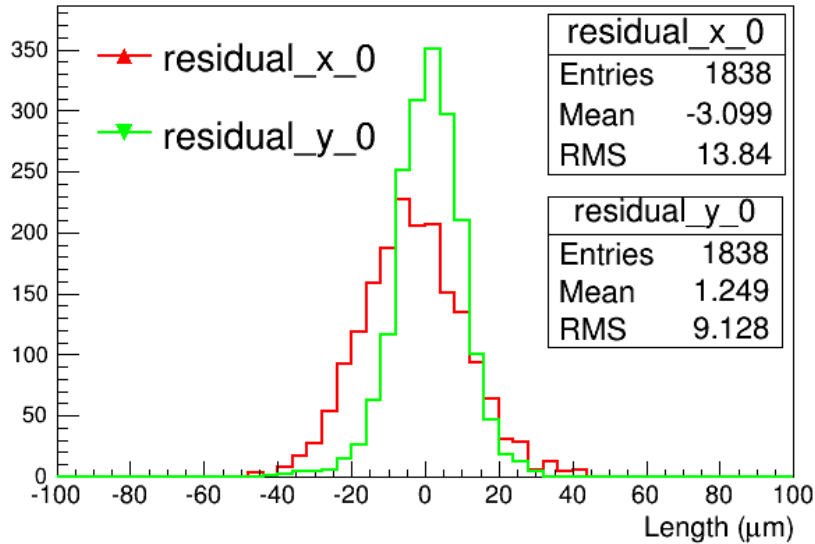


# e, $\mu$ Runs 99, 180. Residuals.

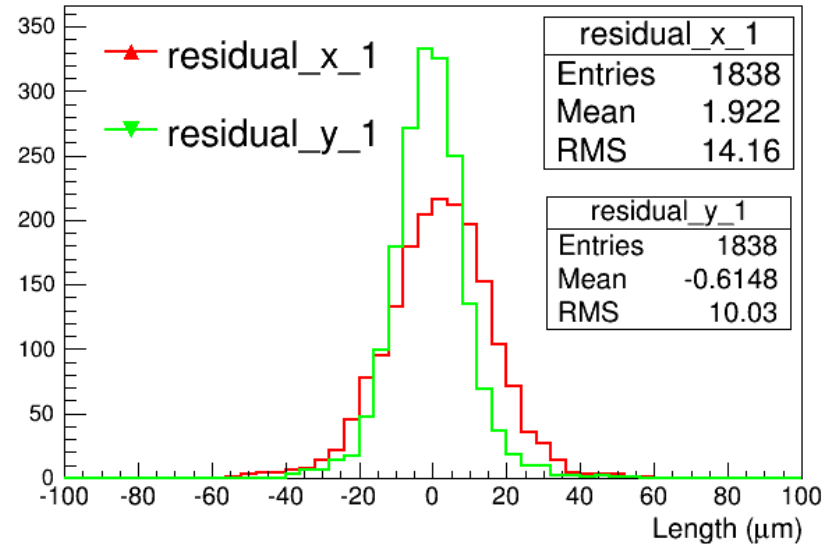


# Hadron Run 233. Residuals.

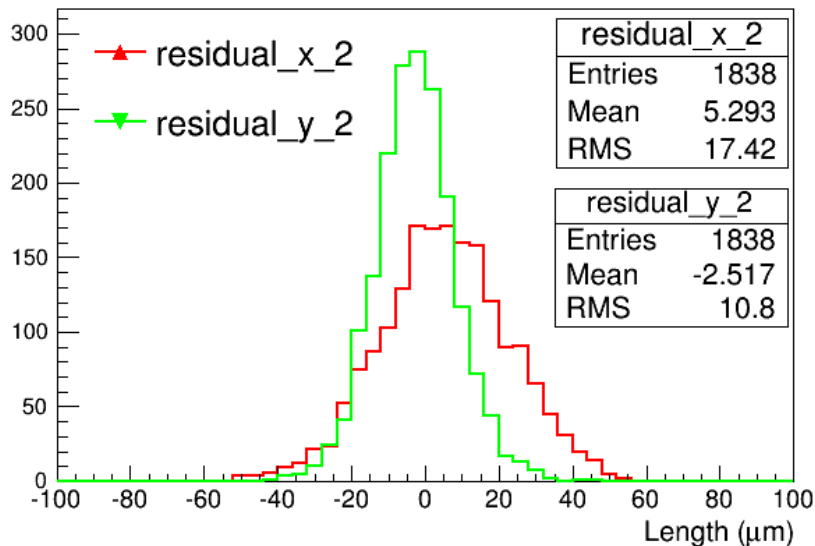
residual\_x\_0



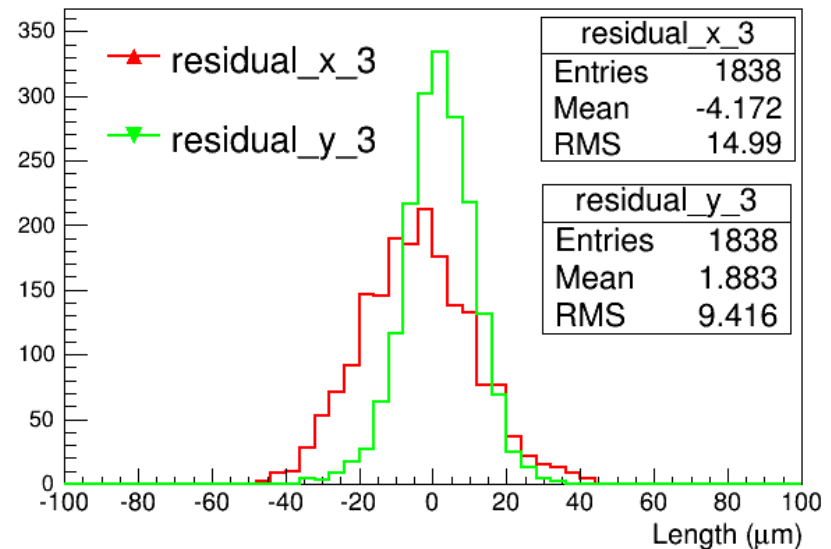
residual\_x\_1



residual\_x\_2



residual\_x\_3



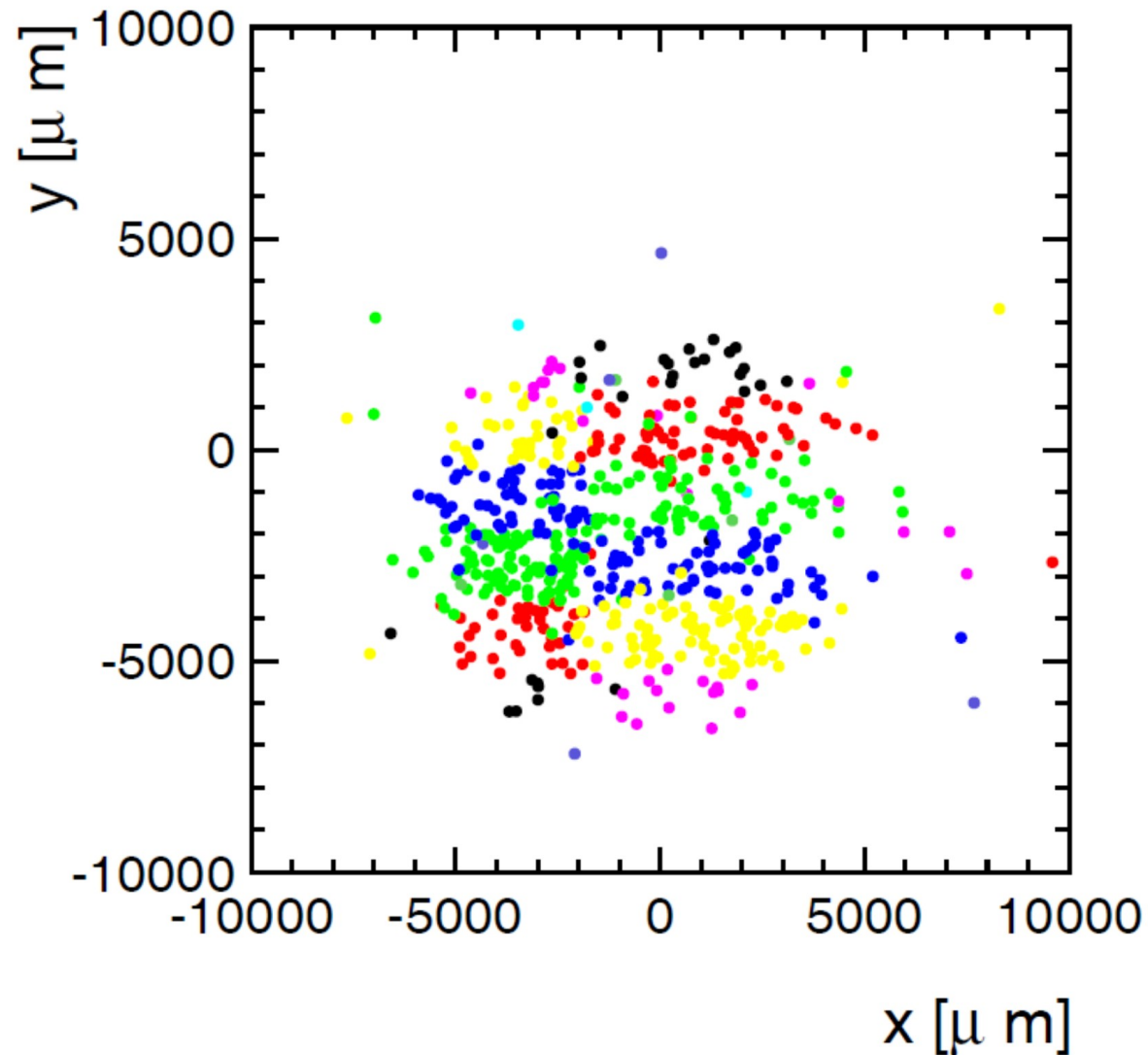
# Tips and trick for sync

- Open data from 2 files in 3 trees (3 indexes) :
  1. LumiCal data
  2. AUX data (TLU number & frame counter)
  3. telescope tree.
- The AUX data is used as the main arbiter.
- The first step is to identify the starting difference between the AUX and LumiCAL data and telescope data can be 0 or 1 (depend on configure process at the beginning of a run), and to identify if the jump in frame counter between consecutive triggers is regular or double (compared with jumps in telescope data).
- Inside the loop over all events in a file :
  1. Check if TLU moves by 1. (if not moved there is a problem, if moved  $> 1$  there were missing events and you need to adjust the AUX counter and the difference between AUX and telescope).
  2. Check if TLU numbers from AUX and LumiCAL are the same, if not, likely there is a missing event in the LumiCAL data, and you need to skip some event in the AUX (and in Telescope) .
  3. Check if the jump in frame counter from the previous to the current trigger is the same (or double) in both AUX and telescope data. If it's not, probably there is a missing event at the telescope and you need to move to the next trigger and change the difference between AUX and telescope by  $-1$  .



# Data combination result

- Here is an example of data combination, from hadron run in configuration 2, Run 380/233 (LUMI/TELE).
- We extrapolate a single track from telescope to LumiCal first sensor plane to generate a hit map that looks like the trigger profile (9 mm circle).
- Then we assign to the hit a color according to the single channel that had signal ( $> 4 \cdot \text{RMS}$ ) in sample 17.

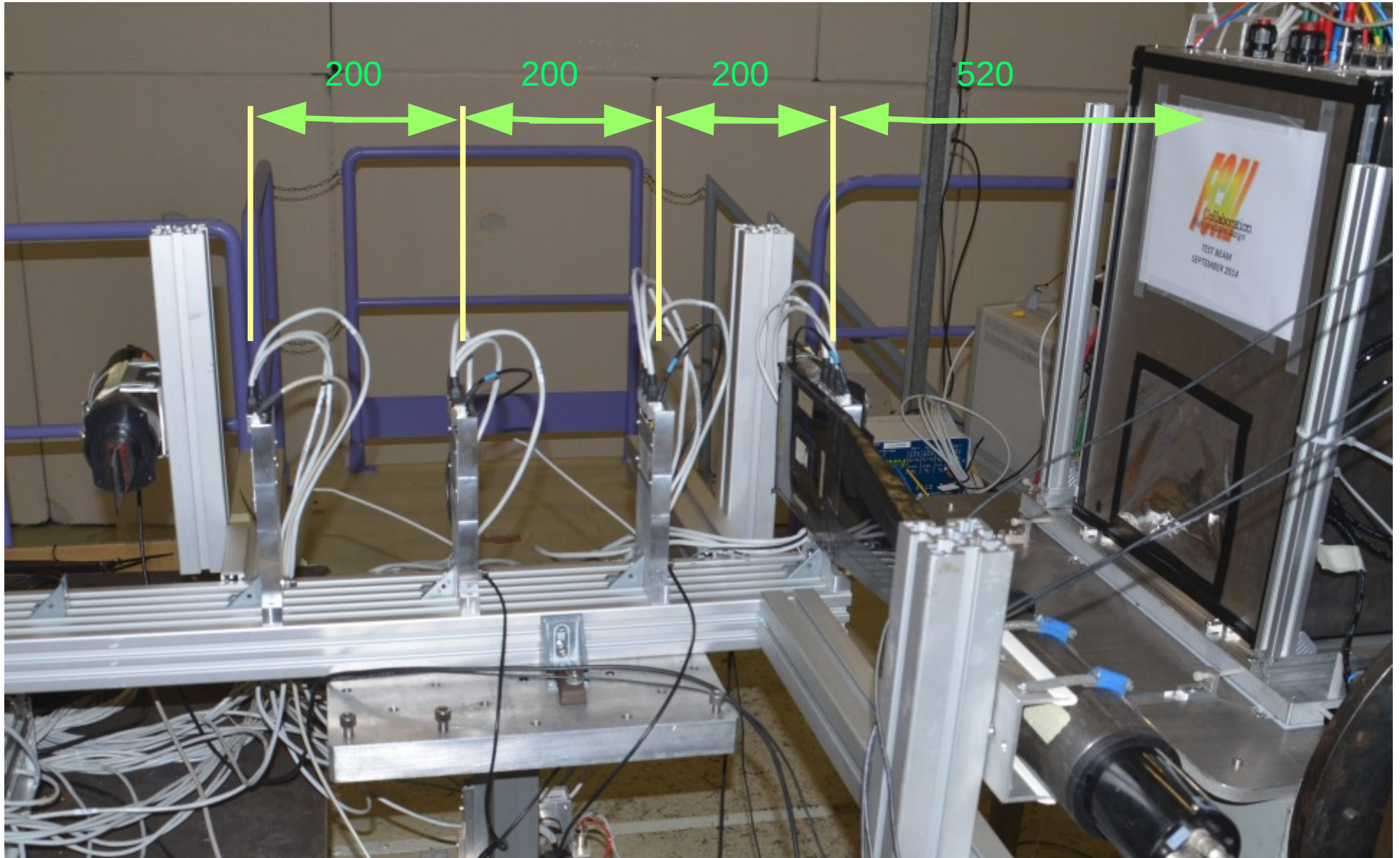


# Summary

- Hardware frame number from mimosa chip is available in telescope reconstructed tree.
- Reconstruction using TAF and Aarhus software shows good results. Though the alignment procedure is not perfect and can be improved.
- Synchronization between telescope and LumiCal based on Mimosa hardware frame number works.

# Backup

# Telescope and LumiCal Layout



# Output Tree with Telescope Data

```
fTelescopeTree = new TTree("ATelescope", "ATelescope");  
fTelescopeTree->Branch("NAllHits", &fNAllHits, "NAllHits/I");  
fTelescopeTree->Branch("AllHits", "std::vector< std::vector< TVector3 > >", &fpAllHits);  
fTelescopeTree->Branch("NTracks", &fNTracks, "NTracks/I");  
fTelescopeTree->Branch("TrackHits", "std::vector< std::vector< TVector3 > >", &fpTrackHits);  
fTelescopeTree->Branch("TrackCentr", "std::vector< TVector3 >", &fpTrackCentr);  
fTelescopeTree->Branch("TrackDir", "std::vector< TVector3 >", &fpTrackDir);  
fTelescopeTree->Branch("FrameId", "std::vector< Int_t >", &fpFrameId);  
fTelescopeTree->Branch("LumiCalPos", "std::vector< TVector3 >", &fpLumiCalPos);
```

# Combining Frames

PXIeBoardReader board 0::GetNextDaqEvent() current frame 0 over 400 in current acquisition -1 over 374

PXIeBoardReader board 0: **Got frame 0 (absolute nb = 2953710)** in acquisition 0.

\*\* board 0 : count 00000000 : AcqId = 0000 - FrameId = 0109 - #triggers = 0001 (or 0001)  
trigger[0, in frame 0] at line 0367 (0367/0367 & frame 856557 & invalid 0), time = 63731

PXIeBoardReader board 0::GetNextDaqEvent() current frame 0 over 400 in current acquisition 0 over 374

PXIeBoardReader board 0: **Got frame 1 (absolute nb = 2953711)** in acquisition 0.

\*\* board 0 : count 00000001 : AcqId = 0000 - FrameId = 0110 - #triggers = 0000 (or 0000)

PXIeBoardReader board 0::GetNextDaqEvent() current frame 1 over 400 in current acquisition 0 over 374

PXIeBoardReader board 0: **Got frame 2 (absolute nb = 2953712)** in acquisition 0.

\*\* board 0 : count 00000002 : AcqId = 0000 - FrameId = 0111 - #triggers = 0000 (or 0000)

PXIeBoardReader board 0::GetNextDaqEvent() current frame 2 over 400 in current acquisition 0 over 374

PXIeBoardReader board 0: **Got frame 3 (absolute nb = 2953713)** in acquisition 0.

\*\* board 0 : count 00000003 : AcqId = 0000 - FrameId = 0112 - #triggers = 0000 (or 0000)

PXIeBoardReader board 0::GetNextDaqEvent() current frame 3 over 400 in current acquisition 0 over 374