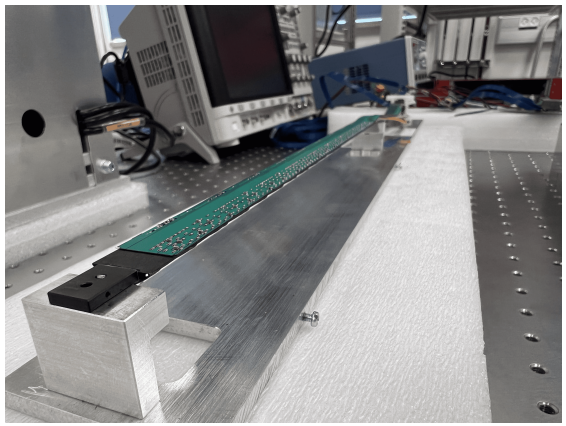
ALPIDE-Stave DAQ development at WIS

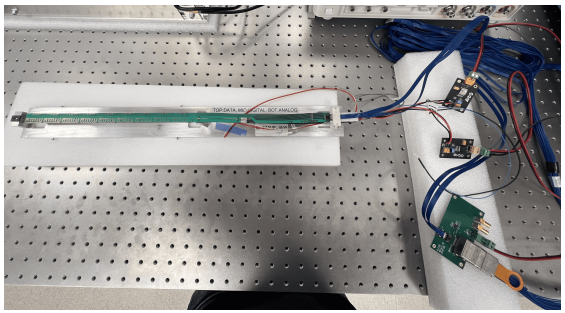
Noam Tal Hod, Oleksandr Borysov, Arka Santra, Roman Urmanov

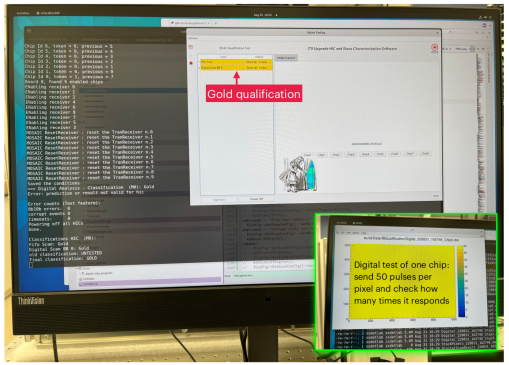
01.02.2023

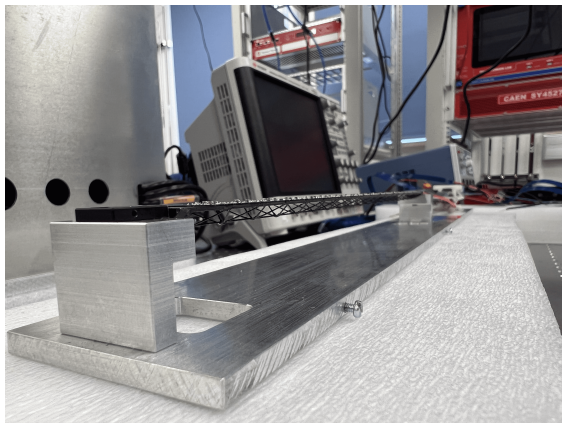
Where we've left off

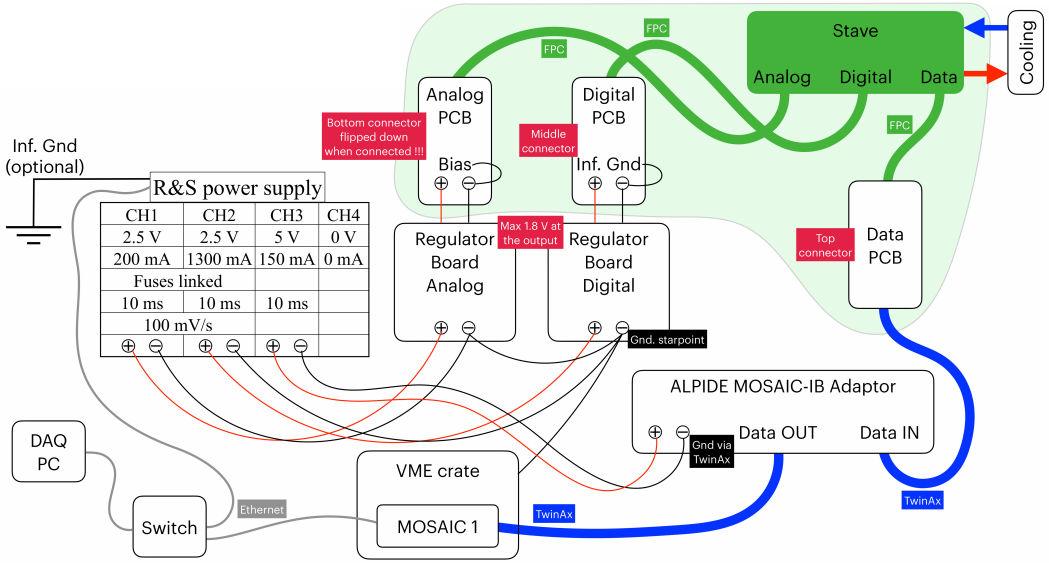
- ALPIDE-Stave has been purchased along with 8 single ALPIDE cards and periphery;
- Stave setup has been assembled (see Fig. below);
- Initial qualification tests have been performed. Stave has qualified as "Gold".







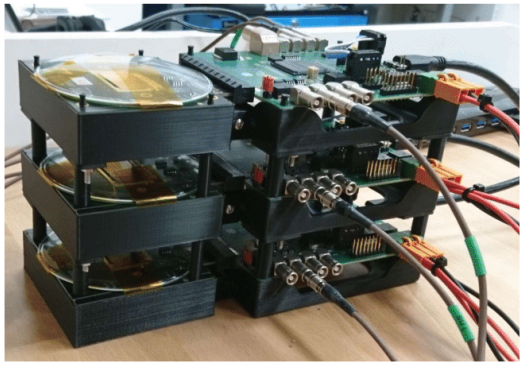
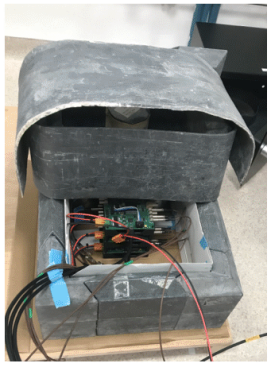
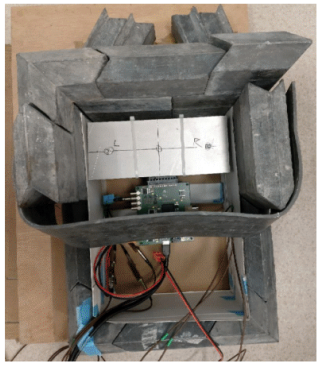




DAQ system: Introduction

- A collection of EUDAQ2-based DAQ libraries is currently being developed;
- Goal: Self-contained DAQ library, that allows for control of the detector and its periphery:
  - Configuration routines for readout board, ALPIDE chips, peripheral components;
  - Data readout, formatting, and sychronization across multiple detector components. Storing of data in a file;
  - User interface of varying degree of complexity;
- Parallel development of two systems: single ALPIDE Telescope, ALPIDE-Stave;
- At present – core functionality has been realized and tested;

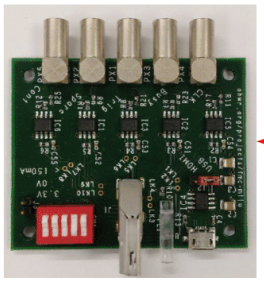
DAQ system: Test setup – Telescope

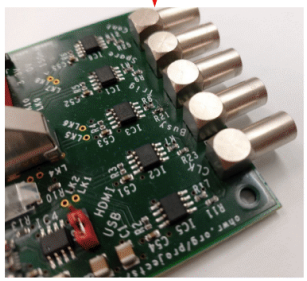

DAQ/ /Control PC

AIDA TLU

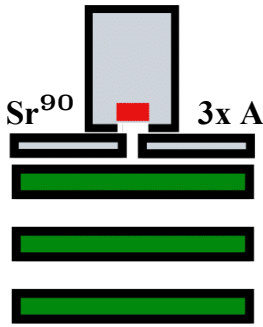
Pulse Generator



LVDS to LVTTL Adapter

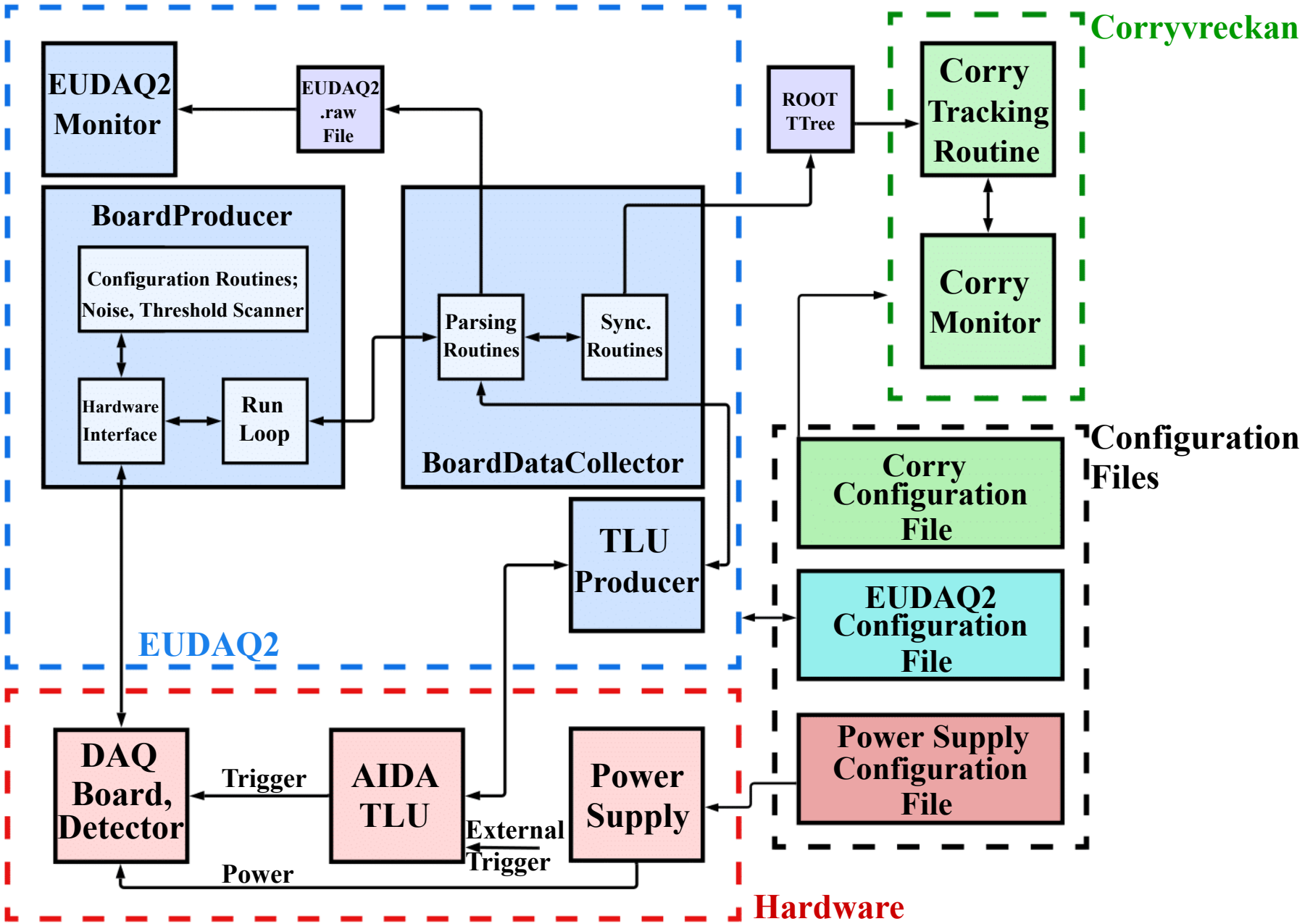


ALPIDE Telescope

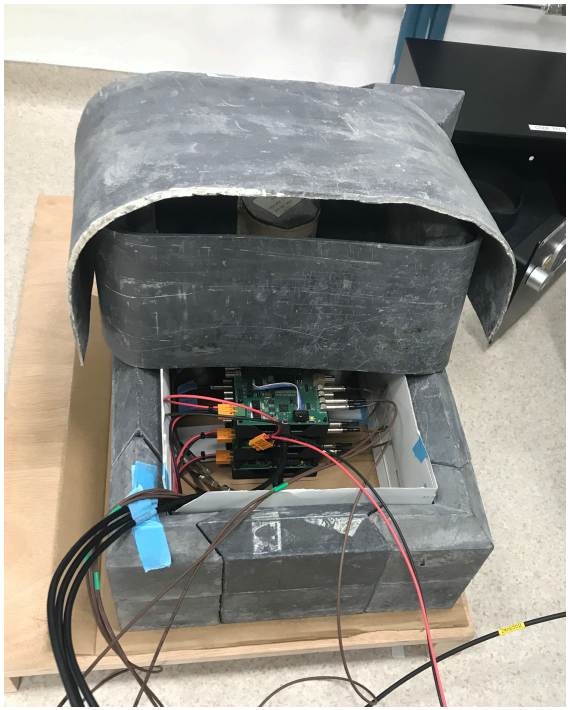


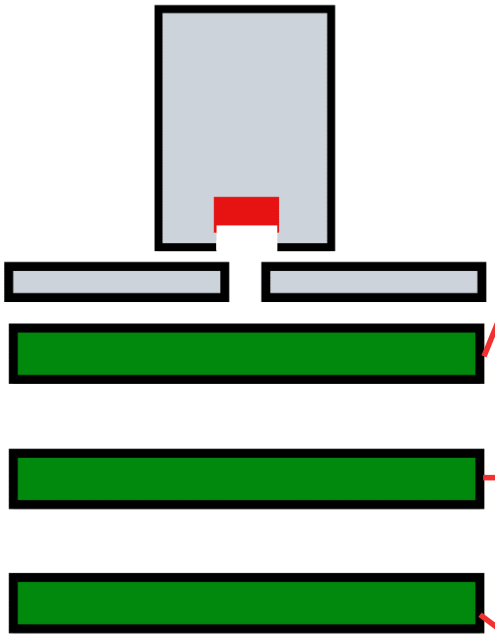
Power Supply

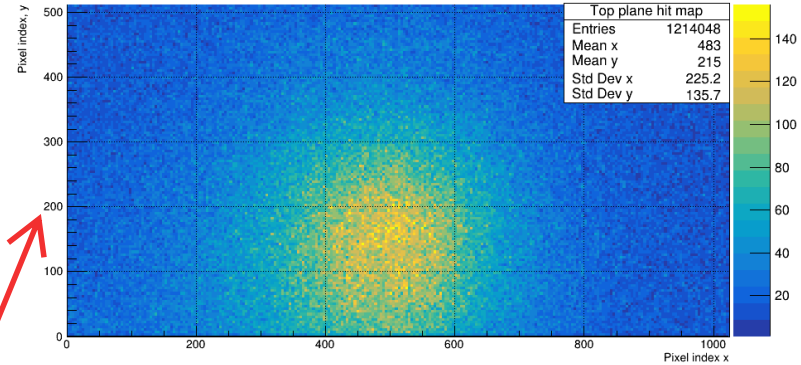
DAQ system: Flow of control and data – Telescope

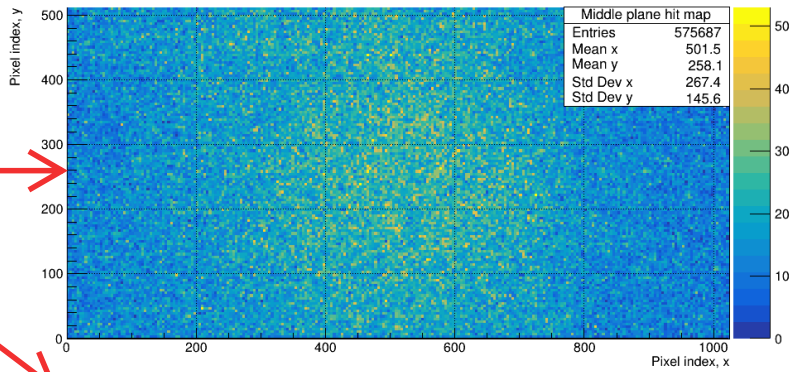


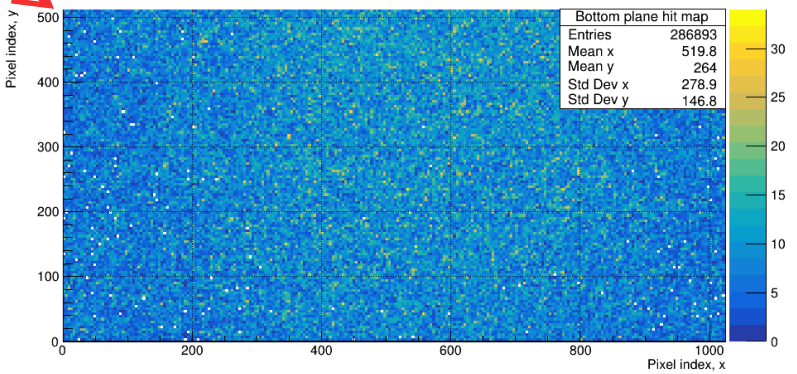
DAQ system: Test runs – Telescope





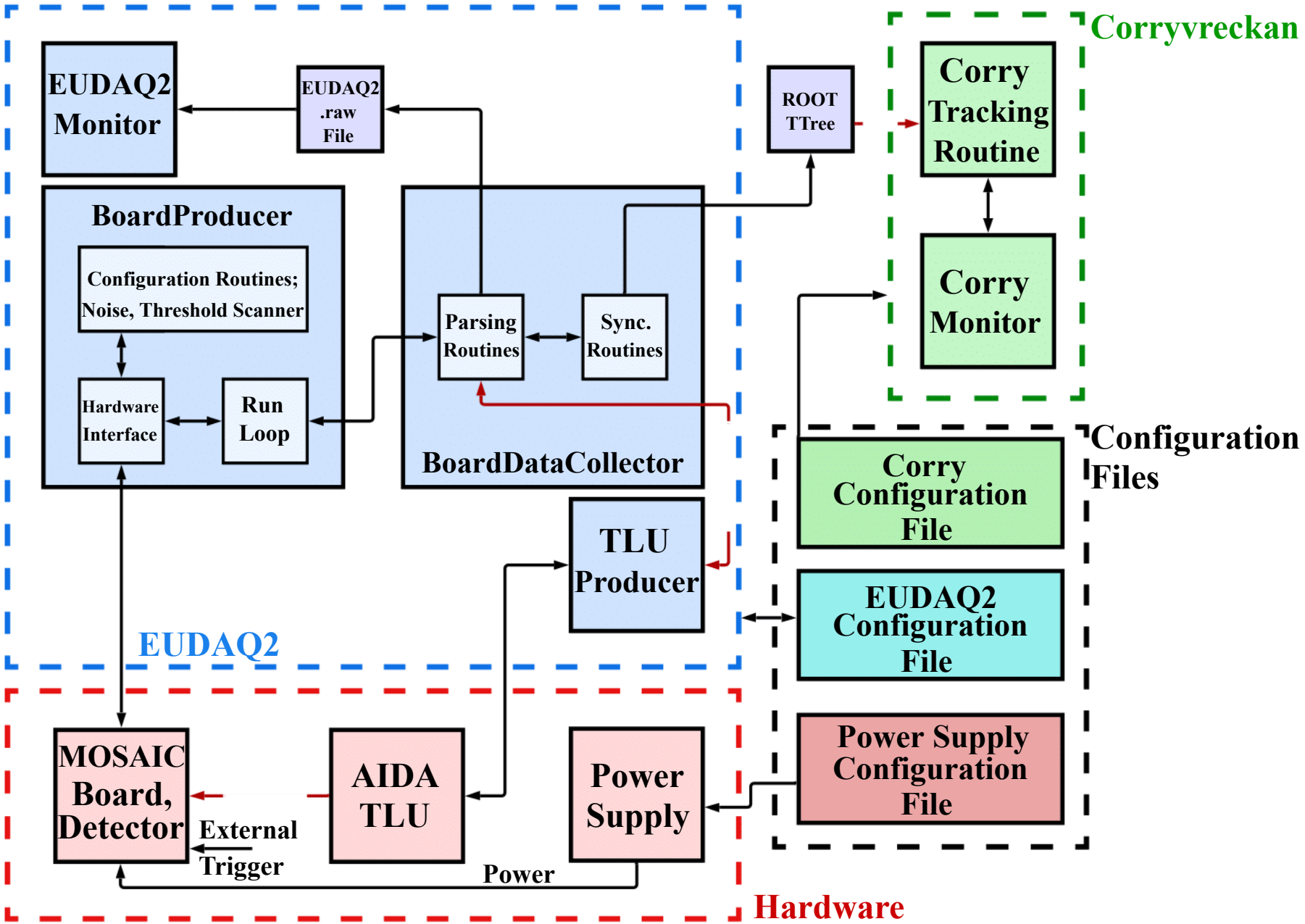






- **Readout tests: Digital pulsing, Electronic noise, Cosmic, Radioactive source;**
- **"Toy" data showed realized readout schemes to be working;**
- **External trigger – 1 kHz, ALPIDE strobe length – 0.95 ms;**
- **Real data is currently being analyzed – to be updated;**

DAQ system: Flow of control and data – Stave



DAQ system: Test setup – Stave

Pulse Generator

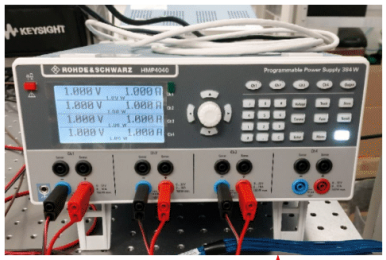
VME Crate

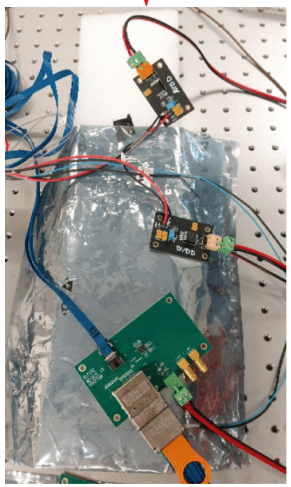
MOSAIC Board

MOSAIC-ALPIDE Adapter
Digital/Analog Power regulators

Sr^{90}

ALPIDE-Stave



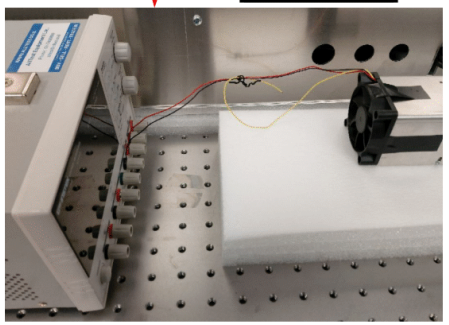


ETH switch

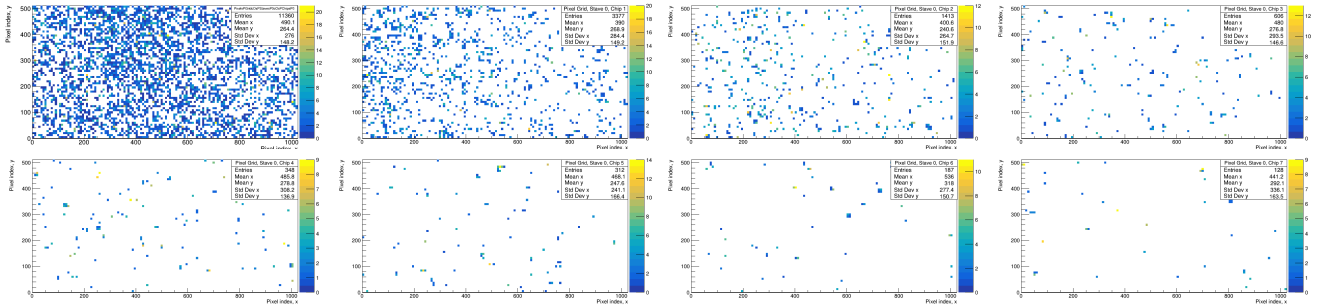
DAQ/ /Control PC

Power Supply

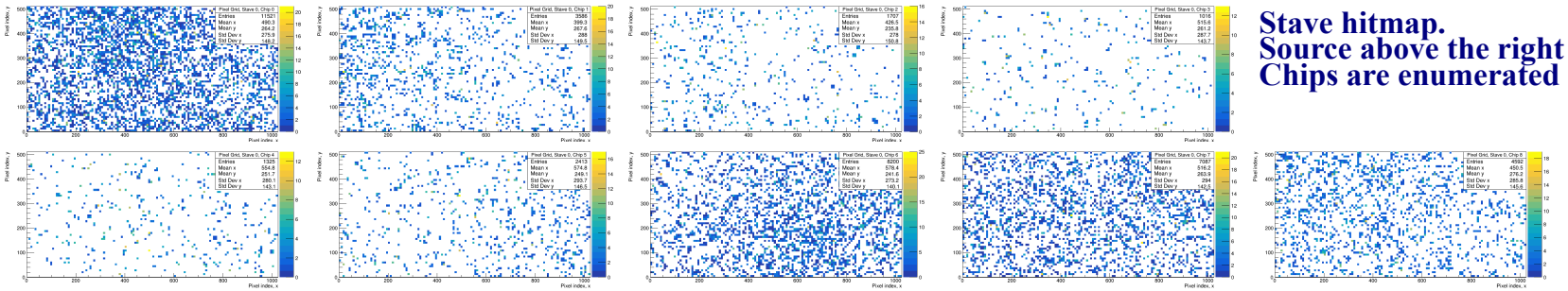
PC Fan



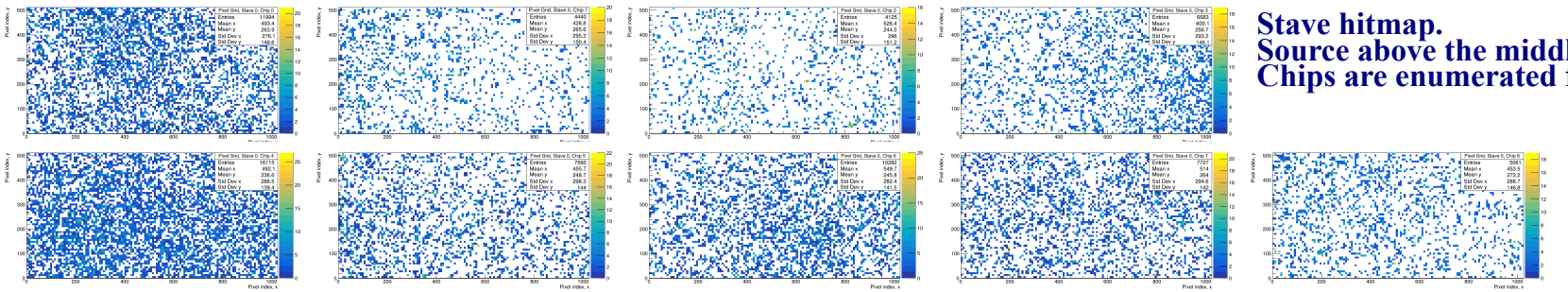
DAQ system: Test runs – Stave



Stave hitmap.
Source above the left end.
Chips are enumerated from left to right.



Stave hitmap.
Source above the right end.
Chips are enumerated from left to right.

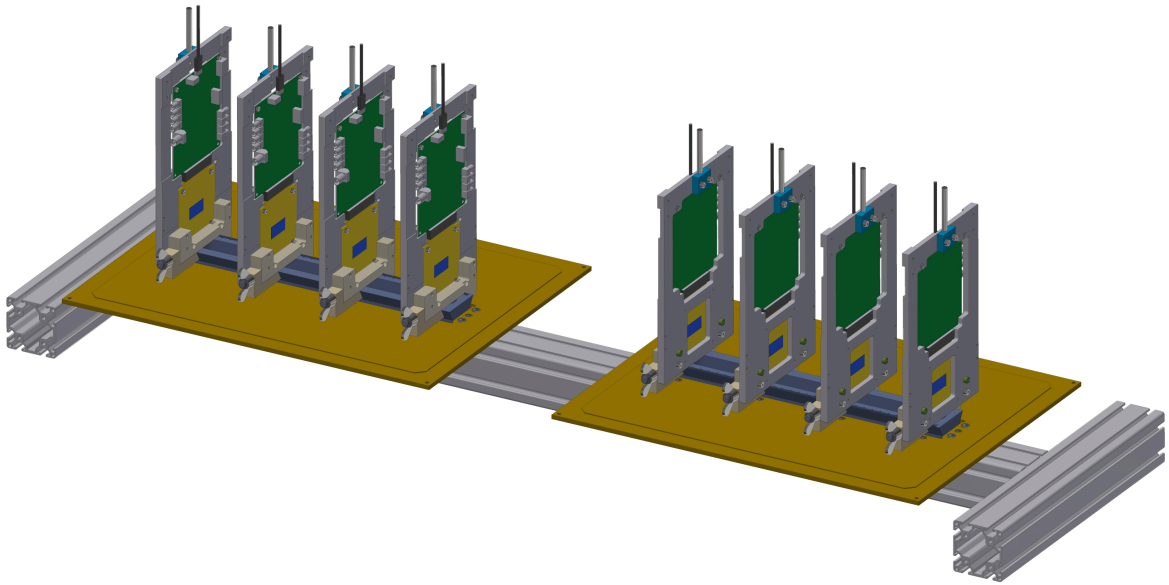


Stave hitmap.
Source above the middle.
Chips are enumerated from left to right.

- External trigger frequency – 1 kHz. Chip strobe window – 95 ms;
- Noise scans have been performed for the whole stave. Threshold accross chips has matched (approximately);
- Source has been placed in: 1. One end of the stave; 2. Opposite end of the stave; 3. Middle of the stave;
- Obtained hitmaps correlate with the source's positions and order of placement;

Further plans

- Implement trigger ID logic in the MOSAIC firmware (adapt NA61 solution);
- PMTs has been purchased for the particle-driven-triggering test. Awaiting for the arrival;
- Assemble Stave-Telescope stand (Fig. below) and perform beam-test runs;
- Measure readout characteristics quantitatively (readout speed, throughput, memory usage, etc.).



Backup: DAQ system: Noise scans

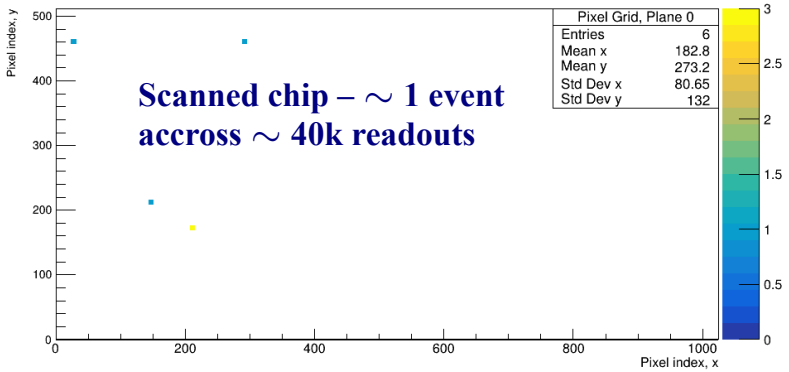
Digital noise scan

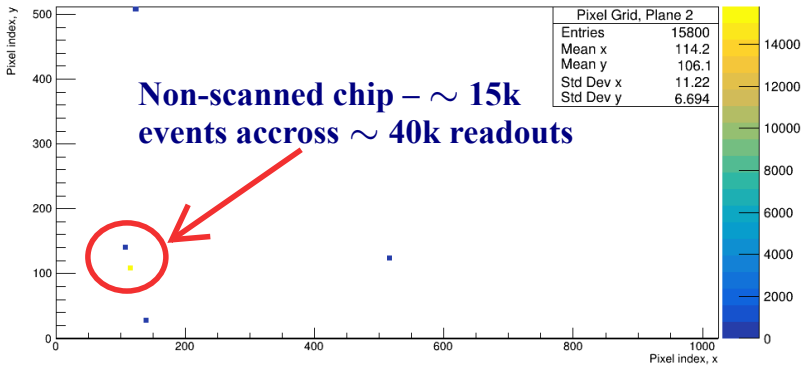
- Mask all matrix rows but one;
- Readout matrix without pulsing (strobe window $\sim$ 1 us);
- Look for the hits. Depending on whether they're masked assign active pixels status "Stuck" (unmasked) or "Bad" (masked);
- Readout matrix after digitally pulsing it (strobe window $\sim$ 1 us);
- Look for the hits. Depending on whether they're masked assign active pixels status "Unpulsable" (unmasked) or "Unmaskable" (masked);
- Perform multiple iterations for each row. Calculate probability of pixel being noisy.

Noisy pixel indeces and respective probabily are stored in a flat-format file for later use in masking procedures

Fake hit rate scan

- Configure ALPIDEs for the desired threshold;
- Readout matrix without pulsing it (strobe window $\sim$ 1 ms);
- Look for active pixels.
- Perform multiple iterations of matrix readout. Calculate probability of pixel being noisy.





Backup: DAQ system: Threshold scan

- Detected threshold discrepancy between different chips – up to 90 units of charge;
- Procedure of threshold matching has been implemented;
- Discrepancy is reduced by an order of magnitude;

Select desired threshold in units of charge. Fix all but one (ITHR) DAC registers responsible for the threshold.

→

Iterate over the ITHR values measuring pixel's S-curve. Repeat procedure for every pixel in the chip matrix.

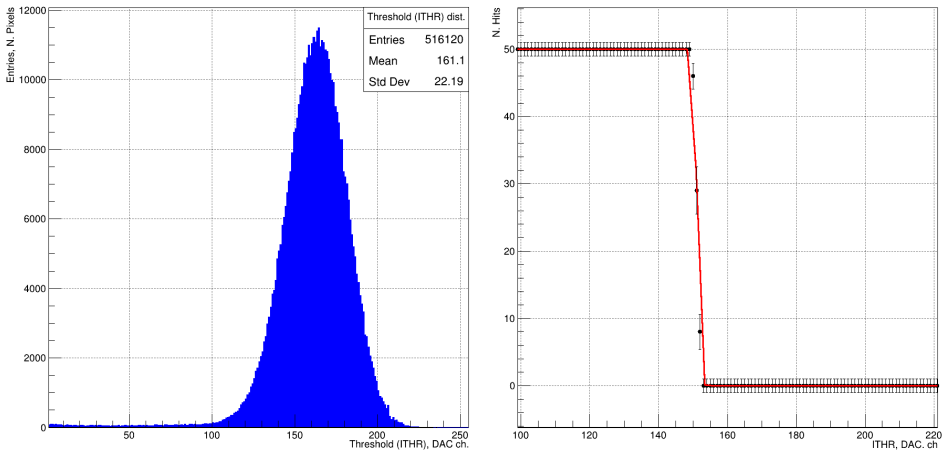
→

If the threshold distribution in units ITHR was obtained successfully fix ITHR value based on its (distribution's) form

→

Iterate over analog pulse amplitudes measuring pixel's S-curve. Check if threshold in units of charge is the desired one

Threshold (ITHR) distribution and pixel's "turning-off" curve



Threshold distribution and pixel's "turning-on" curve