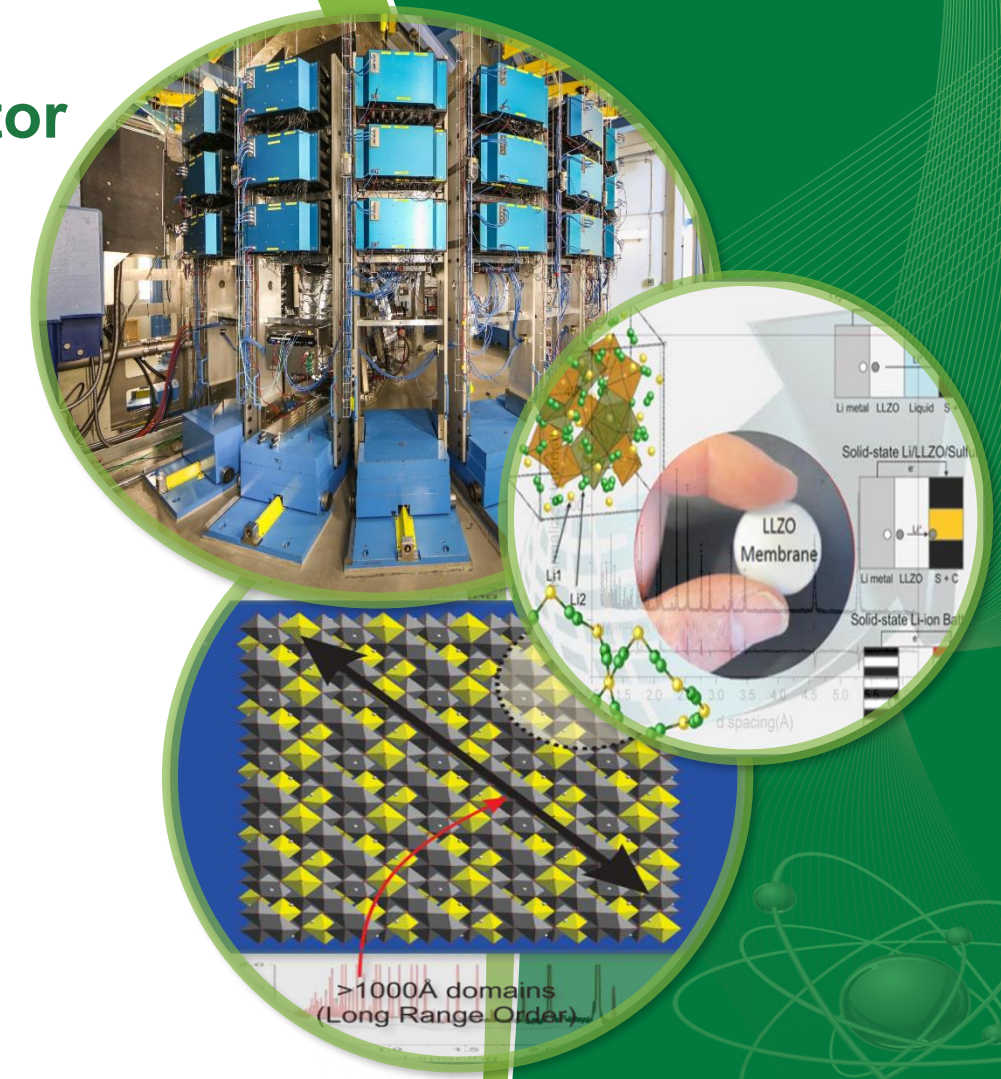
Deployment of ADTimePix3 areaDetector Driver at Neutron and X-ray Facilities

https://github.com/areaDetector/ADTimePix3

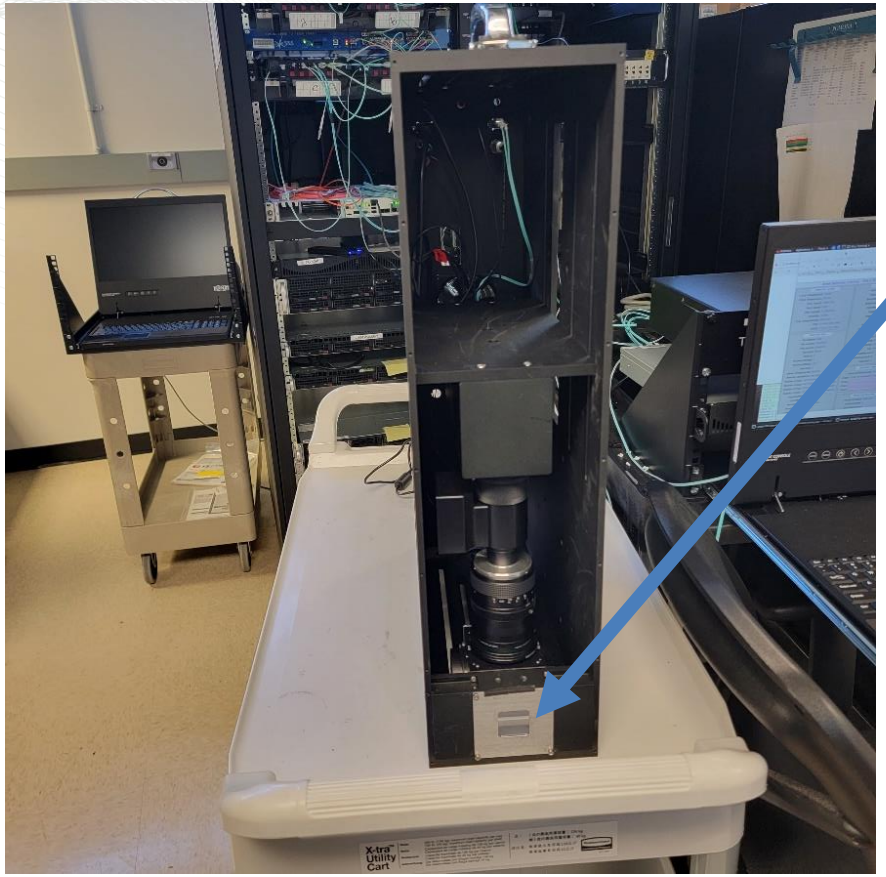
Kazimierz Gofron

ORNL



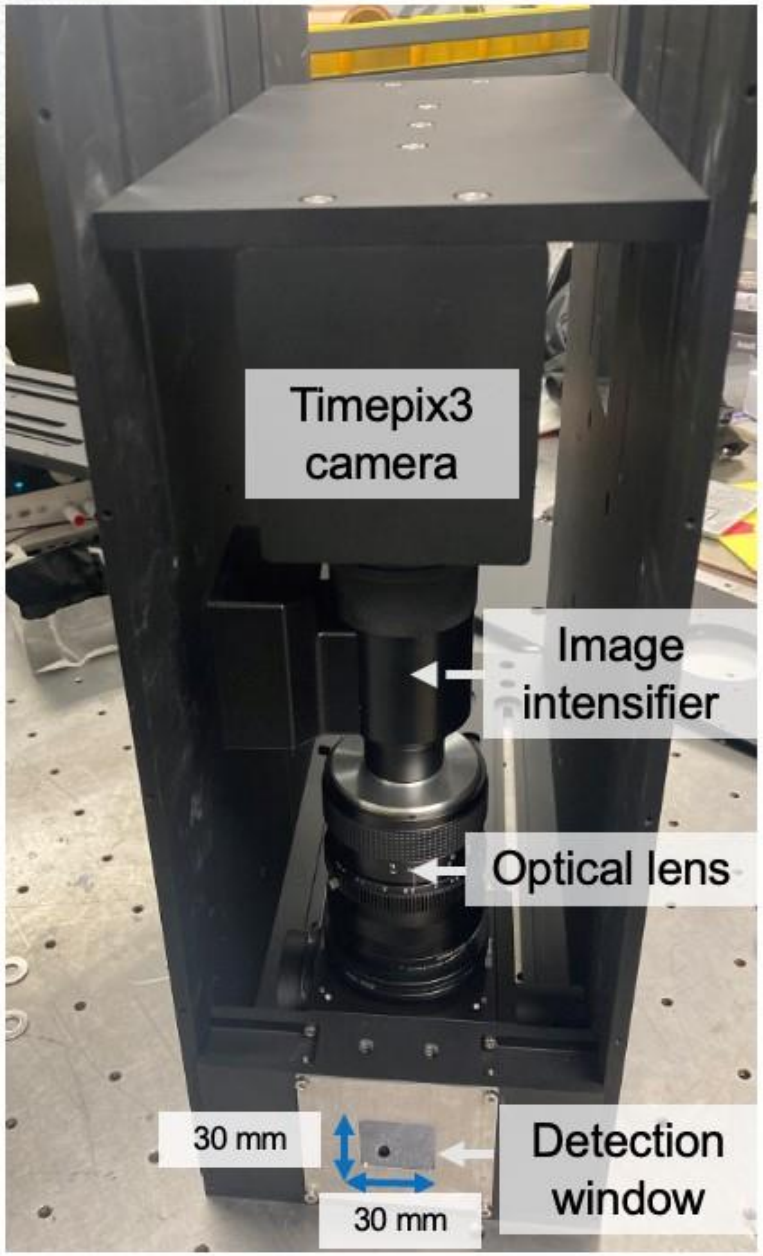
ICALEPCS 2023, Cape Town, South Africa
October 9, 2023

ASI tpx3Cam for SNS, and HFIR reactor beamlines

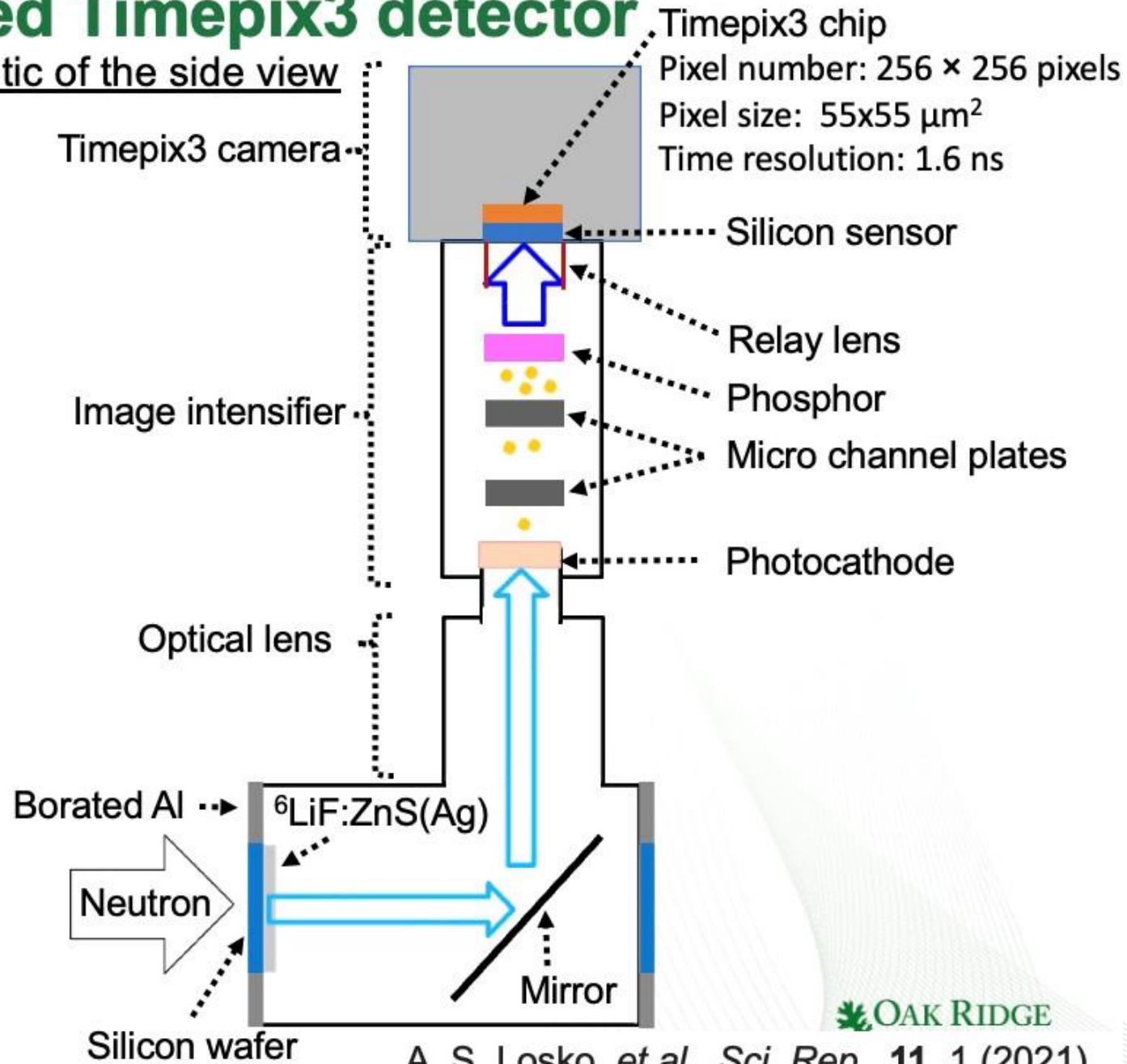


- Thermal neutron detection process
  - ^{6}Li (n + ^{4}He) ^{3}H + 4.78 [MeV] (5-9 um)
  - Scintillator containing ^{6}Li generates light converted in photocathode and amplified by dual MCP stack (intensifier).
  - Electron cluster detected by Tpx3Cam
  - No vacuum pump – sealed intensifier
  - Single chip tpx3Cam
- Alternative ^{10}B doped glass: 1st MCP glass, no scintillator
  - ^{10}B (n+ ^{4}He) ^{7}Li + 2.31 [MeV]
  - A.S. Tremsin et al. / Nuclear Instruments and Methods in Physics Research A 592 (2008) 374–384
  - Four chip in-vacuum TimePix3

Developed scintillator-based Timepix3 detector

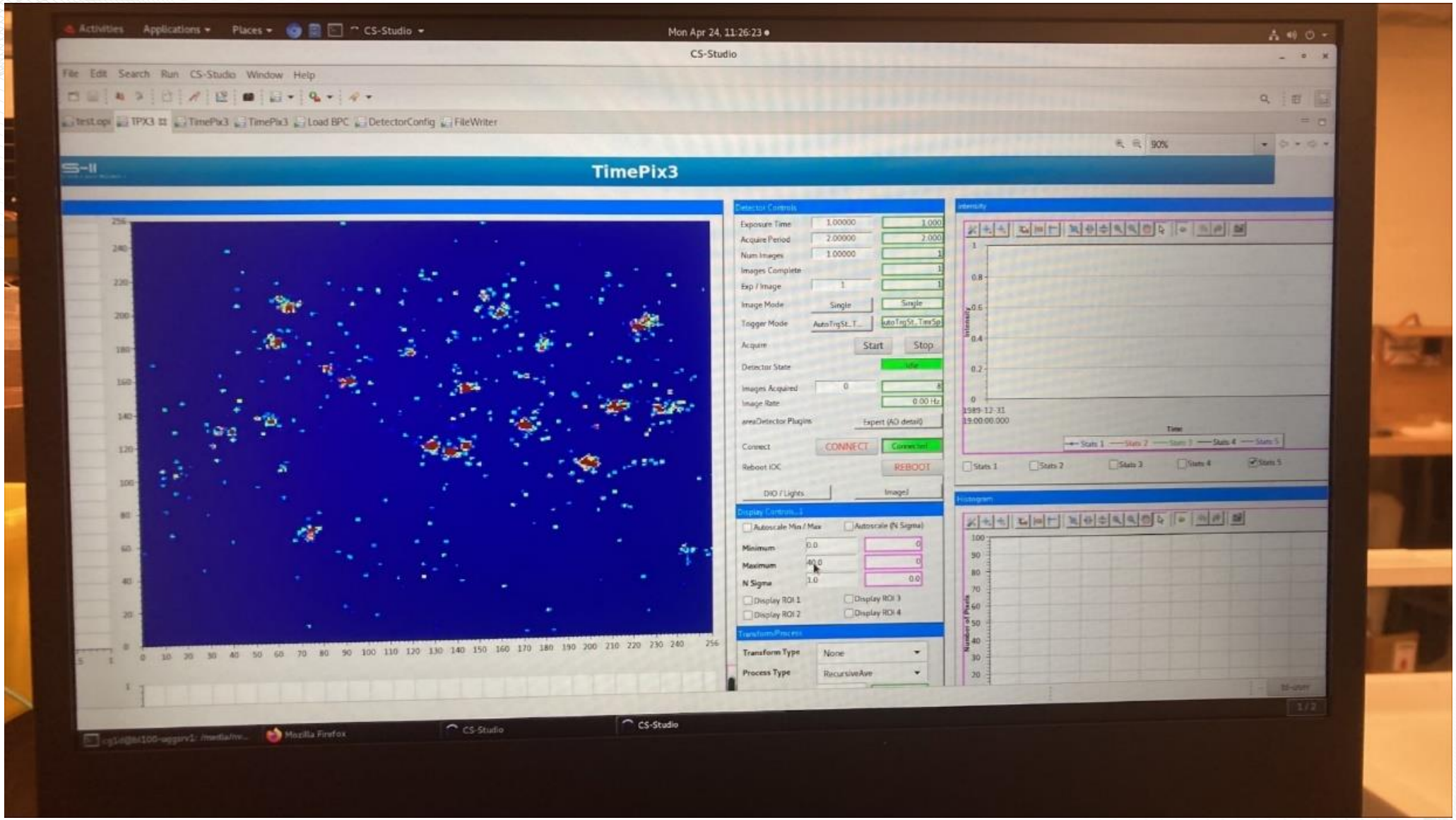


Schematic of the side view

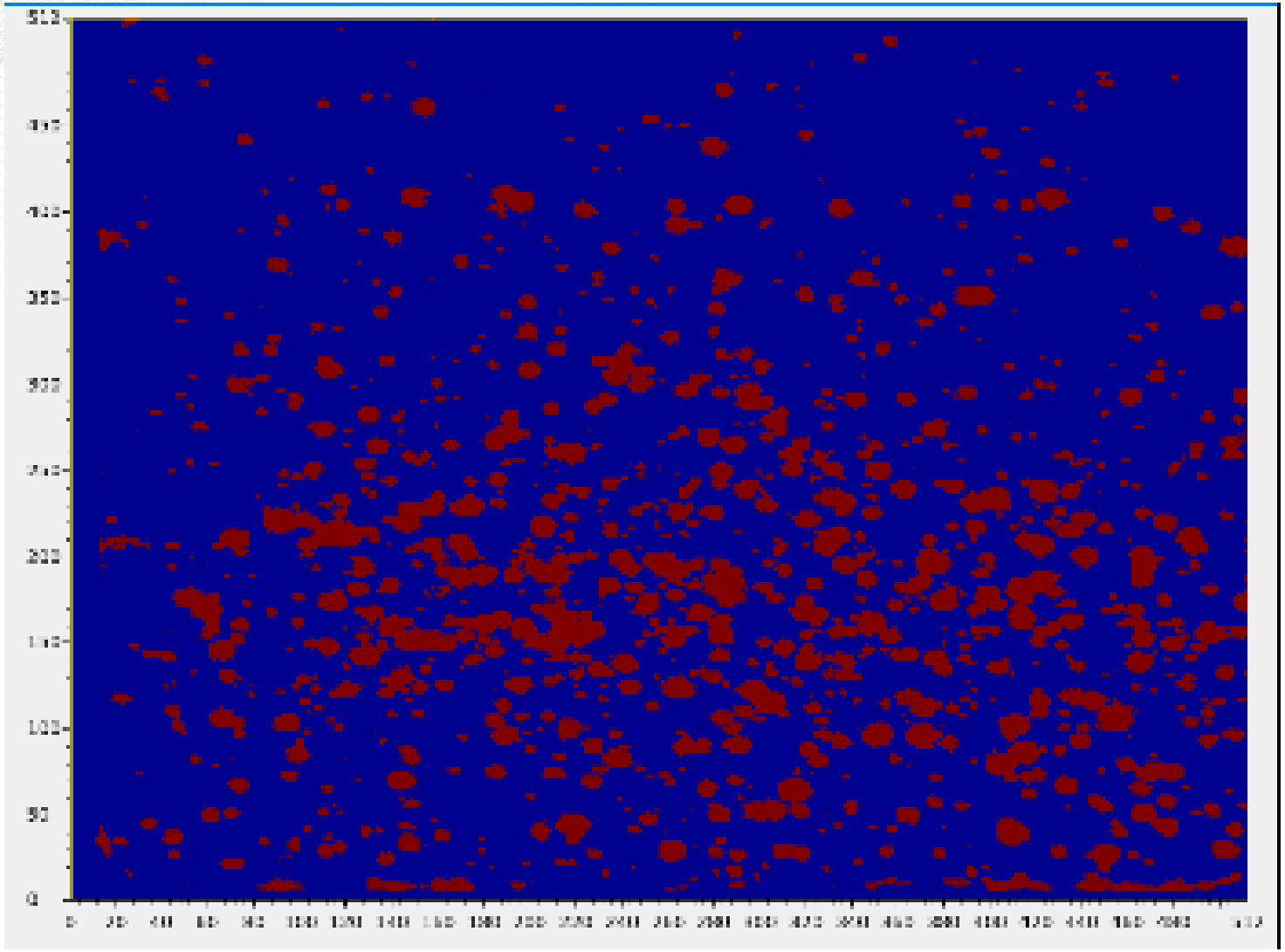


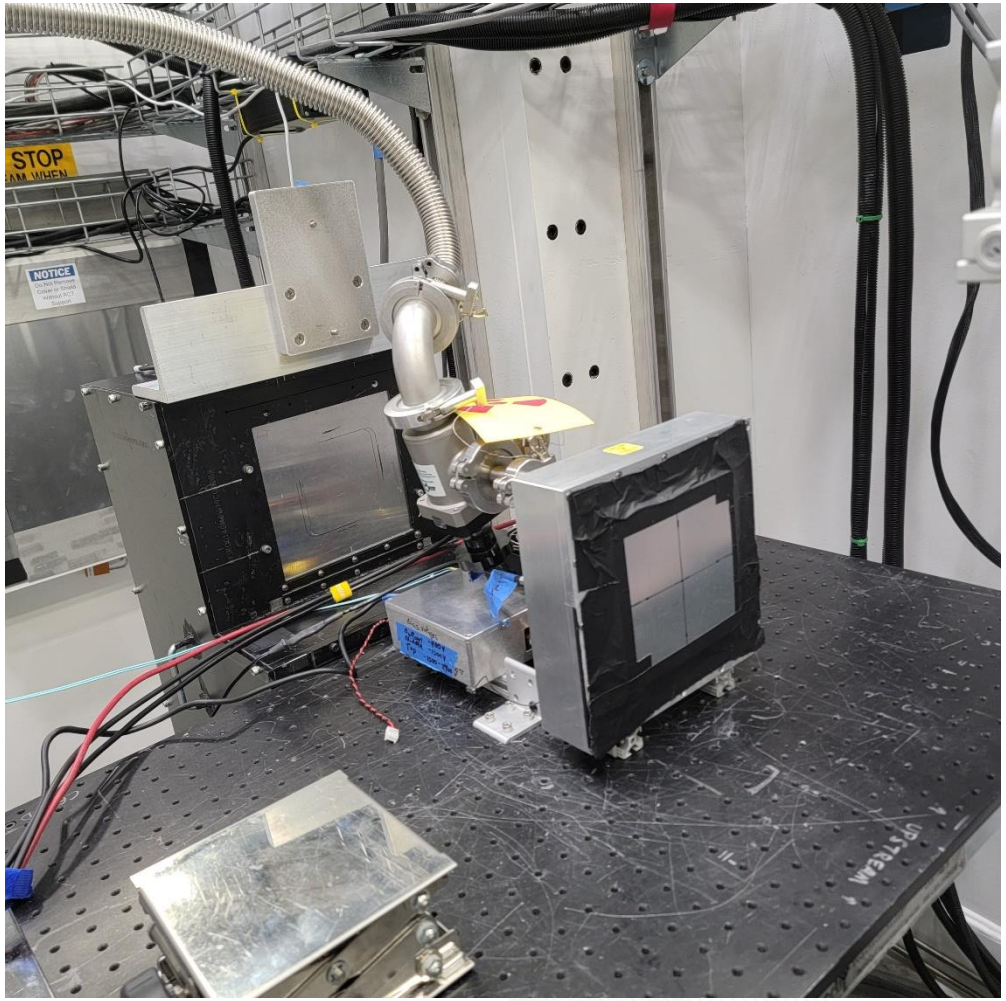
A. S. Losko, *et al.*, *Sci. Rep.* **11,** 1 (2021).

ASI tpx3Cam thermal neutron ‘clusters’ – preview



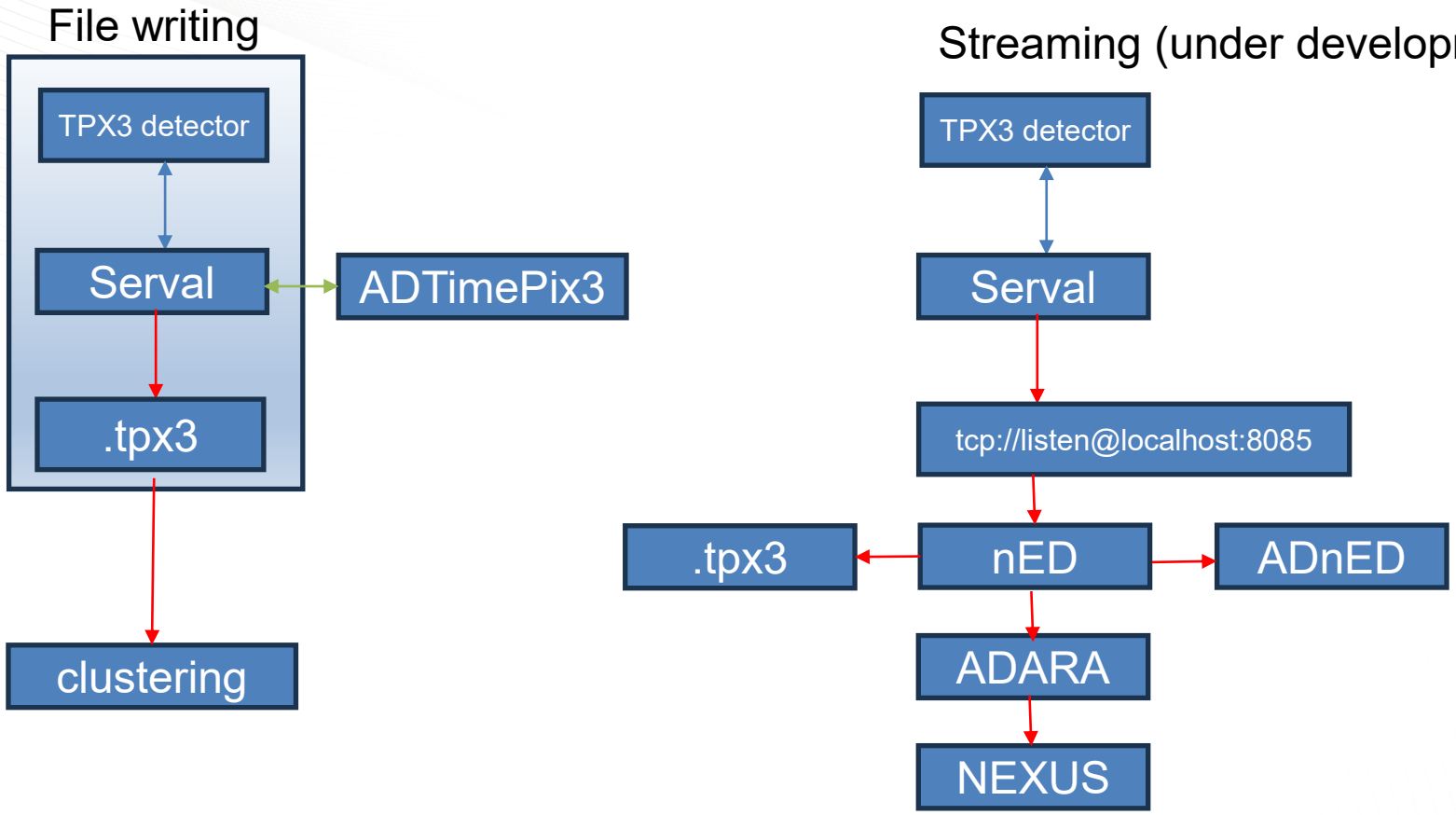
ASI tpx3Cam thermal neutron ‘clusters’: ^{10}B detector





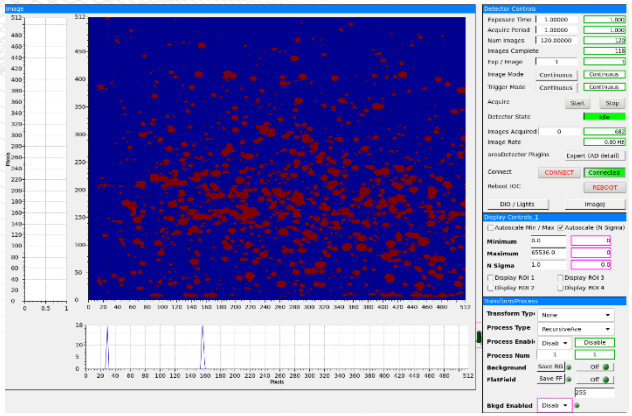
HFIR: CG-1D beamline

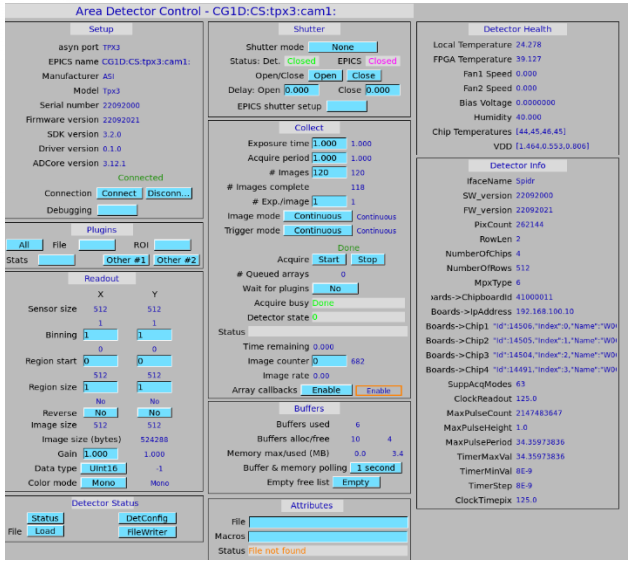
File writing and streaming of hits

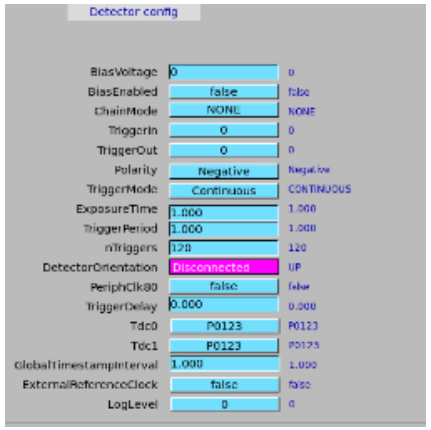


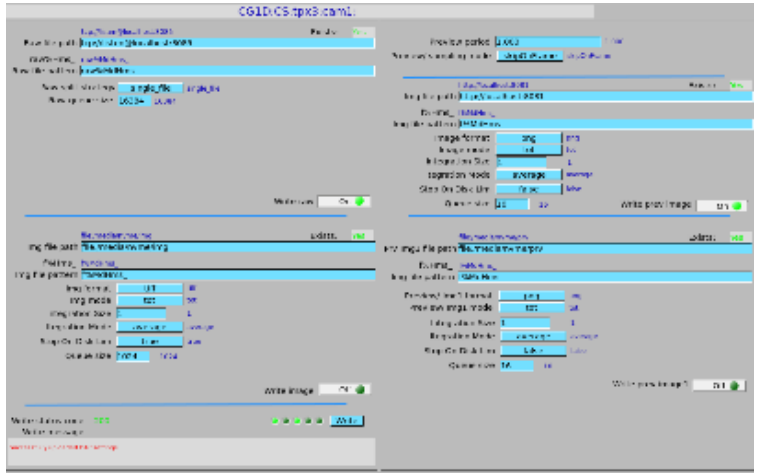
- Funama Fumiaki (tpx3Cam optical)
  - Real time data pipeline
  - Clustering
  - nED
  - ADnED
  - ADARA

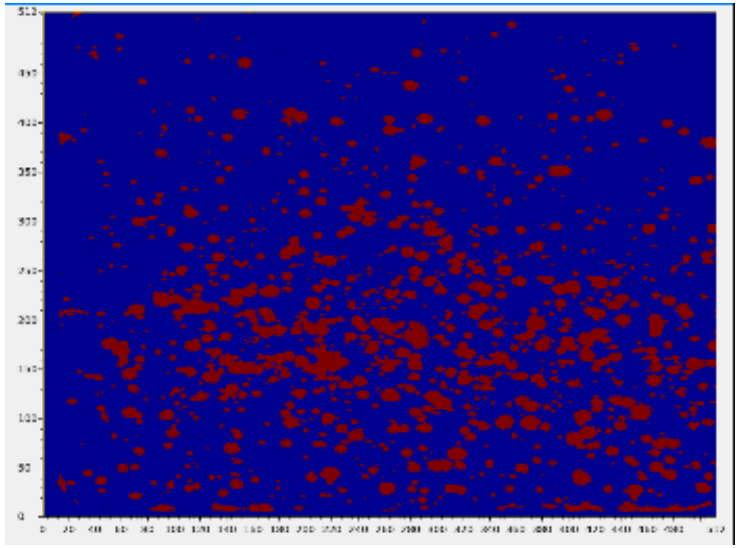
ADTimePix3 screens

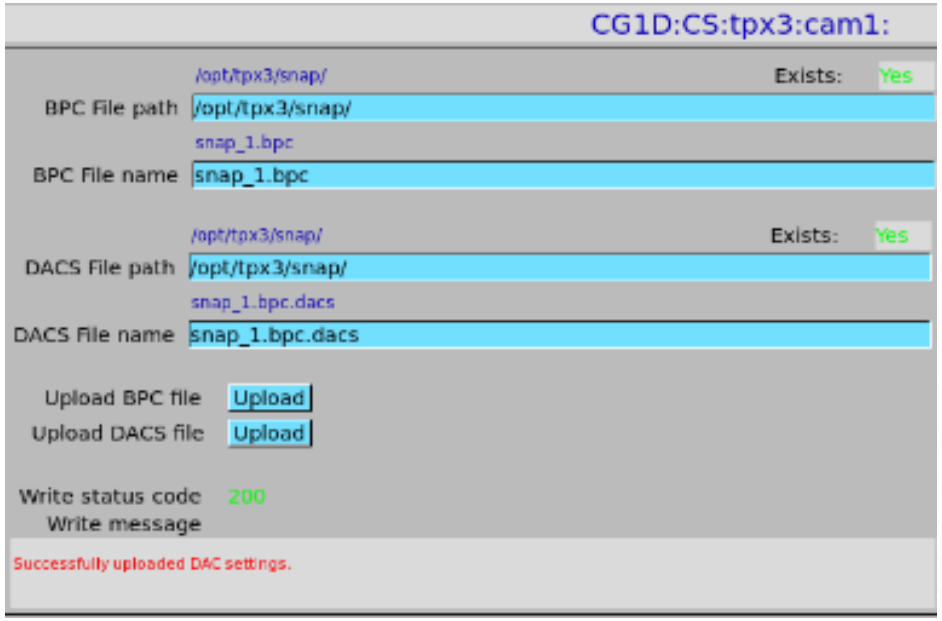


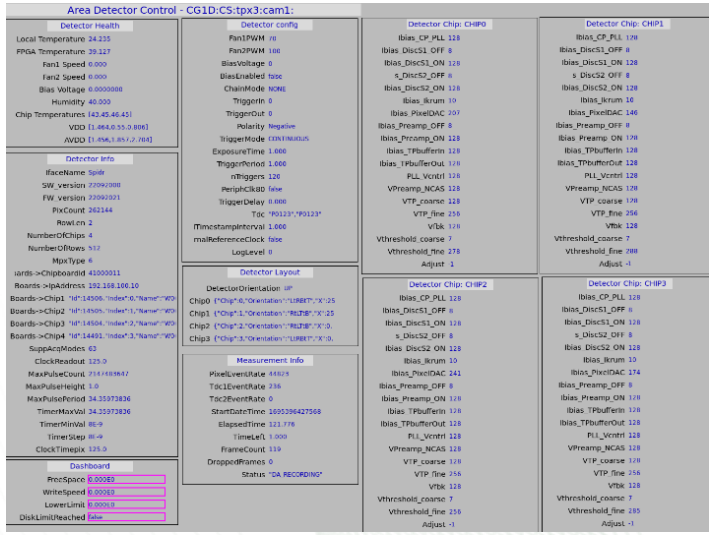








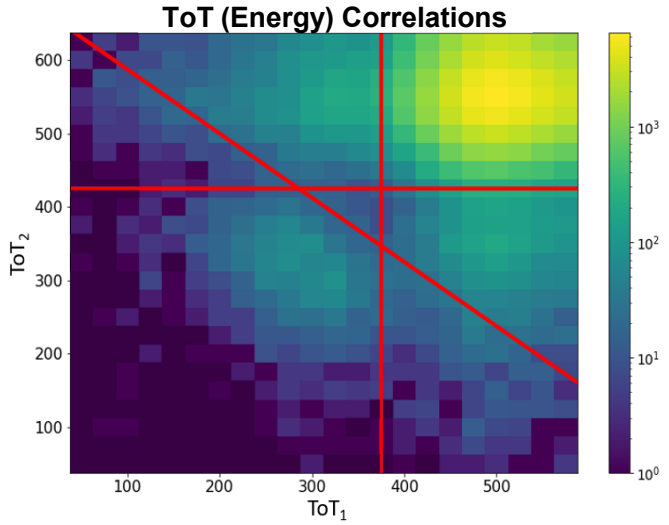


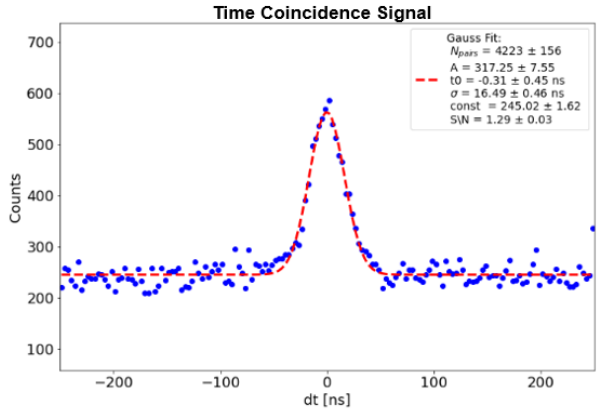


ASI TimePix3 four chip: direct detection of X-ray

- An EPICS areaDetector driver for TimePix3 quad 512 x 512 detector from ASI.
- ASC https://www.amscins.com/
- NSLS2/CHX – Coherent Hard X-ray Scattering beamline.
- TDC – Timing from accelerator (260 ps); DIO: 2 x 3 (timing signals)

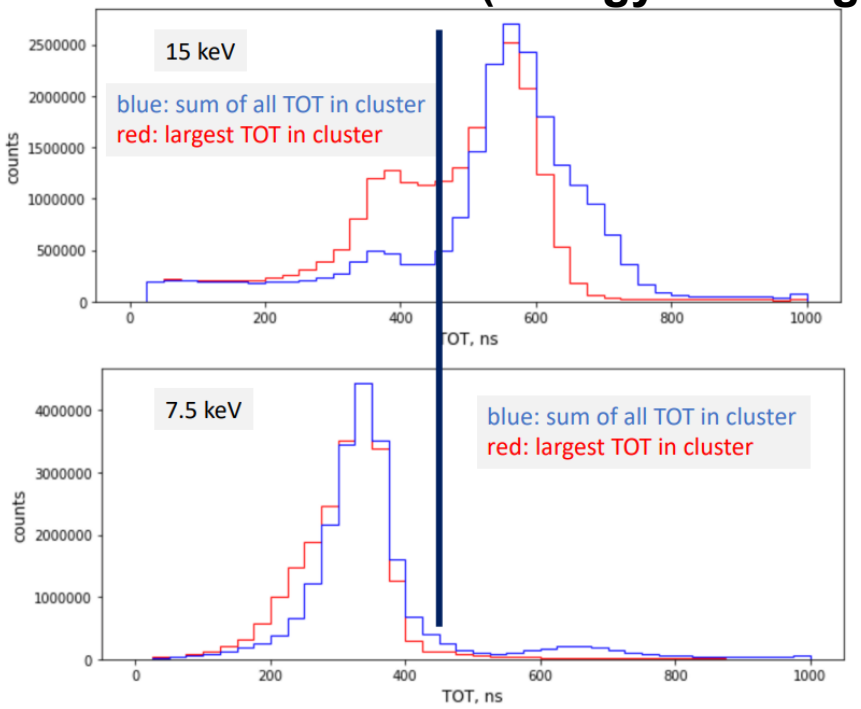
ASI TimePix3 four chip – direct detection

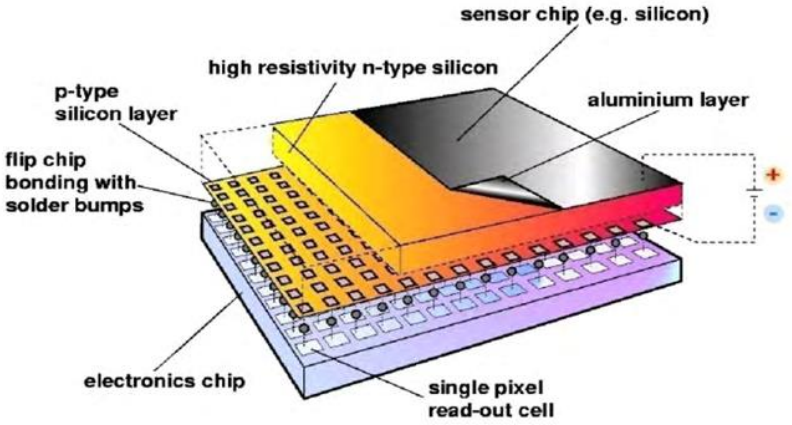




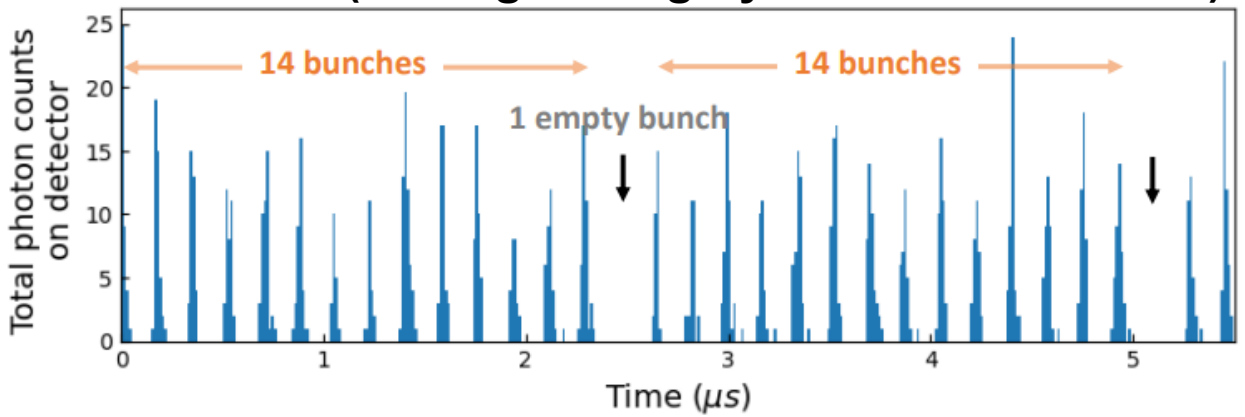
Readout	x, y, ToT, ToA
ToA precision	1.56 ns
Pixel size	55 µm x 55 µm
Pixel matrix	512 x 512 (4 chips)
Data rate	40 Mhits/s/cm^2
Energy range	2 keV – 60 keV

ToT Distributions (Energy Filtering)





Ultrafast ToA (Distinguishing Synchrotron Bunches)

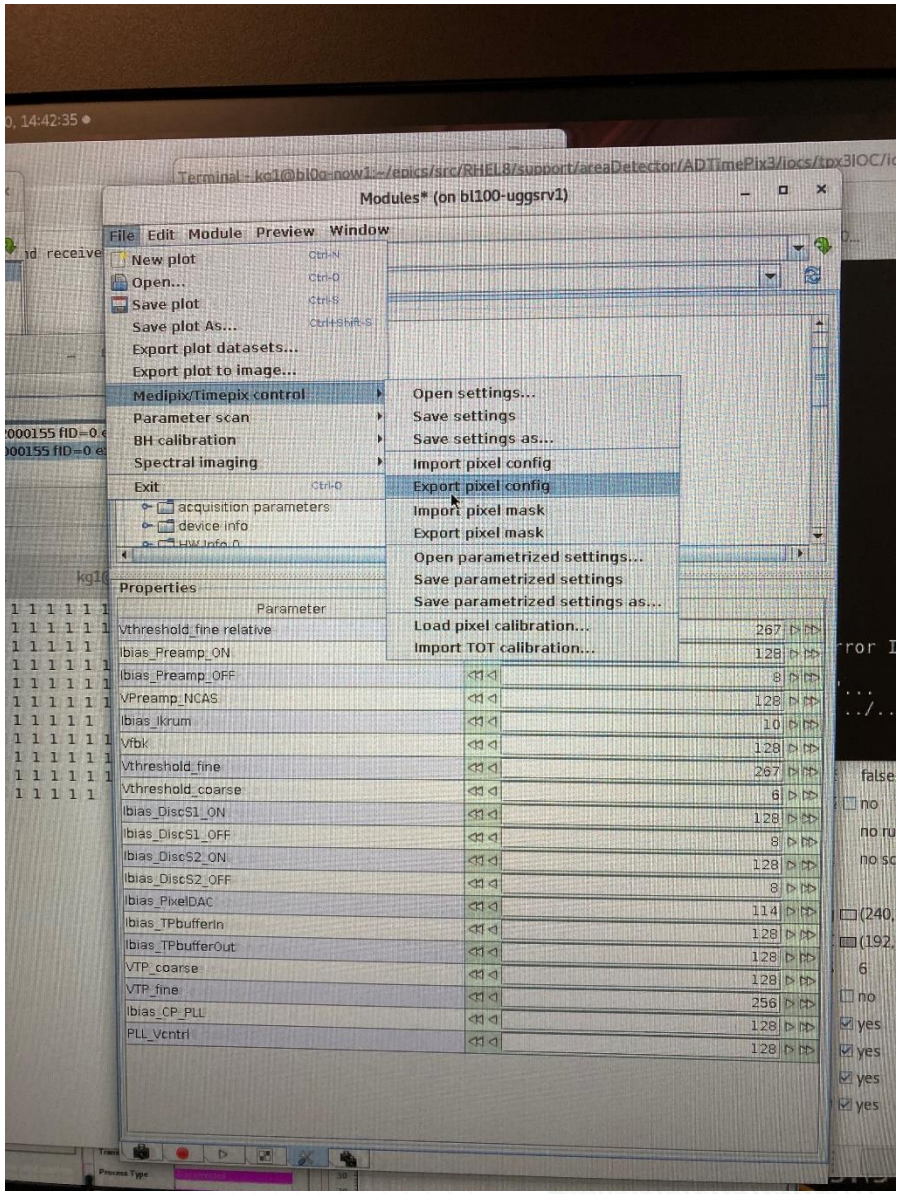


Curtesy of Justin Goodrich, Andrei Fluerasu, et. al. NSLS2

ASI TimePix3 Calibration

- SoPhy vendor software
  - DACS calibration file
  - Binary Pixel Configuration file

- ASI provides calibrations with detector



ADTimePix3 EPICS areaDetector driver

- The ADTimePix3 areaDetector driver was developed using an emulator and SERVAL software provided by Amsterdam Scientific Instruments (ASI) [9]. The Java emulator software to substitute for the real detector.
- Selected cpr [11] and json [12] software libraries as candidates for handling our representational state transfer (REST) [13] application programming interface (API) communication—commands and responses are JSON [6] format.
- Used Python cookiecutter [14] template for an EPICS areaDetector [7] driver, ADDriverTemplate [15], we created the skeleton of a driver.
- Determined the linker commands to link those dependencies into our driver
- An executable was generated, and the external dependencies were linking correctly
- Wrote the input/output controller (IOC) shell function that initializes the driver to collect some basic diagnostic information upon start-up. Once communication with SERVAL from the EPICS IOC was established, additional readback functionality and control commands were incorporated
- A REST API made it easy to break down the work into individual POST/GET requests. Each request was added to the driver, with various EPICS records tied to each field in the command and response.
- The vendor software writes binary .tpx3 data files directly to disk at a preconfigured location and only posts occasional preview images to a network location accessible by SERVAL via a GET request.

ADTimePix3 EPICS areaDetector driver (cont)

- We used a setup used by the “ADURL” areaDetector [16] driver with GraphicsMagick [17] library to read an image file from a network location into areaDetector. First, we request the preview image using the cpr library and then used the same approach to decode the image. This decoded frame is then passed along to areaDetector plugins via a callback
- The cpr GET requests use the libcurl library but do not close the connection, thereby activating many TIME_WAIT connections. Therefore, a Session feature of the cpr library was used to maintain a single connection and transport preview images in the callback method.

SERVAL tuning

- The SERVAL software requires specific IP addresses for the detector readout electronics (192.168.100.10/24) and the host computer (192.168.100.1/24).
- The communication between the readout electronics and the host uses incoming UDP ports 8192, 8193, 8194, and 8195 for detector data
- The host firewall must allow outgoing and incoming TCP traffic on port 50000. SERVAL uses this port to connect to the detector.
- The localhost http port 8081 (*ORNL specific*), which ADTimePix3 communicates to SERVAL web interface.
- Jumbo frames must be enabled, and the maximum transfer unit (MTU) must be set to 9000 on the detector's private 192.168.100.1/24 ethernet interface.
- The receive window memory size was adjusted (net.core.rmem_max = 26214400, and net.core.rmem_default = 26214400).
- The detector network interface was tuned with ASI-provided nictune.sh script (*or using the linux ethtool commands*) to minimize detector packet reordering.
- Since version 3.2, SERVAL reports UDP packet loss. SERVAL settings such as a resourcePoolSize of about 1,048,576 or 2,097,152 minimize UDP packet loss
- The ADTimePix3 detector control was operated on Ubuntu 18.04, 20.04, and 22.04, and on RHEL 7.9 and 9.0.

ADTimePix3 dependencies

- Json C++ cpr, and json library use

```
std::string config;
config = this->serverURL + std::string("/detector/config");

// Detector configuration file
r = cpr::Get(cpr::Url{config},
             cpr::Authentication{"user", "pass", cpr::AuthMode::BASIC},
             cpr::Parameters{{"anon", "true"}, {"key", "value"}});

json config_j = json::parse(r.text.c_str());
config_j["BiasVoltage"] = 103;
config_j["BiasEnabled"] = true;

//config_j["Destination"]["Raw"][0]["Base"] = "file:///home/kgofron/Downloads";
//printf("Text JSON server: %s\n", config_j.dump(3,' ', true).c_str());

r = cpr::Put(cpr::Url{config},
             cpr::Body{config_j.dump().c_str()},
             cpr::Header{{"Content-Type", "text/plain"}});

printf("Status code: %li\n", r.status_code);
printf("Text: %s\n", r.text.c_str());
```

```
},
"Config" : {
  "Fan1PWM" : 0,
  "Fan2PWM" : 0,
  "BiasVoltage" : 0,
  "BiasEnabled" : false,
  "ChainMode" : "NONE",
  "TriggerIn" : 0,
  "TriggerOut" : 0,
  "Polarity" : "Positive",
  "TriggerMode" : "PEXSTART_NEXSTOP",
  "ExposureTime" : 0.1,
  "TriggerPeriod" : 0.5,
  "nTriggers" : 0,
  "PeriphClk80" : false,
  "TriggerDelay" : 0.0,
  "Tdc" : [ "PN0123", "PN0123" ],
  "GlobalTimestampInterval" : 0.0,
  "ExternalReferenceClock" : false,
  "LogLevel" : 1
},
"Layout" : {
  "DetectorOrientation" : "UP",
  "Original" : {
    "Chips" : [ {
      "Chip" : 0,
      "X" : 256,
      "Y" : 0,
      "Orientation" : "LtRBtT"
    }, {
```

ADTimePix3 opi – disk data structure

- Detector configuration, and controls.
- Currently data written to NVME internal disk for performance
- NVME
  - Raw
  - Img
  - Prv
    - Img
    - Hst {Serval experimental version support}
- Real time data using socket – SNS/HFIR primary
- Data write rates (~ 1GB/s)
- EPICS preview channel used for live viewing
- .tpx3 data format
-

Summary

- An EPICS ADTimePix3 areaDetector driver for ASI TimePix3 was developed.
  - Uses cpr (Curl for human's library)
  - Uses json library
- X-ray, and charged particles use direct detection
- Thermal neutrons use indirect detection through fission of ^{6}Li in scintillator, or ^{10}B fission reaction
- Primary method of storing data is through .tpx3 file, and/or live stream.

Acknowledgements

- Jakub Wlodek, NSLS2, Brookhaven National Lab
- CHX/NSLS2 team
  - Andrei Fluerasu, Xiaoqian Chen, Justin C. Goodrich, NSLS2, Brookhaven National Lab
  - Andrei Nomerotski, Physics Department, Brookhaven National Lab
- ORNL
  - Vacaliuc, Bogdan, Greg Guyotte, Klemen Vodopivec, Matt Waddel, James Kohl, Zach Thurman, Alex Sobhani
  - Starra Lyons, Seth Giles
  - Funama Fumiaki, Chong Su-Ann, Zhang, Chen