DEPLOYMENT OF ADTimePix3 areaDetector DRIVER AT NEUTRON AND X-RAY FACILITIES

Kazimierz J. Gofron*,1, Bogdan Vacaliuc1, Jakub Wlodek2, Greg Guyotte1, Starra Lyons1, Seth Giles1, Fumiaki Funama1, Su-Ann Chong1
1Spallation Neutron Source and High Flux Isotope Reactor, Oak Ridge National Laboratory, Oak Ridge, TN, USA
2National Synchrotron Light Source II, Brookhaven National Laboratory, Upton, NY, USA

Abstract

TimePix3 is a 65k hybrid pixel readout chip with simultaneous time-of-arrival (ToA) and time-over-threshold (ToT) recording in each pixel. The chip operates without a trigger signal, and employs a sparse readout (i.e., only pixels containing events are read out). The flexible architecture enables achieving up to 40 MHits/s/chip readout throughput @40 MHz clock and up to 80 MHits/s/chip @80 MHz clock, using simultaneous readout and acquisition by sharing readout logic with transport logic of a super pixel matrix formed using a 2 × 4 pixel structure. The chip ToA records 1.5625 ns minimum timing resolution [1].

The X-ray and charged particle events are counted directly by the TimePix3 detector. However, indirect neutron counts use ^{6}Li fission in a scintillator matrix. The fission energy is generated by the reaction ^{6}Li(n,^{4}He)^{3}H + 4.78 MeV, where ^{6}Li is embedded in the scintillator material, such as ZnS(Ag). The fission space–charge region is limited to 5–9 µm. A photon from the scintillator material excites a photocathode electron, which is further multiplied in a dual-stack Microchannel plate (MCP) [2], with an integrated solution such as with an image intensifier tube [3].

Alternatively, a neutron event is detected in thermal neutron capture by ^{10}B doped MCP glass [4] with the nuclear fission reaction ^{10}B(n,^{4}He)^{7}Li + 2.31 MeV. The charged particle products enter microchannels of the first MCP. This design does not use a scintillator. For both thermal neutron detection designs, a neutron "event" is defined by a cluster of electron hits detected by the TimePix3 chip.

We report on the Experimental Physics Industrial Control System (EPICS) [5], areaDetector [6], and ADTimePix3 [7] driver that interfaces with the Amsterdam Scientific Instruments (ASI) [8] SERVAL software using JavaScript Object Notation (JSON) [9] objects. The Serval program controls and distributes data collected from the readout electronics. The ADTimePix3 driver directs data to storage and a real-time processing pipeline, and also configures the chip. The time-stamped data are either stored in a raw .tpx3 [8] file format or passed through a network socket for further processing of the hits. To identify individual neutrons, a clustering software module groups hits into clusters [10]. The conventional 2D images are available as image files for each exposure frame, and a preview is useful for sample alignment. The ADTimePix3 [7] areaDetector driver integrates the time-enhanced capabilities of this detector into the neutron beamline controls at the Spallation Neutron Source (SNS) and High Flux Isotope Reactor (HFIR), resulting in unprecedented time resolution.

AREA DETECTOR DRIVER

The ADTimePix3 areaDetector driver was developed using an emulator and SERVAL software provided by Amsterdam Scientific Instruments (ASI) [8]. The Java emulator executes concurrently with the Java SERVAL software to substitute for the real detector. Detector emulation allows for more efficient iterative testing with the EPICS [5] integration and provides for a simpler initial development environment.

After configuring the emulator, we identified the cpr [11] and json [12] software libraries as good candidates for handling our representational state transfer (REST) [13] application programming interface (API) communication—commands and responses are JSON [9] format—and began working on adapting these libraries to build with EPICS. This process required configuring a GNU Makefile to install the required header and library files to directories corresponding to the 64-bit x86 host architecture used for development.

Using a Python cookiecutter [14] template for an EPICS areaDetector [6] driver, ADDriverTemplate [15], we created the skeleton of a driver and determined the linker commands to link those dependencies into our driver executable so that it would compile via EPICS.

After an executable was generated and the external dependencies were linking correctly, we began to build the driver, focusing specifically on communication with the detector emulator. To begin, we 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.

Unlike many areaDetector drivers, the vendor API did not provide a callback function with direct access to a block of memory holding image data. The vendor software writes binary 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.

* gofronkj@ornl.gov

Consequently, to obtain the preview image, we used a similar setup as that used by the "ADURL" areaDetector [16] driver. ADURL uses the GraphicsMagick [17] library to read an image file from a network location into areaDetector. We modified this code to first 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.

As of this writing, we have used SERVAL 3.2 release from ASI.

TUNING

The Java SERVAL software provides a web interface which along with the cpr [11] and json [12] libraries make the setup of JSON [9] web requests simple. The GUIs consist of Control System Studio (CSS) [18] operator interface (OPI) engineering screens and Phoebus [19] OPI production screens. These interface operation with the neutron event distribution (nED) [20] and Accelerating Data Acquisition Reduction and Analysis (ADARA) [21] real-time streaming data acquisition services.

On the specific computer systems in use at the time of this work, Oak Ridge National Laboratory (ORNL) used the Redhat Enterprise Linux (RHEL) 7.9 operating system, not the vendor-recommended Ubuntu 20.04 operating system. The RHEL 7.9 system configuration required some level of tuning, which is discussed in the ADTimePix3 repository. Particularly important was memory tuning, jumbo frame setup, and the opening of firewall rules for several user datagram protocol (UDP) ports, as used for data transport on the 192.168.100.1 network interface (10 Gbps) between the detector and SERVAL. On this ORNL system, port 8081 must be used for the SERVAL web interface because SERVAL's default port 8080 is used by another process.

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 UDP ports. The host firewall must allow outgoing and incoming TCP traffic on port 50000. SERVAL uses this port to connect to the detector and port 8081 (*ORNL specific*), which ADTimePix3 communicates to over a web interface. The firewall must also allow incoming UDP ports 8192, 8193, 8194, and 8195 for detector data. 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.

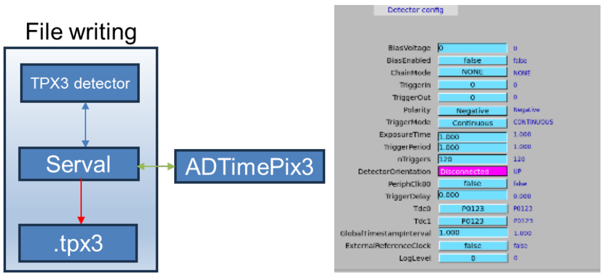
The receive window memory size was adjusted (net.core.rmem_max = 26214400, and net.core.rmem_default = 26214400). The private detector network interface was tuned with ASI-provided nictune.sh script (*or using the linux ethtool commands*). Network tuning helps minimize detector packet reordering on the private detector interface.

Since version 3.2, SERVAL reports UDP packet loss. SERVAL settings such as a resourcePoolSize of about 1,048,576 or 2,097,152 were observed to minimize UDP packet loss on a specific RHEL 7.9 server with 128 GB system memory. The SERVAL might reduce the requested resourcePoolSize and report it in the terminal window before each acquisition, which increases the acquisition startup time from the typical 4 seconds.

The ADTimePix3 detector control was operated on Ubuntu 18.04, 20.04, and 22.04, and on RHEL 7.9 and 9.2.

SOFTWARE INTEGRATION

Two modes of operation are possible: SERVAL may be configured to (1) write raw data to disk for later processing, or (2) stream hit data over a TCP/IP connection for real-time processing (Fig. 1). Currently we are writing .tpx3 raw files, and actively working on the development of a process for streaming hits to our data collection infrastructure (nED, ADnED, ADARA).



BiasVoltage	0	0
BiasEnabled	false	false
ChainMode	NONE	NONE
TriggerIn	0	0
TriggerOut	0	0
Polarity	Negative	Negative
TriggerMode	Continuous	CONTINUOUS
ExposureTime	1.000	1.000
TriggerPeriod	1.000	1.000
nTriggers	120	120
DetectorOrientation	Disconnected	UP
PeriphClk80	false	false
TriggerDelay	0.000	0.000
Tdc0	P0123	P0123
Tdc1	P0123	P0123
GlobalTimestampInterval	1.000	1.000
ExternalReferenceClock	false	false
LogLevel	0	0

Figure 1: (left) ADTimePix3 configures SERVAL for writing .tpx3 raw files. (right) Detector parameter configuration control screen.

The FileWriter OPI screen (Fig. 2 [left]) allows the user to select the data output channels. Of interest to us are the raw .tpx3 format and the preview channel when data are written to disk. The preview channel is useful for sample alignment and monitoring data collection because it provides visual feedback on the response of the detector. Only the raw .tpx3 or streaming-to-socket format contain timing information; all other image channels send data as pixel frames without timing information. Based on the OPI selections, FileWriter creates the JSON configuration, which configures SERVAL to enable specific data channels.

Many of the important detector parameters are configured from the DetectorConfig OPI screen (Fig. 1 [right]). Providing an external timing pulse, such as from a neutron chopper via the Local Time Stamp (TDC) input signal, enables the calculation of time-of-flight as referenced to a sync signal for the neutron events. Global time stamp (GDC) intervals are used for data collections up to 81 days, enabling long dataset events to be properly ordered in time.

The TimePix3 detector calibration files are typically loaded during IOC boot. The calibration can also be loaded using the Calibrate OPI screen (Fig. 3). The two calibration files are (1) a binary pixel configuration file with .bpc extension, and (2) a corresponding text Digital to Analog Converter configuration file with .dacs extension. Both calibration files are generated by the ASI-provided SoPhy software for calibrating and adjusting the detector. The

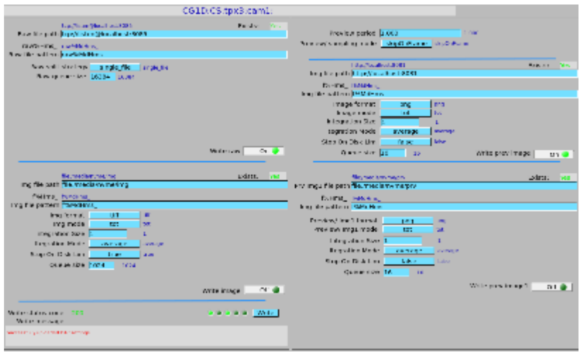

CG1D:CS:tpx3:cam1:

Figure 2: (left) FileWriter OPI screen configures data-writing channels. The OPI screen choices build a JSON configuration, which is written to SERVAL when the WriteData EPICS process variable (PV) is processed. (right) Detector parameter configuration.

files provide controls over individual pixel gain and threshold parameters, a mask of "hot" pixels, and other key per-pixel calibration values.

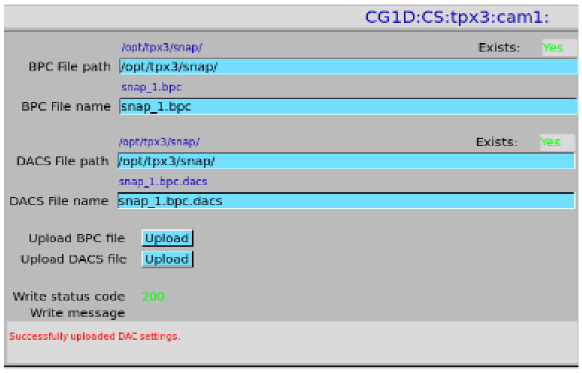

```
CG1D:CS:tpx3:cam1:

                /opt/tpx3/snap/                         Exists:   Yes
BPC File path   /opt/tpx3/snap/
                snap_1.bpc
BPC File name   snap_1.bpc

                /opt/tpx3/snap/                         Exists:   Yes
DACS File path  /opt/tpx3/snap/
                snap_1.bpc.dacs
DACS File name  snap_1.bpc.dacs

Upload BPC file   Upload
Upload DACS file  Upload

Write status code   200
    Write message
Successfully uploaded DAC settings.
```

Figure 3: Detector calibration files.

A response message from SERVAL at the bottom of the screen in Fig. 3 indicates that the calibration file and the corresponding code for JSON communication was successfully loaded.

The exposure time, acquisition time, and number of frames are set from the acquisition screen (Fig. 4). This screen allows the selection of the data collection mode. For neutron detection, we use areaDetector Continuous mode and SERVAL Continuous mode. The X-ray beamlines at the National Synchrotron Light Source II (NSLS2) use AutoTrigger mode with a specified number of images.

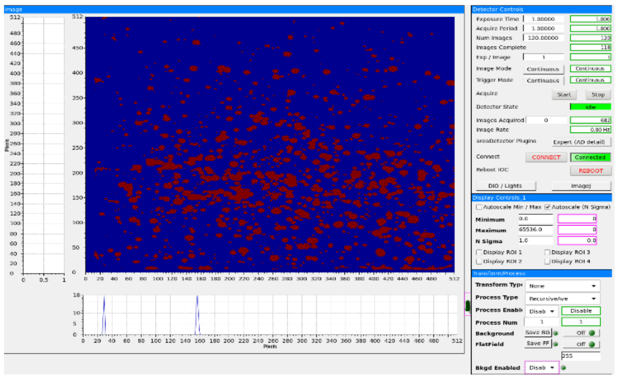

```
Detector Controls
Exposure Time        1.00000        1.000
Acquire Period       1.00000        1.000
Num Images           120.00000      120
Images Complete                     118
Exp / Image          1              1
Image Mode           Continuous     Continuous
Trigger Mode         Continuous     Continuous
Acquire              Start          Stop
Detector State                      Idle
Images Acquired      0              682
Image Rate                          0.00 Hz
areaDetector Plugins                Expert (AD detail)
Connect              CONNECT        Connected
Reboot IOC                          REBOOT
DIO / Lights                        ImageJ

Display Controls_1
☐ Autoscale Min / Max   ☑ Autoscale (N Sigma)
Minimum    0.0        0
Maximum    65536.0    0
N Sigma    1.0        0.0
☐ Display ROI 1   ☐ Display ROI 3
☐ Display ROI 2   ☐ Display ROI 4

Transform/Process
Transform Typ   None
Process Type    RecursiveAve
Process Enabl   Disab      Disable
Process Num     1          1
Background      Save BG    Off
FlatField       Save FF    Off
                           255
Bkgd Enabled    Disab
```

Figure 4: Data acquisition engineering screen.

The detector status screen (Fig. 5) shows the status of the measurement, such as hit rate, number of exposures, and acquisition time remaining. It also shows the parameters related to the health of the detector such as the temperature of the electronic components and the fan speed.

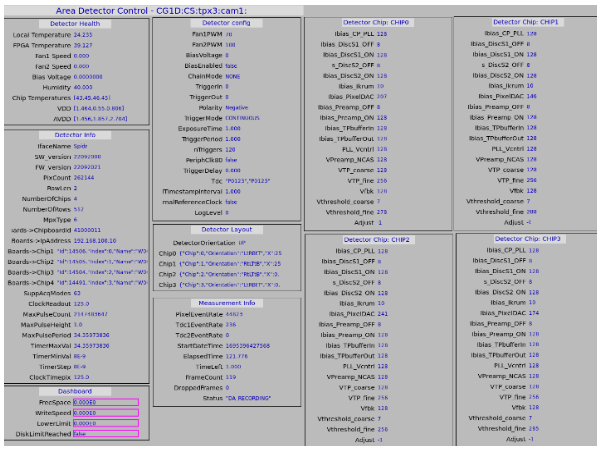

```
Area Detector Control - CG1D:CS:tpx3:cam1:

Detector Health
   Local Temperature 24.235
    FPGA Temperature 39.127
          Fan1 Speed 0.000
          Fan2 Speed 0.000
        Bias Voltage 0.0000000
            Humidity 40.000
   Chip Temperatures [43.45.46.45]
                 VDD [1.464.0.55.0.806]
                AVDD [1.456,1.857,2.704]

Detector Info
           IfaceName Spidr
          SW_version 22092000
          FW_version 22092021
            PixCount 262144
              RowLen 2
        NumberOfChips 4
       NumberOfRows 512
             MpxType 6
 ards->ChipboardId 41000011
 Boards->IpAddress 192.168.100.10
   Boards->Chip1 "Id":14506,"Index":0,"Name":"W0
   Boards->Chip2 "Id":14505,"Index":1,"Name":"W0
   Boards->Chip3 "Id":14504,"Index":2,"Name":"W0
   Boards->Chip4 "Id":14491,"Index":3,"Name":"W0
       SuppAcqModes 63
       ClockReadout 125.0
      MaxPulseCount 2147483647
     MaxPulseHeight 1.0
     MaxPulsePeriod 34.35973836
        TimerMaxVal 34.35973836
        TimerMinVal 8E-9
          TimerStep 8E-9
        ClockTimepix 125.0

Dashboard
          FreeSpace 0.000E0
         WriteSpeed 0.000E0
         LowerLimit 0.000E0
   DiskLimitReached false

Detector config
            Fan1PWM 70
            Fan2PWM 100
        BiasVoltage 0
        BiasEnabled false
          ChainMode NONE
          TriggerIn 0
         TriggerOut 0
           Polarity Negative
        TriggerMode CONTINUOUS
       ExposureTime 1.000
      TriggerPeriod 1.000
          nTriggers 120
        PeriphClk80 false
       TriggerDelay 0.000
                Tdc "P0123","P0123"
 lTimestampInterval 1.000
 rnalReferenceClock false
           LogLevel 0

Detector Layout
 DetectorOrientation UP
 Chip0 {"Chip":0,"Orientation":"LtRBtT","X":25
 Chip1 {"Chip":1,"Orientation":"RtLTtB","X":25
 Chip2 {"Chip":2,"Orientation":"RtLTtB","X":0,
 Chip3 {"Chip":3,"Orientation":"LtRBtT","X":0,

Measurement Info
     PixelEventRate 44823
    Tdc1EventRate 236
    Tdc2EventRate 0
     StartDateTime 1695396427568
       ElapsedTime 121.776
          TimeLeft 1.000
        FrameCount 119
     DroppedFrames 0
            Status "DA_RECORDING"

Detector Chip: CHIP0
        Ibias_CP_PLL 128
   Ibias_DiscS1_OFF 8
    Ibias_DiscS1_ON 128
       s_DiscS2_OFF 8
    Ibias_DiscS2_ON 128
        Ibias_Ikrum 10
     Ibias_PixelDAC 207
   Ibias_Preamp_OFF 8
    Ibias_Preamp_ON 128
  Ibias_TPbufferIn 128
 Ibias_TPbufferOut 128
         PLL_Vcntrl 128
      VPreamp_NCAS 128
         VTP_coarse 128
           VTP_fine 256
               Vfbk 128
   Vthreshold_coarse 7
     Vthreshold_fine 278
              Adjust -1

Detector Chip: CHIP1
        Ibias_CP_PLL 128
   Ibias_DiscS1_OFF 8
    Ibias_DiscS1_ON 128
       s_DiscS2_OFF 8
    Ibias_DiscS2_ON 128
        Ibias_Ikrum 10
     Ibias_PixelDAC 146
   Ibias_Preamp_OFF 8
    Ibias_Preamp_ON 128
  Ibias_TPbufferIn 128
 Ibias_TPbufferOut 128
         PLL_Vcntrl 128
      VPreamp_NCAS 128
         VTP_coarse 128
           VTP_fine 256
               Vfbk 128
   Vthreshold_coarse 7
     Vthreshold_fine 288
              Adjust -1

Detector Chip: CHIP2
        Ibias_CP_PLL 128
   Ibias_DiscS1_OFF 8
    Ibias_DiscS1_ON 128
       s_DiscS2_OFF 8
    Ibias_DiscS2_ON 128
        Ibias_Ikrum 10
     Ibias_PixelDAC 241
   Ibias_Preamp_OFF 8
    Ibias_Preamp_ON 128
  Ibias_TPbufferIn 128
 Ibias_TPbufferOut 128
         PLL_Vcntrl 128
      VPreamp_NCAS 128
         VTP_coarse 128
           VTP_fine 256
               Vfbk 128
   Vthreshold_coarse 7
     Vthreshold_fine 256
              Adjust -1

Detector Chip: CHIP3
        Ibias_CP_PLL 128
   Ibias_DiscS1_OFF 8
    Ibias_DiscS1_ON 128
       s_DiscS2_OFF 8
    Ibias_DiscS2_ON 128
        Ibias_Ikrum 10
     Ibias_PixelDAC 174
   Ibias_Preamp_OFF 8
    Ibias_Preamp_ON 128
  Ibias_TPbufferIn 128
 Ibias_TPbufferOut 128
         PLL_Vcntrl 128
      VPreamp_NCAS 128
         VTP_coarse 128
           VTP_fine 256
               Vfbk 128
   Vthreshold_coarse 7
     Vthreshold_fine 285
              Adjust -1
```

Figure 5: Detector status screen.

The detailed areaDetector screen for the ADTimePix3 detector is shown in Fig. 6. The screen allows acquisition parameters such as exposure, acquisition mode, and number of frames to be collected. The acquisition can be started using the Start/Stop PV action buttons.

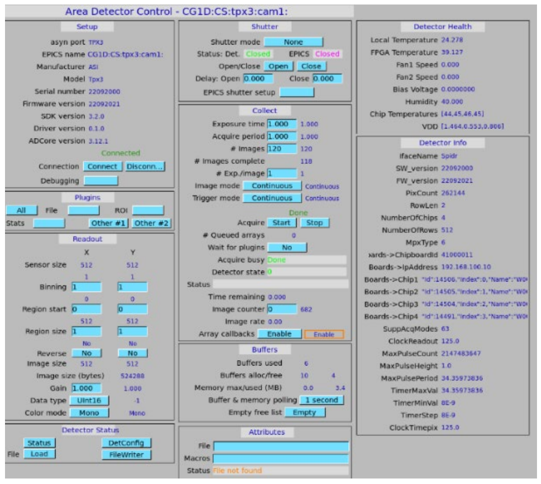

```
Area Detector Control - CG1D:CS:tpx3:cam1:

Setup
        asyn port TPX3
       EPICS name CG1D:CS:tpx3:cam1:
     Manufacturer ASI
            Model Tpx3
    Serial number 22092000
 Firmware version 22092021
      SDK version 3.2.0
   Driver version 0.1.0
   ADCore version 3.12.1
                  Connected
       Connection Connect  Disconn...
        Debugging

Plugins
 All   File        ROI
 Stats      Other #1  Other #2

Readout
                    X        Y
 Sensor size        512      512
                    1        1
 Binning            1        1
                    0        0
 Region start       0        0
                    512      512
 Region size        1        1
                    No       No
 Reverse            No       No
 Image size         512      512
 Image size (bytes)          524288
 Gain               1.000    1.000
 Data type          UInt16   -1
 Color mode         Mono     Mono

Detector Status
      Status     DetConfig
 File Load       FileWriter

Shutter
      Shutter mode  None
       Status: Det. Closed   EPICS Closed
        Open/Close  Open     Close
       Delay: Open  0.000    Close 0.000
 EPICS shutter setup

Collect
      Exposure time 1.000    1.000
     Acquire period 1.000    1.000
           # Images 120      120
  # Images complete          118
       # Exp./image 1        1
         Image mode Continuous  Continuous
       Trigger mode Continuous  Continuous
                             Done
            Acquire Start    Stop
    # Queued arrays          0
   Wait for plugins No
       Acquire busy Done
     Detector state 0
 Status
     Time remaining 0.000
      Image counter 0        682
         Image rate 0.00
    Array callbacks Enable   Enable

Buffers
        Buffers used         6
   Buffers alloc/free        10     4
 Memory max/used (MB)        0.0    3.4
 Buffer & memory polling  1 second
      Empty free list  Empty

Attributes
   File
 Macros
 Status File not found

Detector Health
   Local Temperature 24.278
    FPGA Temperature 39.127
          Fan1 Speed 0.000
          Fan2 Speed 0.000
        Bias Voltage 0.0000000
            Humidity 40.000
   Chip Temperatures [44.45.46.45]
                 VDD [1.464.0.553,0.806]

Detector Info
           IfaceName Spidr
          SW_version 22092000
          FW_version 22092021
            PixCount 262144
              RowLen 2
        NumberOfChips 4
       NumberOfRows 512
             MpxType 6
 ards->ChipboardId 41000011
 Boards->IpAddress 192.168.100.10
   Boards->Chip1 "Id":14506,"Index":0,"Name":"W0
   Boards->Chip2 "Id":14505,"Index":1,"Name":"W0
   Boards->Chip3 "Id":14504,"Index":2,"Name":"W0
   Boards->Chip4 "Id":14491,"Index":3,"Name":"W0
       SuppAcqModes 63
       ClockReadout 125.0
      MaxPulseCount 2147483647
     MaxPulseHeight 1.0
     MaxPulsePeriod 34.35973836
        TimerMaxVal 34.35973836
        TimerMinVal 8E-9
          TimerStep 8E-9
        ClockTimepix 125.0
```

Figure 6: Detailed areaDetector screen.

Integration of the ADTimePix3 driver into the beamline run control system at HFIR was accomplished primarily through the development of a proxy IOC known as the "Timepix Gateway". This proxy stands between the beamline experiment run control system and the ADTimePix3 driver, translating run requests made by the scientific user into specific configurations of various parameters within the ADTimePix3 driver. As illustrated in the above figures, configuration of ADTimePix3 is achieved entirely through standard EPICS channel access protocol, via EPICS PV's exposed by the ADTimePix3. The Timepix Gateway IOC is responsible for interacting with this PV interface to configure the detector according to the needs of the given experiment. This proxy IOC succeeds in abstracting the deep technical details and complexity from the beamline user, who need not directly access or understand the detector configuration screens.

RESULTS

ADTimePix3 is used to control data acquisition at High Flux Isotope Reactor (HFIR) and National Synchrotron Light Source 2 (NSLS2) user facilities. The X-ray facilities use direct detection: X-rays directly interact with the Si depletion layer, creating a hit equivalent to an event. Thus, a commercial ASI-provided detector is used for X-rays.

Thermal neutron detection requires indirect detection, which involves the conversion of a neutron into energetic charged particles and/or photons. The thermal neutron is converted into a photon in a scintillator, or else a charged particle from nuclear fission enters the microchannel of the first MCP. For scintillator design, the photon further excites a photoelectron of a photocathode, and then the photoelectron is multiplied by the MCP. The accelerated charge cloud impacts the TimePix3 detector, creating multiple hits—a cluster, interpreted as a neutron event.

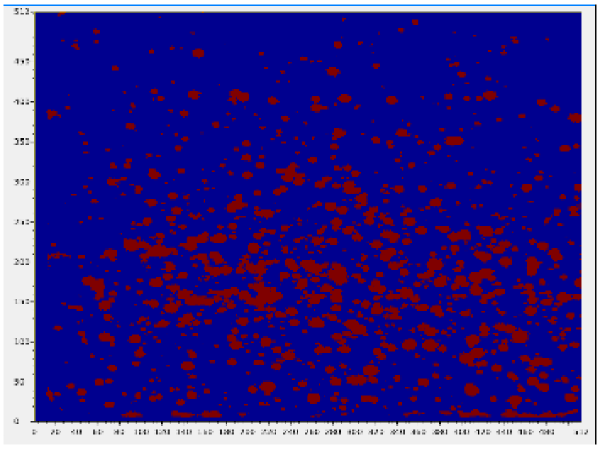


Figure 7: Preview of the neutron-generated clusters on the TimePix3 quad-detector.

At ORNL, we used two different TimePix3 detector designs. In one of the designs, a scintillator is outside of the vacuum of the MCP stack, and photons are reimaged onto the photocathode element of the MCP stack. In the other design, the scintillator and neutron are converted directly in the first MCP element with ^{10}B embedded in MCP material. The image from this type of detector is shown in Fig. 7.

The image of Fig. 7 was collected at the Multimodal Advanced Radiography Station (MARS) instrument at HFIR. Each cluster of hits represents a neutron event. The data were collected in ADTimePix3 continuous mode with 1 s exposure per frame. A Cadmium chopper was used for time-of-flight and time-over-threshold data collection modes. The chopper had two narrow opposing (180°) slots, and an optical sensor provided two trigger signals per chopper disk rotation. The trigger signal was passed to the Spidr readout TDC input and the ORNL data system processor that establishes timing for the data collection framework.

SUMMARY

An EPICS interface was developed and tuned to collect data from a range of ASI detectors that provide excellent space and time resolution.

Using ADTimePix3 enables EPICS instrument control systems to quickly integrate data collection with instrument synchronization. We have documented and referenced the relevant considerations needed for successful operation and provided references to the open-source software needed to assemble working acquisition systems.

ACKNOWLEDGEMENTS

We acknowledge Alex Sobhani's help on the visualization setup for TimePix3 detectors. We acknowledge Klemen Vodopivec, Matt Waddel, Zach Thurman, and James Kohl for their work on the streaming acquisition software that uses the ADTimePix3 IOC. Erik Maddox's help on detector inner working is acknowledged.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://www.energy.gov/doe-public-access-plan).

REFERENCES

[1] T. Poikela *et al.*, "Timepix3: a 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout", *J. Instrum.*, vol. 9, p. C05013, 2014. doi:10.1088/1748-0221/9/05/C05013

[2] https://en.wikipedia.org/wiki/Microchannel_plate_detector

[3] https://www.photonis.com/products/image-intensifier-tube

[4] A.S. Tremsin *et al.*, "On the possibility to image thermal and cold neutron with sub-15 µm spatial resolution", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 592, pp. 374–384, 2008. doi:10.1016/j.nima.2008.03.116

[5] https://epics-controls.org/

[6] https://github.com/areaDetector

[7] https://github.com/areaDetector/ADTimePix3

[8] Amsterdam Scientific Instruments, online: https://www.amscins.com/

[9] https://en.wikipedia.org/wiki/JSON

[10] C. Zhang and Z. Morgan, "Advanced Image Reconstruction for MCP Detector in Event Mode", *Commun. Comput. Inf. Sci.*, vol. 1512, pp 383–397. 2022. doi:10.1007/978-3-030-96498-6_22 supplementary: https://github.com/ornlneutronimaging/mcpevent2hist

[11] https://github.com/libcpr/cpr

[12] https://github.com/nlohmann/json

[13] https://en.wikipedia.org/wiki/REST

[14] https://github.com/cookiecutter/cookiecutter

[15] https://github.com/jwlodek/ADDriverTemplate

[16] https://github.com/areaDetector/ADURL

[17] http://www.graphicsmagick.org/

[18] https://controlsystemstudio.org/

[19] https://controlssoftware.sns.ornl.gov/css_phoebus/

[20] K. Vodopivec and B. Vacaliuc, "High Throughput Data Acquisition with EPICS", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 213-217. doi:10.18429/JACoW-ICALEPCS2017-TUBPA05

[21] G. Shipman *et al.*, "Accelerating Data Acquisition, Reduction, and Analysis at the Spallation Neutron Source," *2014 IEEE Int. Conf. .e-Sci.*, pp. 223-230, 2014. doi 10.1109/eScience.2014.31