areaDetector: What’s New?

Mark Rivers

GeoSoilEnviroCARS, Advanced Photon Source

University of Chicago

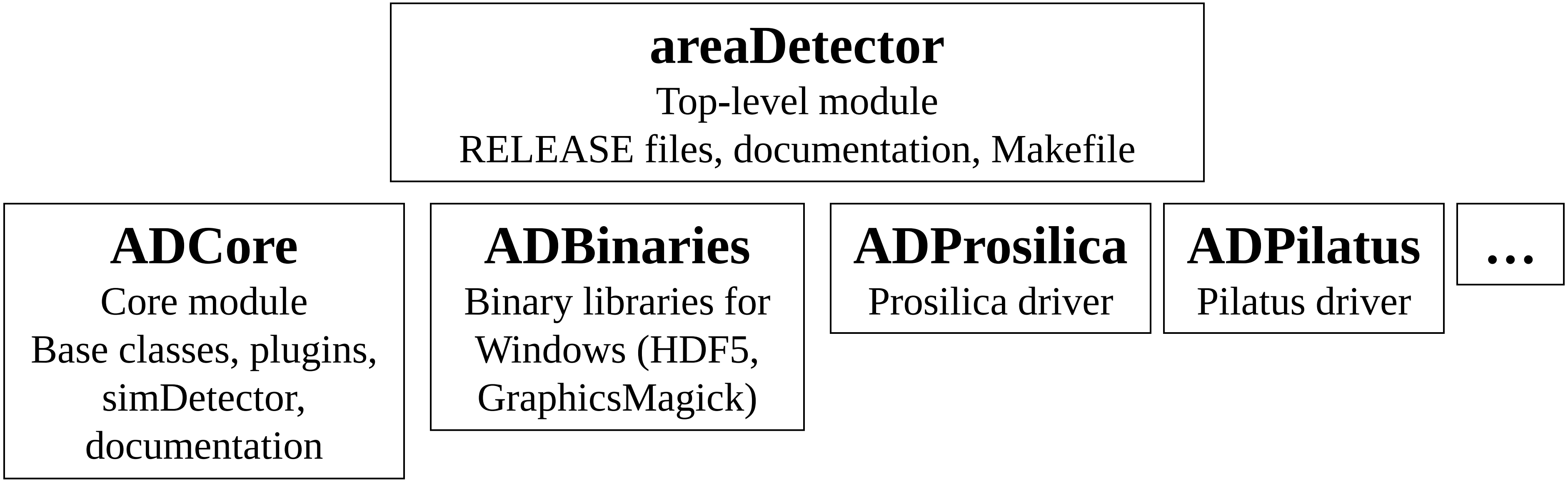
Outline

- Last TWG areaDetector talk was 2009
  - Won’t try cover everything since then!
- Changes in the last year, R2-0, R2-1, R2-2
- New and improved drivers
- New and improved plugins
- Future plans

Moved areaDetector to github

- areaDetector was getting too big.
  - New releases being held up waiting for testing on one detector types, etc.
- Hard to collaborate with other sites using APS Subversion repository
  - git and github provide much better tools for multi-site collaborations
- Moved in December 2013, with R2-0 release in April 2014
- Split into 3 “core” repositories, and separate repositories for each detector

New Organization



areaDetector
Top-level module
RELEASE files, documentation, Makefile

ADCore
Core module
Base classes, plugins, simDetector, documentation

ADBinaries
Binary libraries for Windows (HDF5, GraphicsMagick)

ADProsilica
Prosilica driver

ADPilatus
Pilatus driver

…

– Each box above is a separate git repository

– Can be released independently

– Hosted at http://github.com/areaDetector project

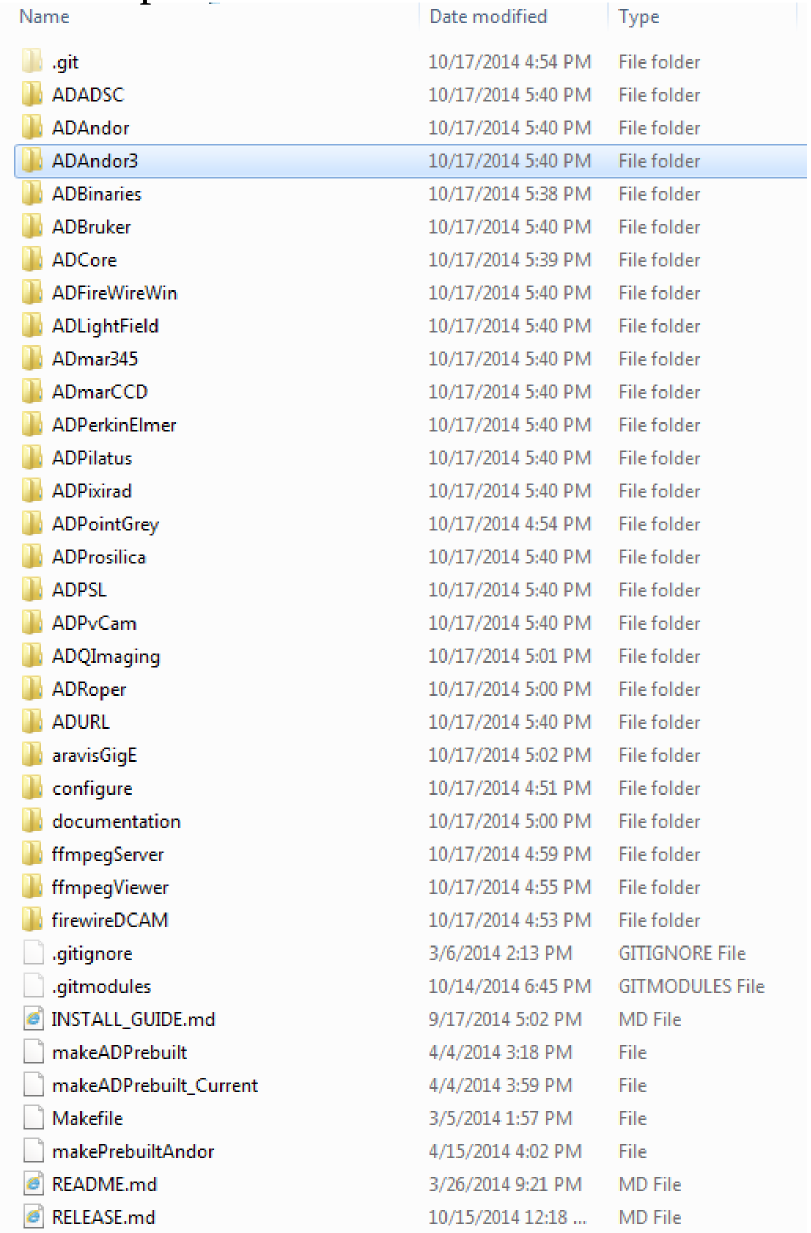
– Each repository is a submodule under areaDetector/areaDetector

– Can clone individual repositories, or clone the entire project with

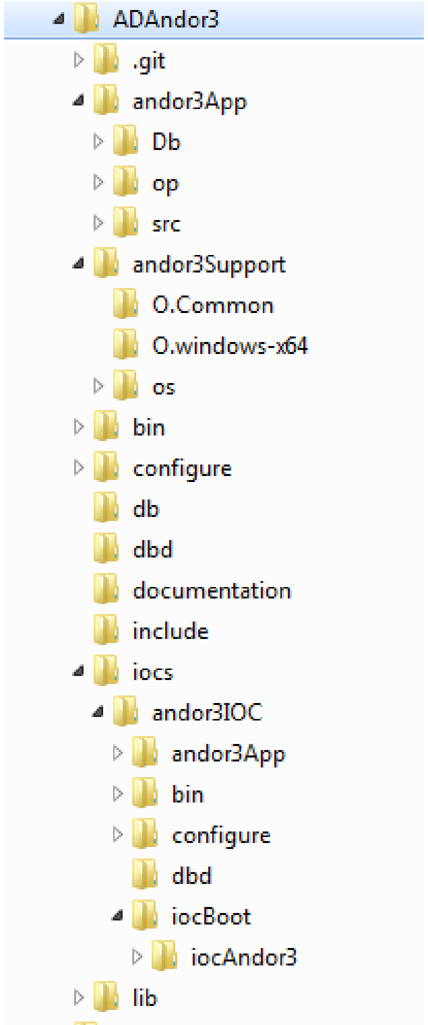
```
git clone -recursive https://github.com/areaDetector.git
```

New Organization

Top-level areaDetector



Name	Date modified	Type
.git	10/17/2014 4:54 PM	File folder
ADADSC	10/17/2014 5:40 PM	File folder
ADAndor	10/17/2014 5:40 PM	File folder
ADAndor3	10/17/2014 5:40 PM	File folder
ADBinaries	10/17/2014 5:38 PM	File folder
ADBruker	10/17/2014 5:40 PM	File folder
ADCore	10/17/2014 5:39 PM	File folder
ADFireWireWin	10/17/2014 5:40 PM	File folder
ADLightField	10/17/2014 5:40 PM	File folder
ADmar345	10/17/2014 5:40 PM	File folder
ADmarCCD	10/17/2014 5:40 PM	File folder
ADPerkinElmer	10/17/2014 5:40 PM	File folder
ADPilatus	10/17/2014 5:40 PM	File folder
ADPixirad	10/17/2014 5:40 PM	File folder
ADPointGrey	10/17/2014 4:54 PM	File folder
ADProsilica	10/17/2014 5:40 PM	File folder
ADPSL	10/17/2014 5:40 PM	File folder
ADPvCam	10/17/2014 5:40 PM	File folder
ADQImaging	10/17/2014 5:01 PM	File folder
ADRoper	10/17/2014 5:00 PM	File folder
ADURL	10/17/2014 5:40 PM	File folder
aravisGigE	10/17/2014 5:02 PM	File folder
configure	10/17/2014 4:51 PM	File folder
documentation	10/17/2014 5:00 PM	File folder
ffmpegServer	10/17/2014 4:59 PM	File folder
ffmpegViewer	10/17/2014 4:55 PM	File folder
firewireDCAM	10/17/2014 4:53 PM	File folder
.gitignore	3/6/2014 2:13 PM	GITIGNORE File
.gitmodules	10/14/2014 6:45 PM	GITMODULES File
INSTALL_GUIDE.md	9/17/2014 5:02 PM	MD File
makeADPrebuilt	4/4/2014 3:18 PM	File
makeADPrebuilt_Current	4/4/2014 3:59 PM	File
Makefile	3/5/2014 1:57 PM	File
makePrebuiltAndor	4/15/2014 4:02 PM	File
README.md	3/26/2014 9:21 PM	MD File
RELEASE.md	10/15/2014 12:18 ...	MD File



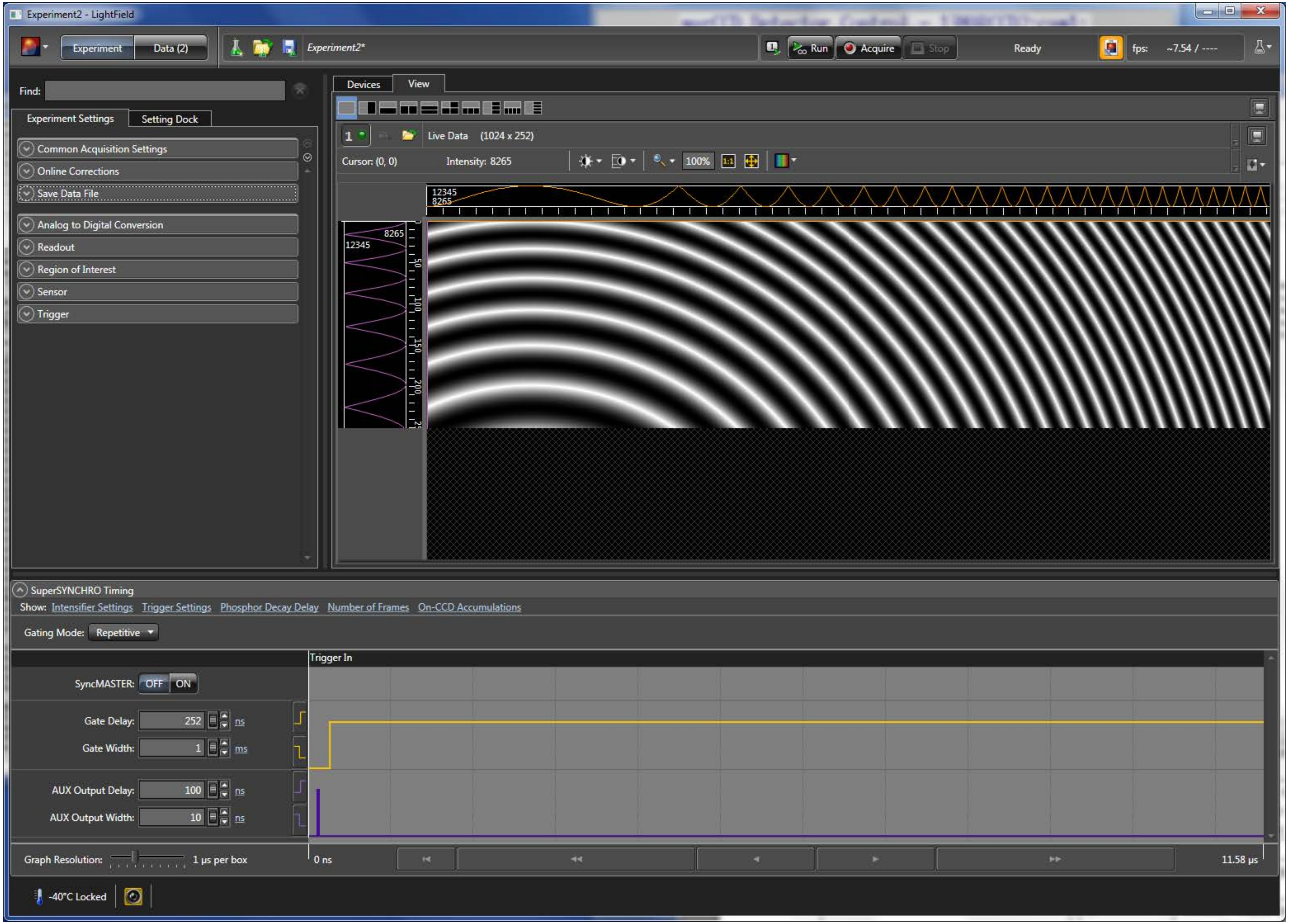
```
ADAndor3
  .git
  andor3App
    Db
    op
    src
  andor3Support
    O.Common
    O.windows-x64
    os
  bin
  configure
  db
  dbd
  documentation
  include
  iocs
    andor3IOC
      andor3App
      bin
      configure
      dbd
      iocBoot
        iocAndor3
  lib
```

- andor3App/ builds only a driver library, and depends only on base and asyn
- iocs/ builds an application and depends on autosave, busy, etc. Can disable building this.

New or improved drivers

- ADAndor3 driver for sCMOS cameras from Andor
- ADLightField driver for Princeton Instruments cameras using their LightField application
- ADAndor added Shamrock spectrometer control for Andor cameras
- ADmarCCD added support for triggered acquisition using new high-speed (-HS) detectors from Rayonix
- ADPSL major rewrite of Photonic Sciences Limited driver to support new server features
- PICAM driver (ADPiCam)
  - Princeton Instruments cameras, including recent models
  - Controlled via PICAM C library
  - Written by John Hammonds

LightField driver



Experiment2 - LightField

Experiment2*

Ready

fps: ~7.54 / ----

Live Data (1024 x 252)

Cursor: (0, 0) Intensity: 8265

SuperSYNCHRO Timing

Gating Mode: Repetitive

SyncMASTER: OFF ON

Gate Delay: 252 ns

Gate Width: 1 ms

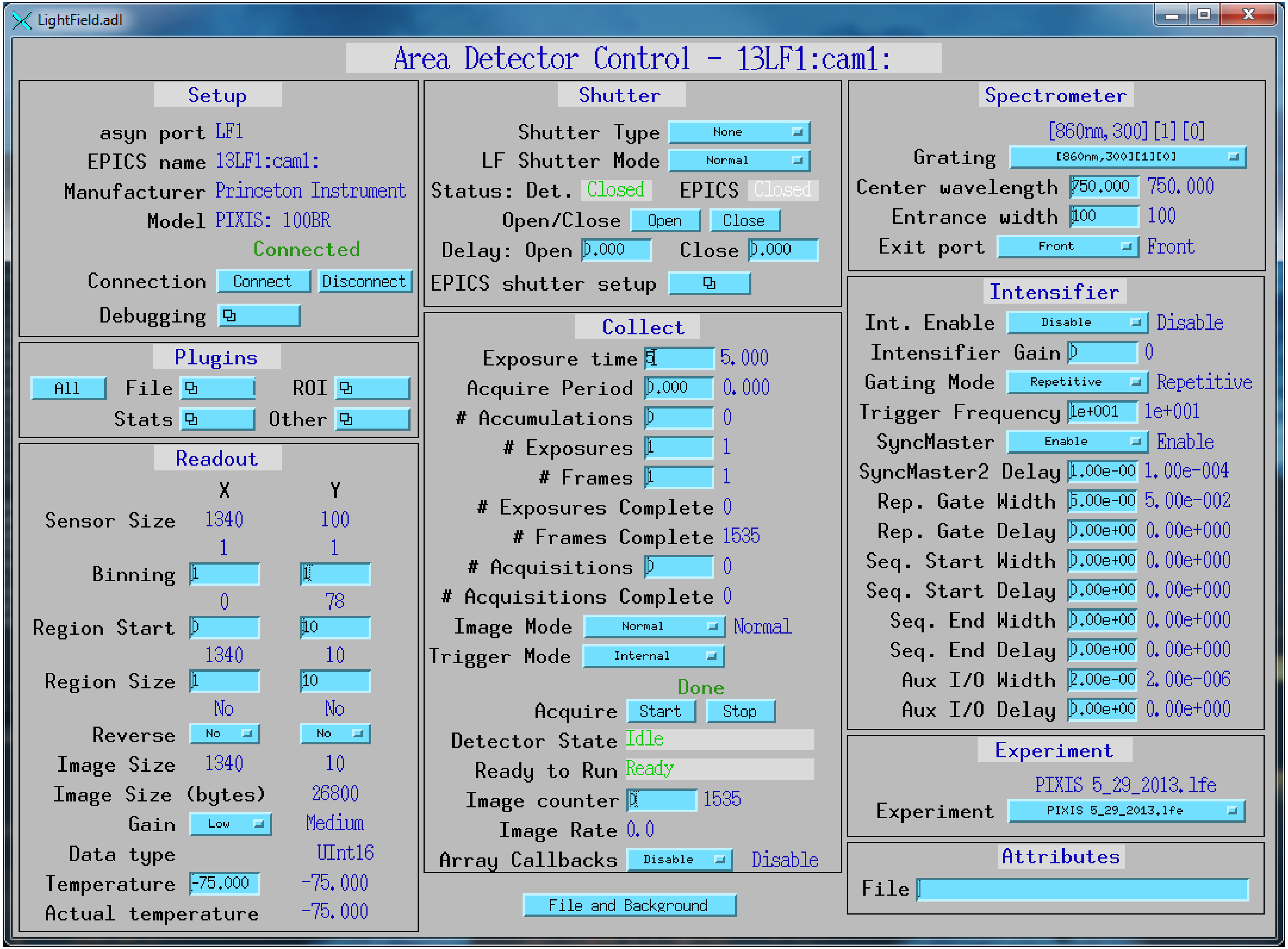
AUX Output Delay: 100 ns

AUX Output Width: 10 ns

Graph Resolution: 1 µs per box

-40°C Locked

LightField driver



LightField.adl

Area Detector Control - 13LF1:cam1:

Setup

asyn port LF1
EPICS name 13LF1:cam1:
Manufacturer Princeton Instrument
Model PIXIS: 100BR
Connected
Connection Connect Disconnect
Debugging

Plugins

All File ROI
Stats Other

Readout

	X		Y	
Sensor Size	1340		100	
Binning	1	1	1	1
Region Start	0	0	10	78
Region Size	1	1340	10	10
Reverse	No	No	No	No
Image Size	1340		10	
Image Size (bytes)			26800	
Gain	Low		Medium	
Data type			UInt16	
Temperature	-75.000		-75.000	
Actual temperature			-75.000	

Shutter

Shutter Type None
LF Shutter Mode Normal
Status: Det. Closed EPICS Closed
Open/Close Open Close
Delay: Open 0.000 Close 0.000
EPICS shutter setup

Collect

Exposure time	5	5.000
Acquire Period	0.000	0.000
# Accumulations	0	0
# Exposures	1	1
# Frames	1	1
# Exposures Complete		0
# Frames Complete		1535
# Acquisitions	0	0
# Acquisitions Complete		0
Image Mode	Normal	Normal
Trigger Mode	Internal	

Done
Acquire Start Stop
Detector State Idle
Ready to Run Ready
Image counter 0 1535
Image Rate 0.0
Array Callbacks Disable Disable

File and Background

Spectrometer

Grating [860nm,300][1][0] — [860nm,300][1][0]
Center wavelength 750.000 750.000
Entrance width 100 100
Exit port Front Front

Intensifier

Int. Enable	Disable	Disable
Intensifier Gain	0	0
Gating Mode	Repetitive	Repetitive
Trigger Frequency	1e+001	1e+001
SyncMaster	Enable	Enable
SyncMaster2 Delay	1.00e-00	1.00e-004
Rep. Gate Width	5.00e-00	5.00e-002
Rep. Gate Delay	0.00e+00	0.00e+000
Seq. Start Width	0.00e+00	0.00e+000
Seq. Start Delay	0.00e+00	0.00e+000
Seq. End Width	0.00e+00	0.00e+000
Seq. End Delay	0.00e+00	0.00e+000
Aux I/O Width	2.00e-00	2.00e-006
Aux I/O Delay	0.00e+00	0.00e+000

Experiment

Experiment PIXIS 5_29_2013.lfe — PIXIS 5_29_2013.lfe

Attributes

File

Detector drivers (continued)

- Generic GigE driver (aravisGigE)
  - Should work with any GigEVision compliant camera. From Tom Cobb at Diamond.
  - Controlled using the Aravis reverse-engineered GigEVision library
- QImaging driver (ADQImaging)
  - QImaging cameras.
  - Controlled using Qimaging SDK
  - Written by Arthur Glowacki
- ADPvAccess
  - Driver that receives NDArrays over EPICS V4
  - Allows plugins to run in an EPICS IOC on a different machine than the detector
  - Written by David Hickin from Diamond.
  - Another version by Bruno Martins from BNL will likely be part of ADCore R3-0.

Detector drivers (continued)

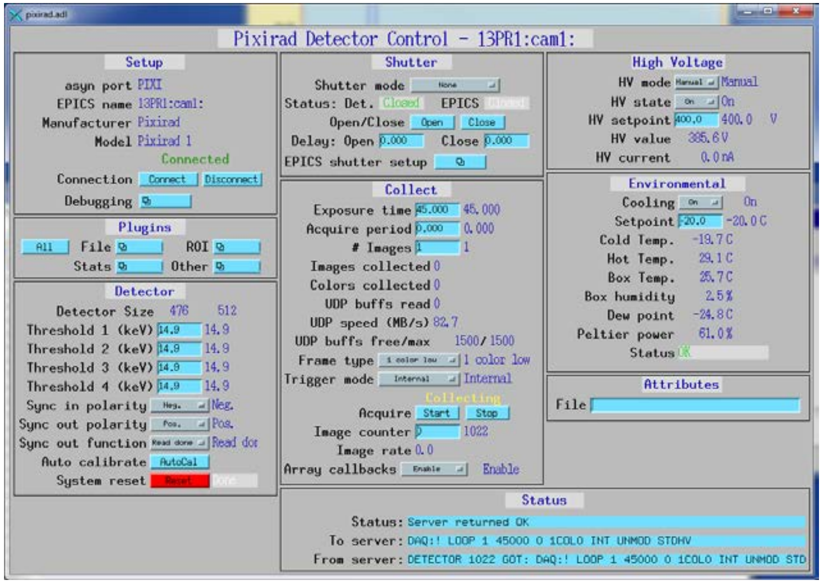
- Perkin Elmer Dexela driver (ADDexela)
  - For Perkin Elmer Dexela CMOS flat-panel detectors
  - Written by Mark Rivers
- Quantum Detectors Merlin driver (ADMerlin)
  - For Merlin Medipix3 detector
  - Written by Giles Knapp at Diamond
- ADFastCCD
  - Driver for LBNL FastCCD.
  - Written by Stuart Wilkins at NSLS-II

Point Grey driver

- New driver for all cameras from Point Grey using their FlyCap2 SDK.
- Firewire, GigE and USB 3.0
- High performance, low cost
- Example: Model GS3-U3-23S6M
  - 1920 x 1200 global shutter CMOS
  - No smear • Distortion-free
  - Dynamic range of 73 dB
  - Peak QE of 76%
  - Read noise of 7e-
  - Max frame rate of 162 fps (~400 MB/S, 4X faster than GigE)
  - USB 3.0 interface
  - $1,295
  - Comparable to PCO Edge and Andor Zyla for 10X less money

R2-0: Pixirad driver

- New driver for Pixirad CdTe pixel array detector
- Similar to Pilatus, but CdTe gives very high efficiency to 80 keV or more
- 2 energy thresholds, so 2 different energy images simultaneously
- In detector pool



pixirad.adl

Pixirad Detector Control - 13PR1:cam1:

Setup

asyn port PIXI
EPICS name 13PR1:cam1:
Manufacturer Pixirad
Model Pixirad 1
Connected
Connection Connect Disconnect
Debugging

Plugins

All File ROI
Stats Other

Detector

Detector Size 476 512
Threshold 1 (keV) 14.9 14.9
Threshold 2 (keV) 14.9 14.9
Threshold 3 (keV) 14.9 14.9
Threshold 4 (keV) 14.9 14.9
Sync in polarity Neg. Neg.
Sync out polarity Pos. Pos.
Sync out function Read done Read dor
Auto calibrate AutoCal
System reset Reset Done

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 45.000 45.000
Acquire period 0.000 0.000
Images 1 1
Images collected 0
Colors collected 0
UDP buffs read 0
UDP speed (MB/s) 82.7
UDP buffs free/max 1500/ 1500
Frame type 1 color low 1 color low
Trigger mode Internal Internal
Collecting
Acquire Start Stop
Image counter 0 1022
Image rate 0.0
Array callbacks Enable Enable

High Voltage

HV mode Manual Manual
HV state On On
HV setpoint 400.0 400.0 V
HV value 395.6 V
HV current 0.0 nA

Environmental

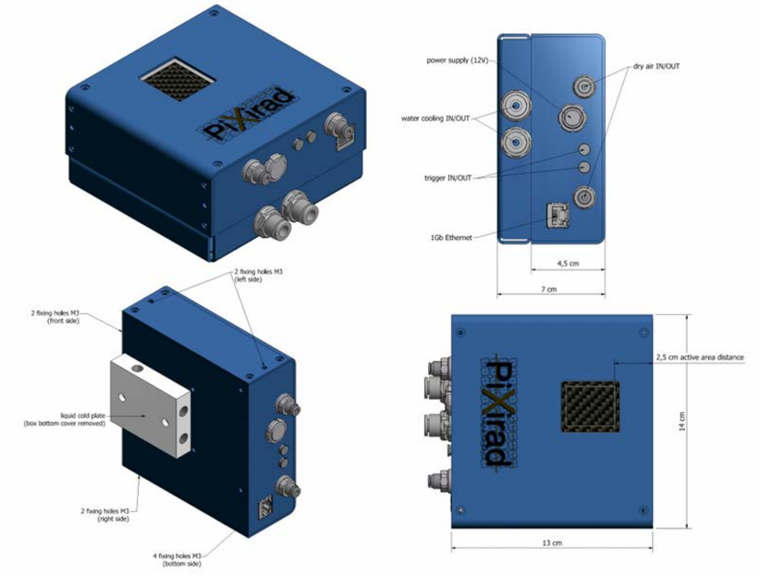
Cooling On On
Setpoint -20.0 -20.0 C
Cold Temp. -19.7 C
Hot Temp. 29.1 C
Box Temp. 25.7 C
Box humidity 2.5 %
Dew point -24.8 C
Peltier power 61.0 %
Status OK

Attributes

File

Status

Status: Server returned OK
To server: DAQ:! LOOP 1 45000 0 1COLO INT UNMOD STDHV
From server: DETECTOR 1022 GOT: DAQ:! LOOP 1 45000 0 1COLO INT UNMOD STD


PiXirad
power supply (12V)
dry air IN/OUT
water cooling IN/OUT
trigger IN/OUT
1Gb Ethernet
4,5 cm
7 cm
2 fixing holes M3 (left side)
2 fixing holes M3 (front side)
liquid cold plate (box bottom cover removed)
2 fixing holes M3 (right side)
4 fixing holes M3 (bottom side)
2,5 cm active area distance
14 cm
13 cm
PiXirad

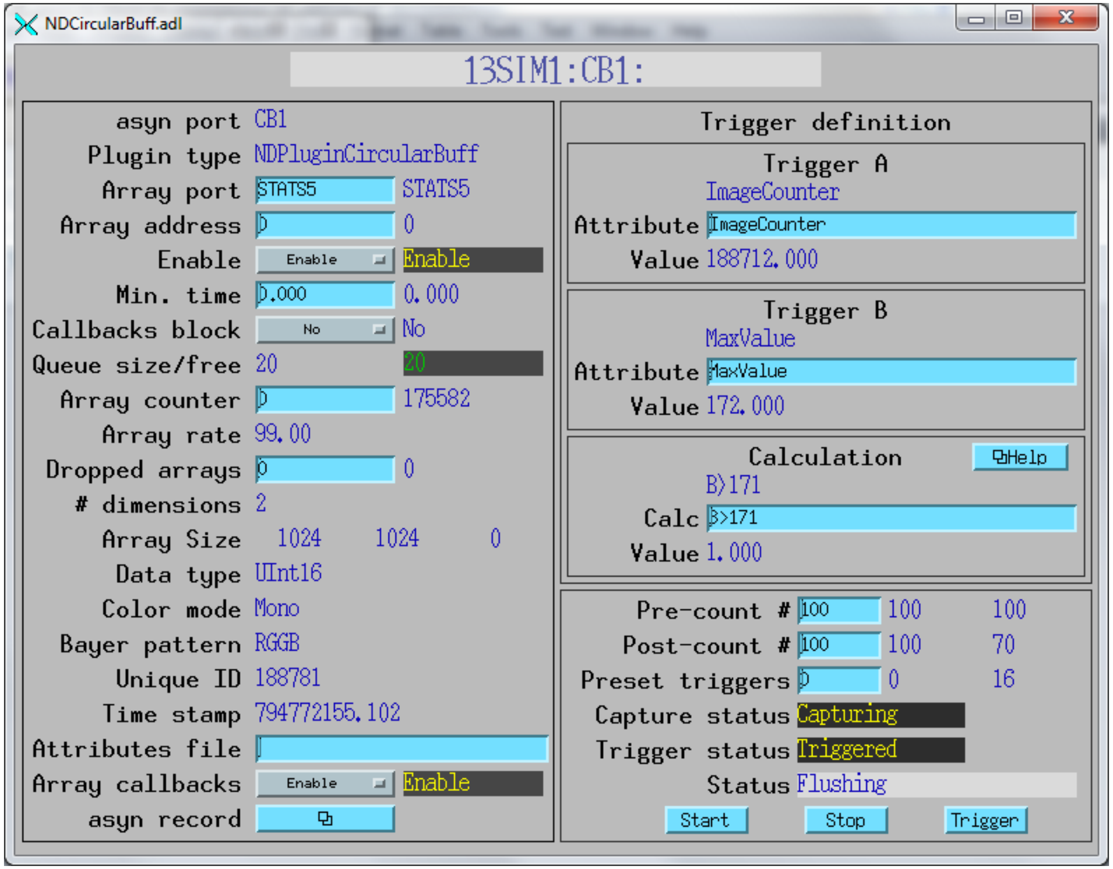
NDAttributes

- Detector drivers and plugins read files like this to add NDAttributes to the arrays
- Values can come from driver/plugin, EPICS PVs, or user-written functions
- Can be written to files, used to control downstream plugins, etc.

```xml
<?xml version="1.0" standalone="no" ?>
<!-- Attributes -->
<Attributes>
    <Attribute name="Acquiretime"     type="PARAM"    source="ACQ_TIME"                    datatype="DOUBLE"     description="Camera acquire time"/>
    <Attribute name="CameraModel"     type="PARAM"    source="MODEL"                       datatype="STRING"     description="CameraModel"/>
    <Attribute name="Comment1"        type="EPICS_PV" source="13BMDPG1:TIFF1:FileTemplate" dbrtype="DBR_STRING"  description="Comment 1"/>
    <Attribute name="Comment2"        type="EPICS_PV" source="13BMDPG1:TIFF1:FilePath"     dbrtype="DBR_STRING"  description="Comment 2"/>
    <Attribute name="Comment3"        type="EPICS_PV" source="13BMDPG1:TIFF1:FileName"     dbrtype="DBR_STRING"  description="Comment 3"/>
    <Attribute name="SampleX"         type="EPICS_PV" source="13BMD:m85.RBV"               dbrtype="DBR_NATIVE"  description="Bottom X stage translation"/>
    <Attribute name="SampleOmega"     type="EPICS_PV" source="13BMD:m38.RBV"               dbrtype="DBR_NATIVE"  description="Sample rotation"/>
    <Attribute name="RotationSpeed"   type="EPICS_PV" source="13BMD:m38.VELO"              dbrtype="DBR_NATIVE"  description="Sample rotation speed"/>
    <Attribute name="SampleY"         type="EPICS_PV" source="13BMD:m90.VAL"               dbrtype="DBR_NATIVE"  description="Sample vertical height"/>
    <Attribute name="SampleXCent"     type="EPICS_PV" source="13BMD:m91.VAL"               dbrtype="DBR_NATIVE"  description="Sample X centering"/>
    <Attribute name="SampleYCent"     type="EPICS_PV" source="13BMD:m89.VAL"               dbrtype="DBR_NATIVE"  description="Sample Y centering"/>
    <Attribute name="CameraX"         type="EPICS_PV" source="13BMD:m33.VAL"               dbrtype="DBR_NATIVE"  description="Camera X position"/>
    <Attribute name="CameraY"         type="EPICS_PV" source="13BMD:m34.VAL"               dbrtype="DBR_NATIVE"  description="Camera Y (focus) position"/>
    <Attribute name="CameraZ"         type="EPICS_PV" source="13BMD:m35.VAL"               dbrtype="DBR_NATIVE"  description="Camera Z position"/>
    <Attribute name="CameraDistance"  type="EPICS_PV" source="13BMD:m70.VAL"               dbrtype="DBR_NATIVE"  description="Camera distance"/>
    <Attribute name="CameraRotation"  type="EPICS_PV" source="13BMD:m37.VAL"               dbrtype="DBR_NATIVE"  description="Camera rotation"/>
    <Attribute name="ExposureTime"    type="EPICS_PV" source="13BMD:SIS1:LNEOutputWidth"   dbrtype="DBR_NATIVE"  description="SIS output width=exposure time"/>
    <Attribute name="ExposurePeriod"  type="EPICS_PV" source="13BMD:SIS1:Dwell"            dbrtype="DBR_NATIVE"  description="SIS Dwell=exposure period"/>
    <Attribute name="LiftHeight"      type="EPICS_PV" source="13BMD:m22.VAL"               dbrtype="DBR_NATIVE"  description="Lift table height"/>
    <Attribute name="LiftX"           type="EPICS_PV" source="13BMD:XAS:t1.EX"             dbrtype="DBR_NATIVE"  description="Lift table X"/>
    <Attribute name="LiftAX"          type="EPICS_PV" source="13BMD:XAS:t1.EAX"            dbrtype="DBR_NATIVE"  description="Lift table AX"/>
    <Attribute name="LiftY"           type="EPICS_PV" source="13BMD:XAS:t1.EY"             dbrtype="DBR_NATIVE"  description="Lift table Y"/>
    <Attribute name="LiftAY"          type="EPICS_PV" source="13BMD:XAS:t1.EAY"            dbrtype="DBR_NATIVE"  description="Lift table AY"/>
    <Attribute name="LiftZ"           type="EPICS_PV" source="13BMD:XAS:t1.EZ"             dbrtype="DBR_NATIVE"  description="Lift table Z"/>
    <Attribute name="LiftAZ"          type="EPICS_PV" source="13BMD:XAS:t1.EAZ"            dbrtype="DBR_NATIVE"  description="Lift table AZ"/>
    <Attribute name="Energy"          type="EPICS_PV" source="13BMA:E:E_RBV"               dbrtype="DBR_NATIVE"  description="Monochromator energy"/>
    <Attribute name="BeamOffset"      type="EPICS_PV" source="13BMA:E:height"              dbrtype="DBR_NATIVE"  description="Monochromator offset"/>
    <Attribute name="MonoFBSetpoint"  type="EPICS_PV" source="13BMA:mono_pid1.VAL"         dbrtype="DBR_NATIVE"  description="Mono feedback setpoint"/>
    <Attribute name="MonoFBReadback"  type="EPICS_PV" source="13BMA:mono_pid1.CVAL"        dbrtype="DBR_NATIVE"  description="Mono feedback readback"/>
    <Attribute name="MonoFBOnOff"     type="EPICS_PV" source="13BMA:mono_pid1.FBON"        dbrtype="DBR_STRING"  description="Mono feedback on/off"/>
    <Attribute name="BMDSlitHSize"    type="EPICS_PV" source="13BMD:BMDHsize.VAL"          dbrtype="DBR_NATIVE"  description="BMD horizontal slit size"/>
    <Attribute name="BMDSlitVSize"    type="EPICS_PV" source="13BMD:BMDVsize.VAL"          dbrtype="DBR_NATIVE"  description="BMD vertical slit size"/>
    <Attribute name="BMDSlitHCenter"  type="EPICS_PV" source="13BMD:BMDHcenter.VAL"        dbrtype="DBR_NATIVE"  description="BMD horizontal slit center"/>
    <Attribute name="BMDSlitVCenter"  type="EPICS_PV" source="13BMD:BMDVcenter.VAL"        dbrtype="DBR_NATIVE"  description="BMD vertical slit center"/>
    <Attribute name="KeithleyGain"    type="EPICS_PV" source="13BMD:A3sens_num.VAL"        dbrtype="DBR_STRING"  description="Keithley gain"/>
    <Attribute name="KeithleyUnits"   type="EPICS_PV" source="13BMD:A3sens_unit.VAL"       dbrtype="DBR_STRING"  description="Keithley units"/>
    <Attribute name="RingCurrent"     type="EPICS_PV" source="S:SRcurrentAI.VAL"           dbrtype="DBR_NATIVE"  description="Ring current"/>
</Attributes>
```

NDPluginCircularBuff

- Buffers NDArrays in a circular buffer.
- 2 NDAttributes can be used in trigger equation.
- Outputs the arrays when trigger calculation is true, or when forced.
- Supports pre-trigger and post-trigger samples
- Written by Alan Greer at Observatory Sciences



13SIM1:CB1:

asyn port	CB1	
Plugin type	NDPluginCircularBuff	
Array port	STATS5	STATS5
Array address	0	0
Enable	Enable	Enable
Min. time	0.000	0.000
Callbacks block	No	No
Queue size/free	20	20
Array counter	0	175582
Array rate	99.00	
Dropped arrays	0	0
# dimensions	2	
Array Size	1024 1024 0	
Data type	UInt16	
Color mode	Mono	
Bayer pattern	RGGB	
Unique ID	188781	
Time stamp	794772155.102	
Attributes file		
Array callbacks	Enable	Enable
asyn record		

Trigger definition

Trigger A

ImageCounter

Attribute	ImageCounter
Value	188712.000

Trigger B

MaxValue

Attribute	MaxValue
Value	172.000

Calculation

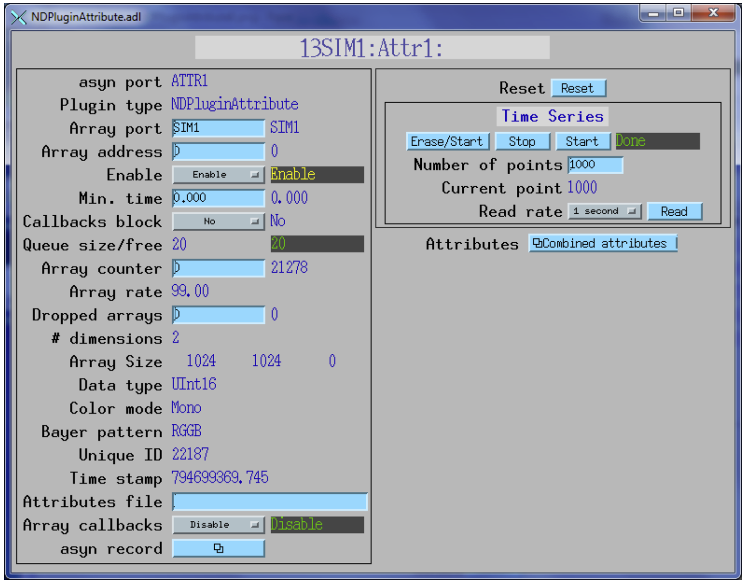
B>171

Calc	B>171
Value	1.000

Pre-count #	100	100	100
Post-count #	100	100	70
Preset triggers	0	0	16
Capture status	Capturing		
Trigger status	Triggered		
Status	Flushing		

NDPluginAttribute

- Extracts NDAttributes from an NDArray and publishes as scalar and time-series arrays
- Written by Matt Pearson at ORNL



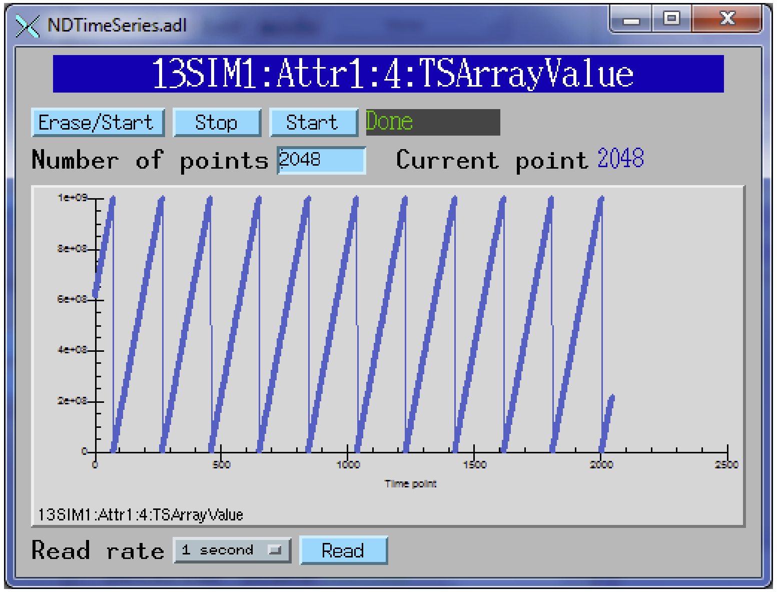
13SIM1:Attr1:

asyn port ATTR1
Plugin type NDPluginAttribute
Array port SIM1 SIM1
Array address 0 0
Enable Enable Enable
Min. time 0.000 0.000
Callbacks block No No
Queue size/free 20 20
Array counter 0 21278
Array rate 99.00
Dropped arrays 0 0
# dimensions 2
Array Size 1024 1024 0
Data type UInt16
Color mode Mono
Bayer pattern RGGB
Unique ID 22187
Time stamp 794699369.745
Attributes file
Array callbacks Disable Disable
asyn record

Reset Reset

Time Series
Erase/Start Stop Start Done
Number of points 1000
Current point 1000
Read rate 1 second Read

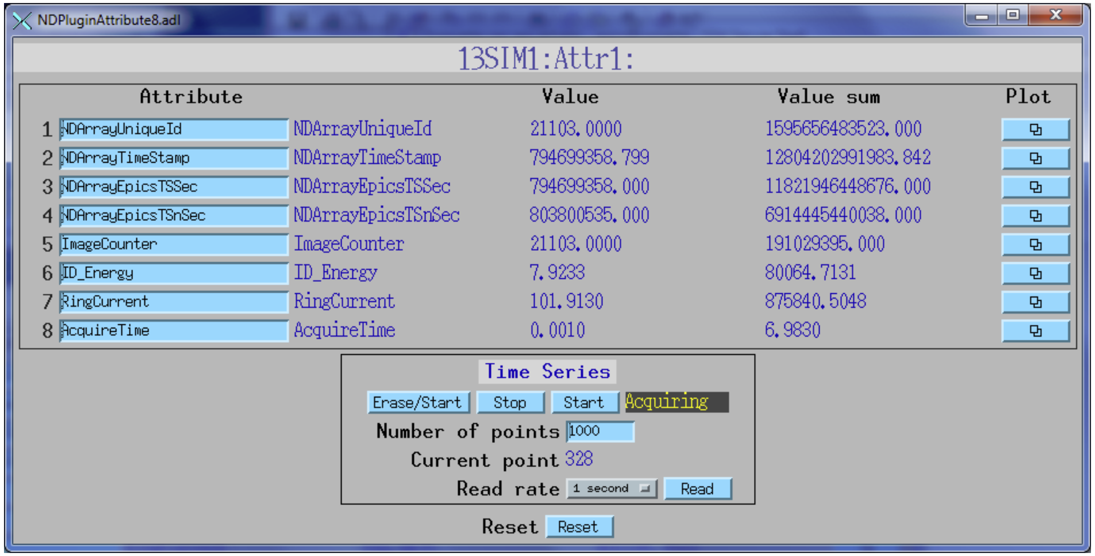
Attributes Combined attributes



13SIM1:Attr1:4:TSArrayValue

Erase/Start Stop Start Done
Number of points 2048 Current point 2048

Read rate 1 second Read



13SIM1:Attr1:

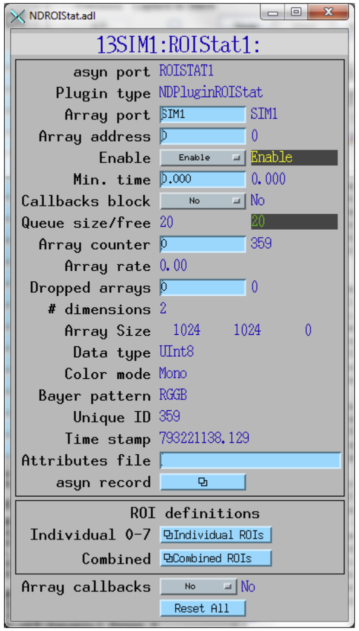
	Attribute		Value	Value sum	Plot
1	NDArrayUniqueId	NDArrayUniqueId	21103.0000	1595656483523.000	
2	NDArrayTimeStamp	NDArrayTimeStamp	794699358.799	12804202991983.842	
3	NDArrayEpicsTSSec	NDArrayEpicsTSSec	794699358.000	11821946448676.000	
4	NDArrayEpicsTSnSec	NDArrayEpicsTSnSec	803800535.000	6914445440038.000	
5	ImageCounter	ImageCounter	21103.0000	191029395.000	
6	ID_Energy	ID_Energy	7.9233	80064.7131	
7	RingCurrent	RingCurrent	101.9130	875840.5048	
8	AcquireTime	AcquireTime	0.0010	6.9830	

Time Series
Erase/Start Stop Start Acquiring
Number of points 1000
Current point 328
Read rate 1 second Read

Reset Reset

NDPluginROIStat

- Supports multiple regions-of-interest with simple statistics on each.
- More efficient and convenient than using NDPluginROI and NDPluginStats when many regions of interest with simple statistics are needed.
- Written by Matthew Pearson at ORNL

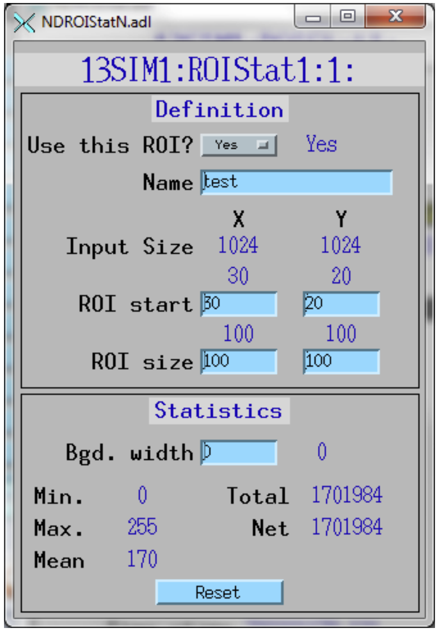


13SIM1:ROIStat1:

asyn port	ROISTAT1	
Plugin type	NDPluginROIStat	
Array port	SIM1	SIM1
Array address	0	0
Enable	Enable	Enable
Min. time	0.000	0.000
Callbacks block	No	No
Queue size/free	20	20
Array counter	0	359
Array rate	0.00	
Dropped arrays	0	0
# dimensions	2	
Array Size	1024 1024 0	
Data type	UInt8	
Color mode	Mono	
Bayer pattern	RGGB	
Unique ID	359	
Time stamp	793221138.129	
Attributes file		
asyn record	⧉	

ROI definitions

Individual 0-7	⧉Individual ROIs
Combined	⧉Combined ROIs
Array callbacks	No No
	Reset All



13SIM1:ROIStat1:1:

Definition

Use this ROI? Yes Yes

Name test

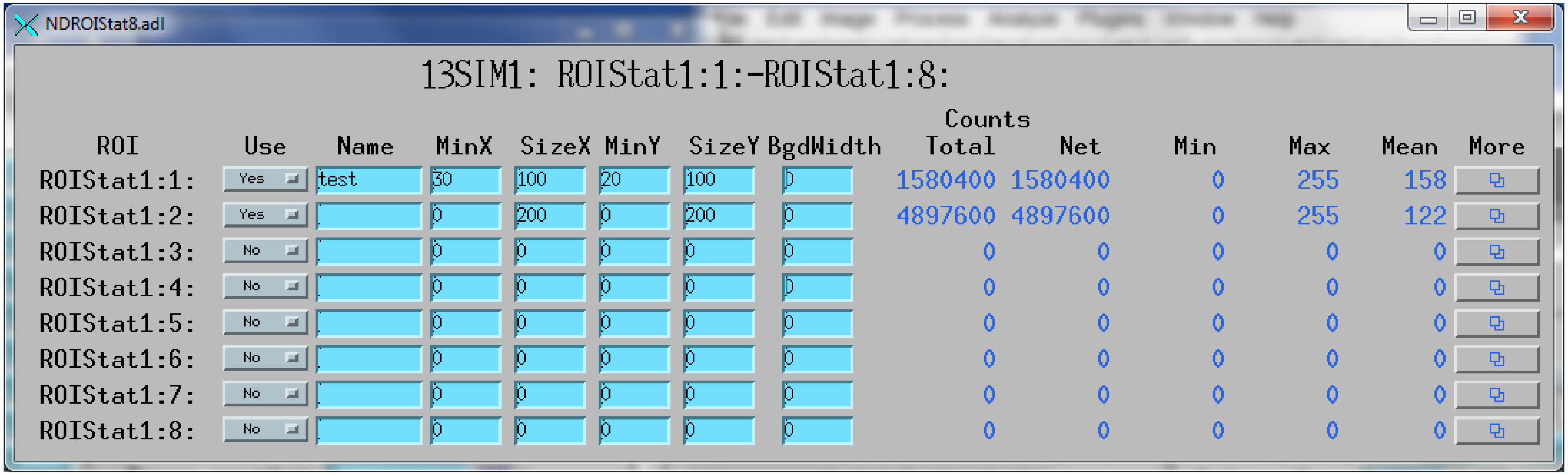
	X	Y
Input Size	1024	1024
	30	20
ROI start	30	20
	100	100
ROI size	100	100

Statistics

Bgd. width 0 0

Min.	0	Total	1701984
Max.	255	Net	1701984
Mean	170		

Reset

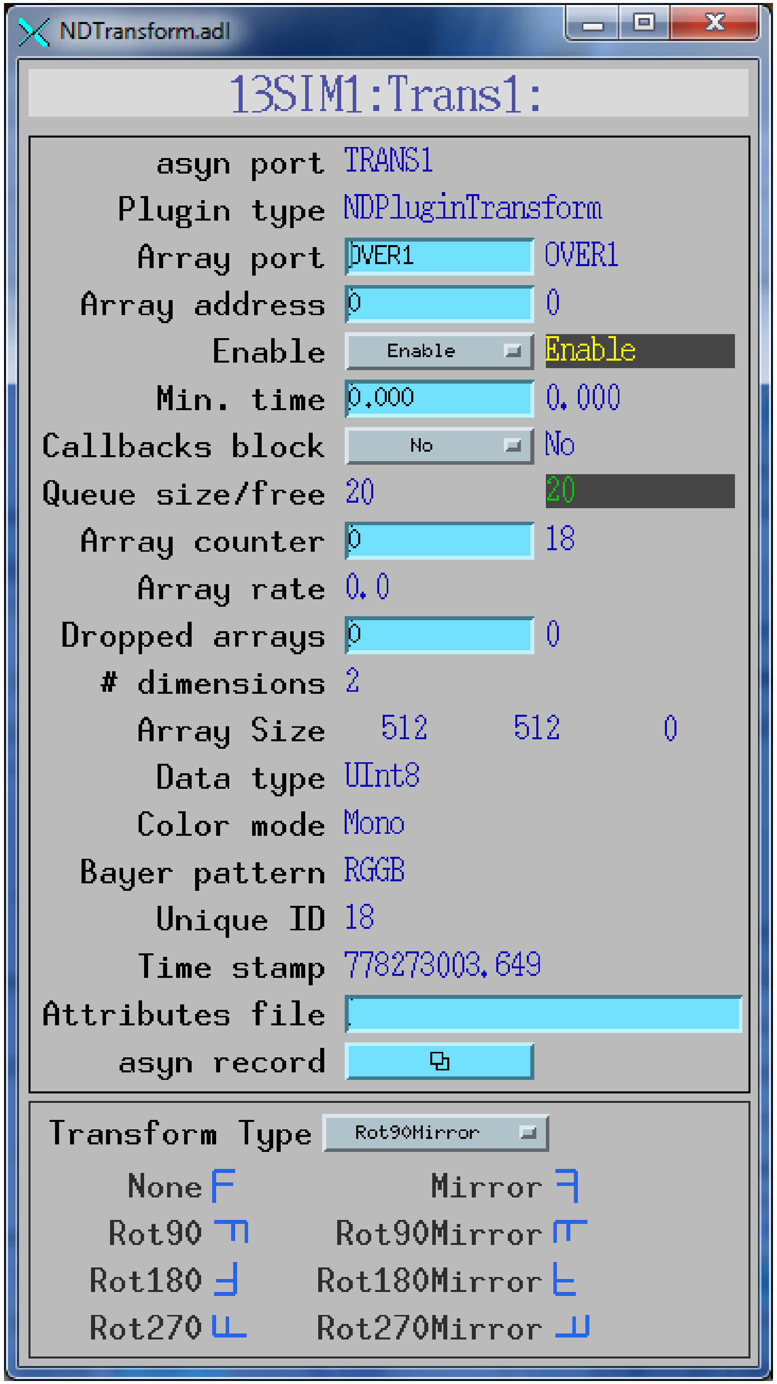


13SIM1: ROIStat1:1:-ROIStat1:8:

ROI	Use	Name	MinX	SizeX	MinY	SizeY	BgdWidth	Counts Total	Net	Min	Max	Mean	More
ROIStat1:1:	Yes	test	30	100	20	100	0	1580400	1580400	0	255	158	⧉
ROIStat1:2:	Yes		0	200	0	200	0	4897600	4897600	0	255	122	⧉
ROIStat1:3:	No		0	0	0	0	0	0	0	0	0	0	⧉
ROIStat1:4:	No		0	0	0	0	0	0	0	0	0	0	⧉
ROIStat1:5:	No		0	0	0	0	0	0	0	0	0	0	⧉
ROIStat1:6:	No		0	0	0	0	0	0	0	0	0	0	⧉
ROIStat1:7:	No		0	0	0	0	0	0	0	0	0	0	⧉
ROIStat1:8:	No		0	0	0	0	0	0	0	0	0	0	⧉

NDPluginTransform improvements

- Greatly simplified: just 8 operations including null operation
- 13-85 times faster than previous releases depending on data type, color mode



13SIM1:Trans1:

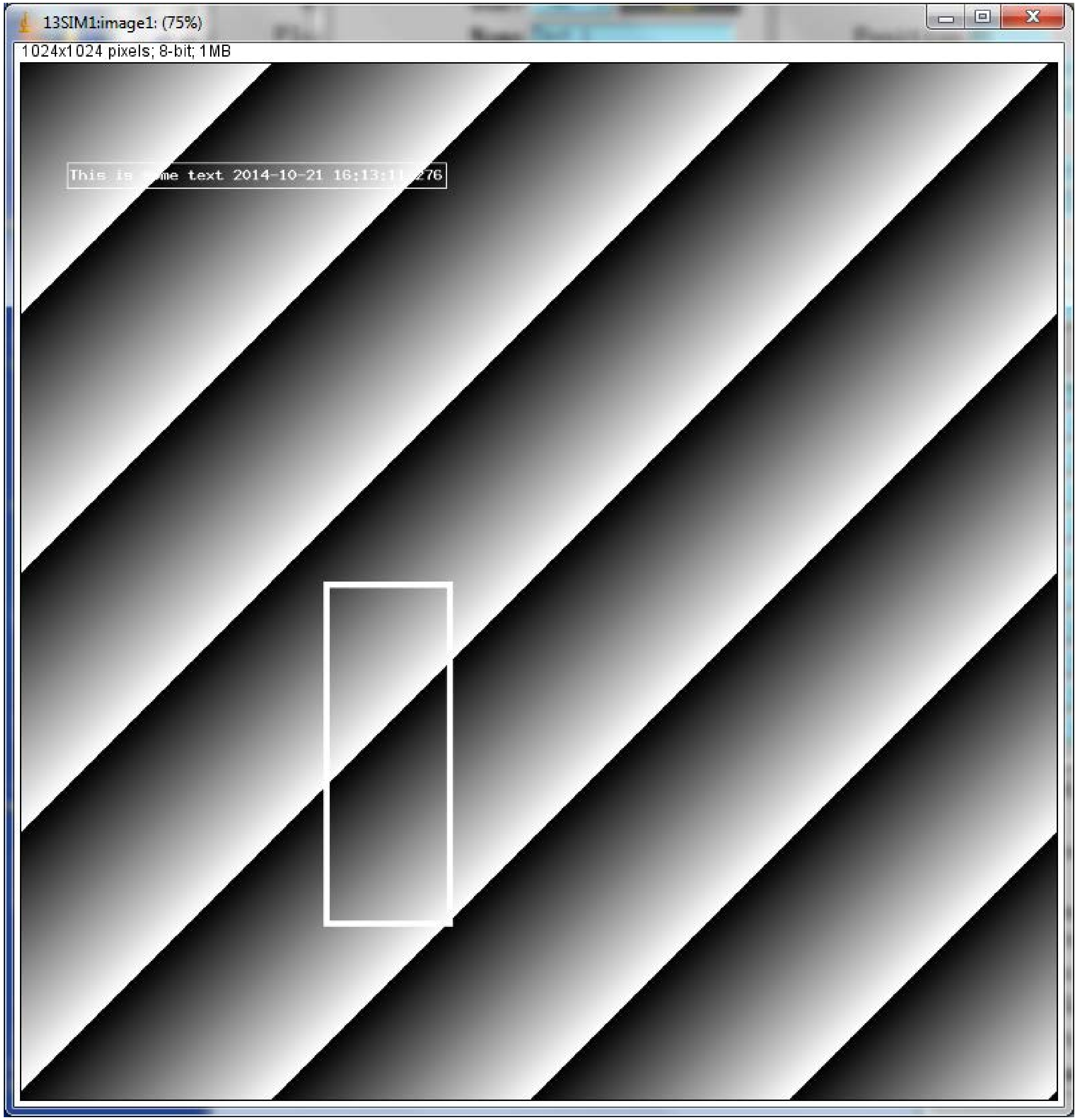
asyn port	TRANS1		
Plugin type	NDPluginTransform		
Array port	OVER1	OVER1	
Array address	0	0	
Enable	Enable	Enable	
Min. time	0.000	0.000	
Callbacks block	No	No	
Queue size/free	20	20	
Array counter	0	18	
Array rate	0.0		
Dropped arrays	0	0	
# dimensions	2		
Array Size	512	512	0
Data type	UInt8		
Color mode	Mono		
Bayer pattern	RGGB		
Unique ID	18		
Time stamp	778273003.649		
Attributes file			
asyn record			

Transform Type: Rot90Mirror

None	Mirror
Rot90	Rot90Mirror
Rot180	Rot180Mirror
Rot270	Rot270Mirror

NDPluginOverlay

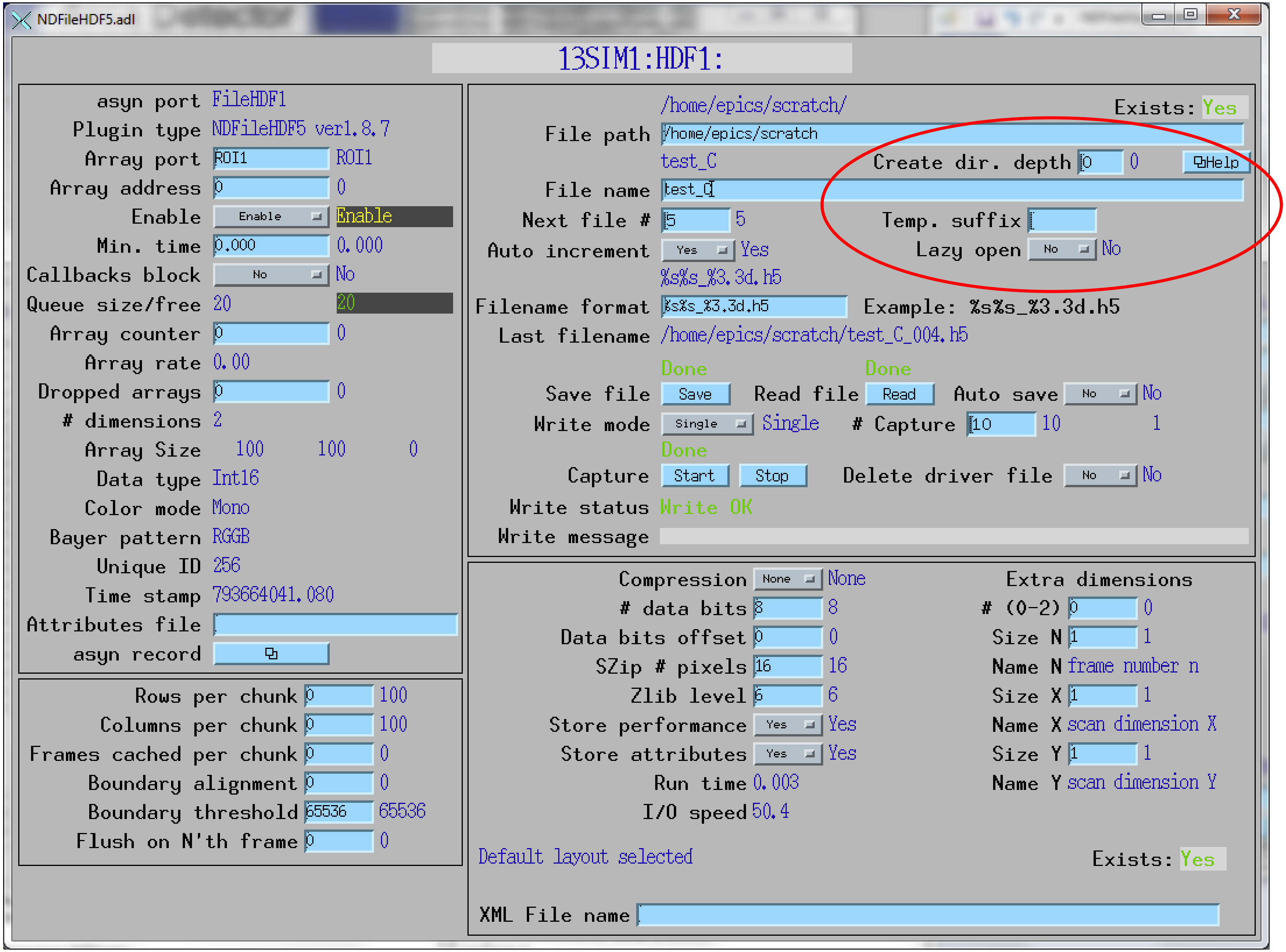
- Added support for text overlays (Keith Brister, LS-CAT)
- Added support for line widths in cross and rectangle overlays (Matt Pearson, ORNL)


13SIM1:image1: (75%)
1024x1024 pixels; 8-bit; 1MB
This is some text 2014-10-21 16:13:11.276

Other Recent Plugins

- ffmpegServer
  - MJPEG server that allows viewing images in a Web browser. From DLS.
  - Puts compressed images on the network, greatly reducing bandwidth compared to uncompressed channel access arrays.
  - Written by Tom Cobb from Diamond
- ADPvAccess
  - Plugin that sends NDArrays over EPICS V4
  - Allows plugins to run in an EPICS IOC on a different machine than the detector
  - Written by David Hickin from Diamond.
- ADPluginEdge
  - Does edge detection using the OpenCV Canny function
  - Written by Keith Brister at LS-CAT

NDPluginFile



NDFileHDF5.adl

13SIM1:HDF1:

asyn port FileHDF1
Plugin type NDFileHDF5 ver1.8.7
Array port ROI1 ROI1
Array address 0 0
Enable Enable Enable
Min. time 0.000 0.000
Callbacks block No No
Queue size/free 20 20
Array counter 0 0
Array rate 0.00
Dropped arrays 0 0
# dimensions 2
Array Size 100 100 0
Data type Int16
Color mode Mono
Bayer pattern RGGB
Unique ID 256
Time stamp 793664041.080
Attributes file
asyn record

Rows per chunk 0 100
Columns per chunk 0 100
Frames cached per chunk 0 0
Boundary alignment 0 0
Boundary threshold 65536 65536
Flush on N'th frame 0 0

/home/epics/scratch/ Exists: Yes
File path /home/epics/scratch
test_C Create dir. depth 0 0 Help
File name test_C
Next file # 5 5 Temp. suffix
Auto increment Yes Yes Lazy open No No
%s%s_%3.3d.h5
Filename format %s%s_%3.3d.h5 Example: %s%s_%3.3d.h5
Last filename /home/epics/scratch/test_C_004.h5
Done Done
Save file Save Read file Read Auto save No No
Write mode Single Single # Capture 10 10 1
Done
Capture Start Stop Delete driver file No No
Write status Write OK
Write message

Compression None None
# data bits 8 8
Data bits offset 0 0
SZip # pixels 16 16
Zlib level 6 6
Store performance Yes Yes
Store attributes Yes Yes
Run time 0.003
I/O speed 50.4

Extra dimensions
# (0-2) 0 0
Size N 1 1
Name N frame number n
Size X 1 1
Name X scan dimension X
Size Y 1 1
Name Y scan dimension Y

Default layout selected Exists: Yes

XML File name

New

NDPluginFile Recent Features

- File plugins can now create directories
  - CreateDirectory record controls whether directories are created if they don't exist.
    - Zero (default), no directories are created.
    - Negative, then absolute value is the maximum of directories that will be created (i.e. -1 will create a maximum of one directory to complete the path, -2 will create a maximum of 2 directories).
    - Positive, then at least that many directories in the path must exist (i.e. a value of 1 will create all directories below the root directory and 2 will not create a directory in the root directory).

NDPluginFile Recent Features

- “Lazy-open”
  - Normally files in stream mode are opened when Capture PV is set to 1
  - This requires that there have already been an NDArray received by that plugin with the correct dimensions and attributes
  - “Lazy-open” is selected the file is not opened until the first NDArray callback happens after Capture is set to 1.
  - Simpler for users, but poorer performance, can lead to dropped arrays
- File plugins can write files with a temporary suffix and then rename the file after writing is complete.
  - Allows rsync, etc. to be used to copy files, with guarantee that they are complete

File Plugin Enhancements

- NDFileTIFF
  - Supports any NDArray data type
  - Stores NDAttributes as ASCII user tags, up to 490.
- NDFileHDF5
  - Now supports using an XML file to define the layout and placement of NDArrays and NDAttributes in the HDF5 file.
  - Can be used to create NeXus-compliant files without using NDFileNeXus plugin
  - NDFileNeXus will probably be deprecated in a future release.
  - Major project, collaboration of APS and Diamond (Arthur Glowacki, Ulrik Pedersen, Alan Greer).

NDFileHDF5

XML file to define file layout

```
<xml>
  <group name="entry">
    <attribute name="NX_class" source="constant" value="NXentry" type="string"></attribute>
    <group name="instrument">
      <attribute name="NX_class" source="constant" value="NXinstrument" type="string"></attribute>
      <group name="detector">
        <attribute name="NX_class" source="constant" value="NXdetector" type="string"></attribute>
        <dataset name="data" source="detector" det_default="true">
          <attribute name="NX_class" source="constant" value="SDS" type="string"></attribute>
          <attribute name="signal" source="constant" value="1" type="int"></attribute>
          <attribute name="target" source="constant" value="/entry/instrument/detector/data"
                     type="string"></attribute>
        </dataset>
        <group name="NDAttributes">
          <attribute name="NX_class" source="constant" value="NXcollection" type="string"></attribute>
          <dataset name="ColorMode" source="ndattribute" ndattribute="ColorMode">
          </dataset>
        </group>          <!-- end group NDAttribute -->
      </group>            <!-- end group detector -->
      <group name="NDAttributes" ndattr_default="true">
        <attribute name="NX_class" source="constant" value="NXcollection" type="string"></attribute>
      </group>            <!-- end group NDAttribute (default) -->
      <group name="performance">
        <dataset name="timestamp" source="ndattribute"></dataset>
      </group>            <!-- end group performance -->
    </group>              <!-- end group instrument -->
    <group name="data">
      <attribute name="NX_class" source="constant" value="NXdata" type="string"></attribute>
      <hardlink name="data" target="/entry/instrument/detector/data"></hardlink>
      <!-- The "target" attribute in /entry/instrument/detector/data is used to
           tell Nexus utilities that this is a hardlink -->
    </group>              <!-- end group data -->
  </group>                <!-- end group entry -->
</xml>
```

Other Recent Enhancements

- Added new attribute type, NDAttrSourceFunct. This type of attribute gets its value from a user-defined C++ function. It can thus be use to get any type of metadata. Previously only EPICS PVs and driver/plugin parameters were available as metadata.
- (P)(R)ADCoreVersion_RBV provides version of ADCore. This allows CA clients to alter their behavior depending on version of ADCore used to build the plugin or driver.
- simDetectorNoIOC
  - Example standalone C++ application that instantiates a simDetector without running an EPICS IOC
  - Shows that areaDetector drivers and plugins only depend on libCom and asyn libraries. Can be used from other control systems.

Future Ideas (R3-0?)

- Simplify NDPluginFile base class and way file saving works
  - Remove the Single/Stream/Capture mode.
- Two parameters
  - # NDArrays to save (already present)
  - # NDArrays per file (new)
  - This allows saving only 1 array per HDF5 file, which is not possible now in Stream mode.
- Capture mode can be replaced:
  - Make input queue large enough OR
  - Use new NDPluginCircularBuffer

Future Ideas

- Put more functionality into ADDriver base class
  - Currently it does not do much, all code is in each driver for:
    - Doing callbacks to plugins
    - Processing new exposure time with writeFloat64 function
  - writeFloat64 in ADDriver base class would call setExposure() in derived class
  - Derived class would call ADDriver::doPluginCallbacks(), which would handle setting attributes, getting timestamp, calling plugins, etc.
- This is the way the Model 3 motor driver, which also uses asynPortDriver, is written
- Demultiplexor/multiplexor plugin
  - Allow multiple plugins to work on the same data stream when it saturates a single core

Future Ideas

- Extend areaDetector concepts to other types of detectors:
  - ADCs
  - Electrometers
  - Waveform digitizers
  - Oscilloscopes?
- They all produce 1-D (or 2-D for multi-channel inputs) arrays that could benefit from plugins for file saving, FFTs, ROI extraction, digital filtering, etc.
- We can’t currently use the areaDetector file plugins to save MCA data, for example.

Future Ideas

- Export NDArrays via EPICS V4
- David Hickin (DLS) has demonstrated:
  - A plugin that exports NDArrays as V4 objects over Channel Access
  - An ADDriver that receives the V4 objects on another machine and has its own set of plugins
- Allows using multiple machines, and multiple processes, not just multiple cores in a single IOC for plugin processing

areaDetector Collaboration

- The move to GitHub has really helped areaDetector become a collaborative effort
- Many more people are contributing via additions and bug fixes.
- Make changes in their fork on github and then issue a “pull request”.
- Collaboration meeting ~monthly on Google Hangout (U. Pedersen, M. Rivers, A. Glowacki, M. Pearson, M. Kraimer, N. Rees, D. Hickin, T. Cobb)
- In-person meetings ~2 times/year.
- Developed a road-map, following it pretty well.