Area detector

Wayne Lewis

Osprey DCS

2018-03-08

What is areaDetector?

- areaDetector supports a wide range of 2-D detectors and cameras
- easy integration of new detectors in framework
- includes many plugins for image processing

- https://github.com/areaDetector
- http://cars9.uchicago.edu/software/epics/areaDetector.html
- http://cars9.uchicago.edu/software/epics/areaDetectorDoc.html

What devices are supported?

- CCD detectors
- Pixel array detectors
- Cameras
- URLs

Starting IOC

- $ cd ~/build-epics
- $ cd iocs/simDetectorIOC/iocBoot/iocSimDetector
- $ ./start_epics

Plugins

- color conversion
- FFT
- file writing
- overlay
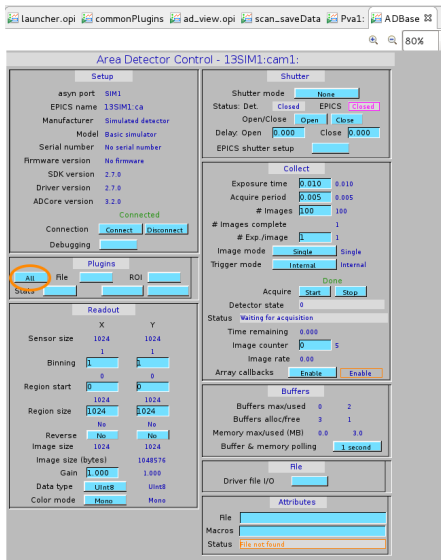- ROI (region of interest)
- statistics
- PVAccess
- and others...

- http://cars9.uchicago.edu/software/epics/pluginDoc.html

Enabling plugins

Figure 1: Accessing plugins control

Enabling plugins

Figure 2: Enabled plugins (without NDPluginPva loaded)

Adding NDPluginPva plugin to IOC

http://cars9.uchicago.edu/software/epics/NDPluginPva.html

Plugin is included in default build, but not enabled in startup script

Plugins are started by the `commonPlugins.cmd` file in the `ADCore/iocBoot` directory.

If necessary, copy from `EXAMPLE_commonPlugins.cmd` file.

Adding NDPluginPva plugin to startup

- Stop IOC (if running)
- `$ cd ~/build-epics`
- `$ cd areaDetector/ADCore/iocBoot`
- Edit `commonPlugins.cmd` and remove comment character (#) from three highlighted lines:

---

commonPlugins.cmd

```
...
# Optional:  load NDPluginPva plugin
NDPvaConfigure("PVA1", $(QSIZE), 0, "$(PORT)", 0,
$(PREFIX)Pva1:Image, 0, 0, 0)
dbLoadRecords("NDPva.template", "P=$(PREFIX),R=Pva1:,
PORT=PVA1,ADDR=0,TIMEOUT=1,NDARRAY_PORT=$(PORT)")
# Must start PVA server if this is enabled
startPVAServer
```

Starting IOC with NDPluginPva plugin

Exactly the same as starting IOC earlier. Changes in `commonPlugins.cmd` will add in PVAccess support.

- $ cd ~/build-epics
- $ cd iocs/simDetectorIOC/iocBoot/iocSimDetector
- $ ./start_epics

Enabling NDPluginPva plugin

Figure 3: NDPluginPva plugin enabled

Testing PVAccess link to image
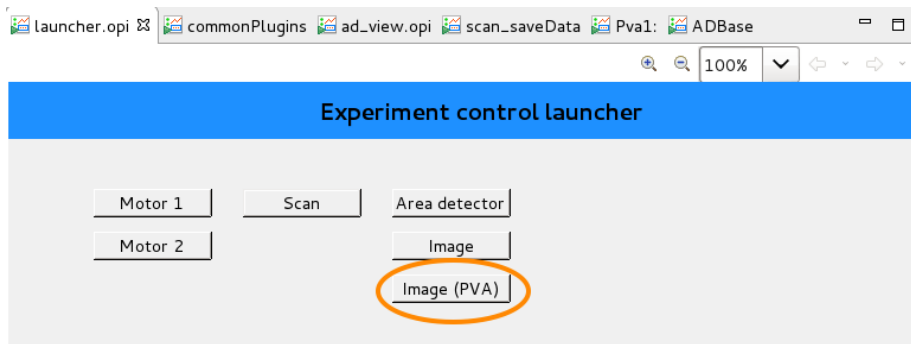
Figure 4: Link to PVAccess display of image

VImage widget
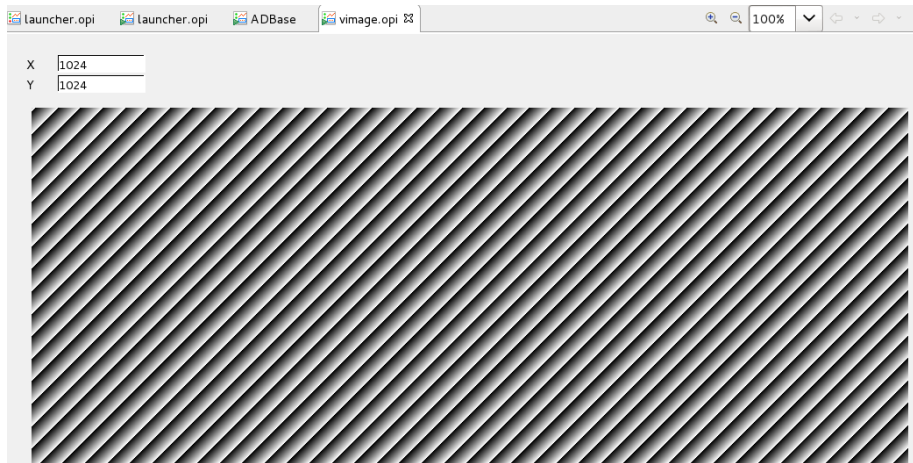
Figure 5: VImage widget display of AD image

Controlling PVAccess server

From IOC shell:
```
epics> stopPVAServer
epics> pvasr
PVA server not running
epics>
```

Controlling PVAccess server

```
epics> startPVAServer
2018-03-02T16:56:55.542 Using dynamically assigned TCP port
34289.
epics> pvasr
VERSION : pvAccess Server v6.0.1-SNAPSHOT
PROVIDER_NAMES : local,
BEACON_ADDR_LIST :
AUTO_BEACON_ADDR_LIST : 1
BEACON_PERIOD : 15
BROADCAST_PORT : 5076
SERVER_PORT : 34289
RCV_BUFFER_SIZE : 16384
IGNORE_ADDR_LIST:
INTF_ADDR_LIST : 0.0.0.0
epics>
```

Command line access to image

```
$ pvget 13SIM1:Pva1:Image
13SIM1:Pva1:Image
structure
union value
ubyte[]
[0,10,20,30,40,50,60,70,80,90 ...]

$ caget 13SIM1:Pva1:Image
Channel connect timed out:  '13SIM1:Pva1:Image' not found.

$ caget -# 10 13SIM1:image1:ArrayData
13SIM1:image1:ArrayData 10 0 10 20 30 40 50 60 70 80 90 ...
```

Demonstrations

- Detector triggering
- Single/multiple/continuous acquisition
- Exposure time vs acquire period
- Trigger modes
- Readout control - binning, region start, region size, reverse
- Gain, data type, color mode
- Shutter control
- Plugins

Plugin control

- Enabling
- Connecting
- Configuring

Plugin demonstration

- Process - background subtraction, flat field correction, scaling, clipping, filtering
- Transform - rotate and/or flip
- Overlay
- Region of interest
- Statistics

Other plugins

- Scatter/gather
- File writing
- FFT

Exercises

- Using plugins, rotate image and add rectangle from (700,700) to (800,800)
- Reverse the order of the above plugins and observe the difference
- Test pvget monitor of PVAccess image data (`13SIM1:Pva1:Image`)
- Use caput to set number of frames to acquire to and image mode to Multiple
- Use caput with put-callback to trigger detector and wait for completion (`caput -c -w 1000 <pv_name> <pv_value>`)
- Test whether pvput can be used for above two operations
- Use an ROI plugin to generate, and display in the CA display, the area from (0,0) to (200,200)