areaDetector Lab Handout

Mark Rivers, University of Chicago

January 23, 2015
10:00 AM
Advanced Photon Source
Room E1100/1200

Overview

The goal of this lab is to gain experience using areaDetector to control and display images from a simulation detector or from a GigE camera if you have one.

Setup

It is preferable to run your areaDetector IOC on your own laptop. The bandwidth requirements are rather large for fast frame rates, so sharing a single machine over the wireless network will likely lead to bottlenecks.

However, if you cannot run areaDetector on your own laptop you can run it on our Linux machine, corvette.cars.aps.anl.gov. The following are the instructions for using corvette to run areaDetector.

Linux host: corvette.cars.aps.anl.gov
Username: (given out in class)
Password: (given out in class)

Copy the areaDetector application

Once you are logged in go to the appropriate subdirectory, e.g. student1. Each student will have a different number

```
$ cd student1
```

Copy the areaDetector application from the /home/epics_class/teacher directory to this directory:

```
cp -rp ../teacher/areaDetector/ .
```

Change to the areaDetector/ directory:

```
$ cd areaDetector/
```

Build the areaDetector application

Edit the file `areaDetector/onfigure/RELEASE_PATHS.local` and change the following line:

```
AREA_DETECTOR= /corvette/home/epics_class/teacher/areaDetector
```

to

```
AREA_DETECTOR= /corvette/home/epics_class/student1/areaDetector
```

but replace student1 with your student number.

Change to the `/home/epics_class/student1/areaDetector/ADCore` directory.

Clean and build the source code (-s means build silently, only showing errors, and –j means parallel build, using many threads):

```
$ make -sj clean uninstall
$ make -sj
```

Customize the setup for your student number

Change to the following directory
`/home/epics_class/student1/areaDetector/ADCore/iocs/simDetectorIOC/iocBoot/iocSimDetector`

```
cd iocs/simDetectorIOC/iocBoot/iocSimDetector/
```

Edit st.cmd. Change this line:

```
epicsEnvSet("PREFIX", "SIM_1:")
```

The PV prefix must be unique for all students. Replace SIM_1: with your student number, i.e. SIM_2: for student 2, etc.

Running the simDetector IOC

These instructions are written for students running on corvette, but basically the same procedure should be followed if you are running locally on your laptop. If you don't have ImageJ installed you should do so, and copy the EPICS_areaDetector folder from areaDetector/Viewers/ImageJ into the plugins folder of your ImageJ installation.

Start MEDM

Type the following command, but replace SIM_1: with your student number, i.e. SIM_2:, SIM_3:, etc.

```
medm –x –macro “P=SIM_1:, R=cam1:” simDetector.adl &
```

Start ImageJ

To display the images you will run ImageJ. Do that with the following commands:

```
$ pushd /usr/local/ImageJ
$ ./run &
$ popd
```

Start the IOC

To start the IOC type the following commands:

```
$ cp envPaths envPaths.linux
$ ../../bin/linux-x86_64/simDetectorApp st.cmd.linux
```

Start the ImageJ plugin EPICS_areaDetector

In ImageJ go to `Plugins/EPICS_areaDetector/EPICS_AD_Viewer`. This will start the viewer. In the PVPrefix field enter your PV Prefix, e.g. SIM_2: for student 2, and hit Enter.

Make the IOC do callbacks to plugins, and enable the NDStdArrays plugin

In the MEDM simDetector.adl screen make sure that ArrayCallbacks are Enabled.

Make sure that the NDStdArrays plugin is enabled. Open Plugins/Other/Image #1, which is the NDStdArrays plugin. Change the Enable PV to Enable.

At this point you should see an image window open, and images displayed on your screen.

Lab exercise #1 Establish communication.

1) Start the simDetector and Viewer as described above. Adjust the exposure time and see how the frame rate varies and images change.

Lab exercise #2 Display the images

1) Change the exposure time.
2) Change AcquirePeriod to change the frame rate at the same exposure time
3) Select a region of interest on the detector
4) Change the data type
5) Change the color mode
6) Make a dynamic line profile through the data with ImageJ
7) Capture images to a movie with ImageJ

Lab exercise #3 Work with plugins

1) NDPluginStats
    a. Display the basic statistics on your images
    b. Display the centroid
    c. Display row and column profiles
    d. Display histogram
2) NDPluginFile
    a. Save individual TIFF files
    b. Save a streamed netCDF file
    c. Read the netCDF file back into ImageJ with the netCDF file reader
    d. Only save a file once per second
3) NDPluginProcess
    a. High clip, low clip
    b. Average frames
    c. Sum of frames
    d. Difference of frames
4) NDPluginROI
    a. Select a Region of Interest
    b. Feed that ROI into the plugins – file, stats, etc.

Lab exercise #4 Work with attributes

1) Create an XML attributes file. Start with iocSimDetector/simDetectorAttributes.xml. Assign some other driver attributes and EPICS PVs
2) Open that file in the simDetector driver
3) Save data to the netCDF plugin in Stream mode. Save 100 frames.
4) Dump the netCDF file, displaying the attributes. Here is an example dump of the header information and the Attr_RingCurrent variable, which is one of the attributes.

```
[epics_class@corvette ~/student1]$ ncdump -v Attr_RingCurrent test_034.nc
```

```
netcdf test_034 {
dimensions:
        numArrays = UNLIMITED ; // (20 currently)
        dim0 = 256 ;
        dim1 = 256 ;
        attrStringSize = 256 ;
variables:
        int uniqueId(numArrays) ;
        double timeStamp(numArrays) ;
        byte array_data(numArrays, dim0, dim1) ;
        int Attr_ColorMode(numArrays) ;
        byte Attr_AcquireTime(numArrays) ;
        double Attr_RingCurrent(numArrays) ;
        char Attr_RingCurrent_EGU(numArrays, attrStringSize) ;
        double Attr_ID_Energy(numArrays) ;
        char Attr_ID_Energy_EGU(numArrays, attrStringSize) ;
        int Attr_ImageCounter(numArrays) ;
        int Attr_MaxSizeX(numArrays) ;
        int Attr_MaxSizeY(numArrays) ;
        char Attr_CameraModel(numArrays, attrStringSize) ;
        char Attr_CameraManufacturer(numArrays, attrStringSize) ;

// global attributes:
                :dataType = 1 ;
                :NDNetCDFFileVersion = 3. ;
                :numArrayDims = 2 ;
                :dimSize = 256, 256 ;
                :dimOffset = 0, 0 ;
                :dimBinning = 1, 1 ;
                :dimReverse = 0, 0 ;
                :Attr_ColorMode_DataType = "Int32" ;
                :Attr_ColorMode_Description = "Color mode" ;
                :Attr_ColorMode_Source =  ;
                :Attr_ColorMode_SourceType = "Driver" ;
                :Attr_AcquireTime_DataType = "Int8" ;
                :Attr_AcquireTime_Description = "Camera acquire time" ;
                :Attr_AcquireTime_Source = "13SIM1:cam1:AcquireTime" ;
                :Attr_AcquireTime_SourceType = "EPICS_PV" ;
                :Attr_RingCurrent_DataType = "Float64" ;
                :Attr_RingCurrent_Description = "Storage ring current" ;
                :Attr_RingCurrent_Source = "S:SRcurrentAI" ;
                :Attr_RingCurrent_SourceType = "EPICS_PV" ;
                :Attr_RingCurrent_EGU_DataType = "String" ;
                :Attr_RingCurrent_EGU_Description = "Storage ring current units" ;
                :Attr_RingCurrent_EGU_Source = "S:SRcurrentAI.EGU" ;
                :Attr_RingCurrent_EGU_SourceType = "EPICS_PV" ;
                :Attr_ID_Energy_DataType = "Float64" ;
                :Attr_ID_Energy_Description = "Undulator energy" ;
                :Attr_ID_Energy_Source = "ID34:Energy" ;
                :Attr_ID_Energy_SourceType = "EPICS_PV" ;
                :Attr_ID_Energy_EGU_DataType = "String" ;
                :Attr_ID_Energy_EGU_Description = "Undulator energy units" ;
                :Attr_ID_Energy_EGU_Source = "ID34:Energy.EGU" ;
                :Attr_ID_Energy_EGU_SourceType = "EPICS_PV" ;
                :Attr_ImageCounter_DataType = "Int32" ;
                :Attr_ImageCounter_Description = "Image counter" ;
                :Attr_ImageCounter_Source = "ARRAY_COUNTER" ;
                :Attr_ImageCounter_SourceType = "Param" ;
                :Attr_MaxSizeX_DataType = "Int32" ;
```

```
                :Attr_MaxSizeX_Description = "Detector X size" ;
                :Attr_MaxSizeX_Source = "MAX_SIZE_X" ;
                :Attr_MaxSizeX_SourceType = "Param" ;
                :Attr_MaxSizeY_DataType = "Int32" ;
                :Attr_MaxSizeY_Description = "Detector Y size" ;
                :Attr_MaxSizeY_Source = "MAX_SIZE_Y" ;
                :Attr_MaxSizeY_SourceType = "Param" ;
                :Attr_CameraModel_DataType = "String" ;
                :Attr_CameraModel_Description = "Camera model" ;
                :Attr_CameraModel_Source = "MODEL" ;
                :Attr_CameraModel_SourceType = "Param" ;
                :Attr_CameraManufacturer_DataType = "String" ;
                :Attr_CameraManufacturer_Description = "Camera manufacturer" ;
                :Attr_CameraManufacturer_Source = "MANUFACTURER" ;
                :Attr_CameraManufacturer_SourceType = "Param" ;
data:

 Attr_RingCurrent = 102.507117432379, 102.507117432379, 102.506850152379,
    102.506850152379, 102.506850152379, 102.506850152379, 102.506850152379,
    102.504344352379, 102.504344352379, 102.504344352379, 102.504344352379,
    102.504344352379, 102.502845992379, 102.502845992379, 102.502845992379,
    102.502845992379, 102.502845992379, 102.501622832379, 102.501622832379,
    102.501622832379 ;
}
```