

APS Classes on areaDetector and asyn

Mark Rivers, University of Chicago

January 22-30, 2015

Advanced Photon Source

Room E1100/1200

Thursday, January 22

Using areaDetector 1 – Lecture and Demonstration

10:00-12:00 Lecture on areaDetector (Overview: detector drivers, plugins, and viewers)

12:00-1:00 Lunch break

1:00-3:00 Demonstration with Prosilica GigE camera

Friday, January 23

Using areaDetector 2 – Hands-On Laboratory

10:00-12:00 Hands-on lab with simulation and/or Prosilica cameras, part 1

12:00-1:00 Lunch break

1:00-3:00 Hands-on lab with simulation and/or Prosilica cameras, part 2

Monday, January 26

Asyn Driver 1 – Lecture and Demonstration

10:00-12:00 Lecture on asyn

12:00-1:00 Lunch break

1:00-3:00 Tutorial using Measurement Computing USB-1608GX-2AO driver as an example

Tuesday, January 27
Asyn Driver 2 – Hands-On Laboratory

10:00-12:00 Hands-on lab writing/modifying writing a driver to talk to a simple device, part 1

12:00-1:00 Lunch break

1:00-3:00 Hands-on lab writing/modifying writing a driver to talk to a simple device, part 2

Thursday, January 29
areaDetector Programming 1 – Writing Detector Drivers

10:00-12:00 Lecture on writing areaDetector drivers

12:00-1:00 Lunch break

1:00-3:00 Hands-on lab modifying the simulator detector driver to talk to add new features

Friday, January 30
areaDetector Programming 2 – Writing Plugins

10:00-12:00 Lecture on writing areaDetector plugins

12:00-1:00 Lunch break

1:00-3:00 Hands-on lab writing a new file plugin to save data in ASCII file format.

Background Materials and Instructions

Using areaDetector 1 (January 22)

It would be helpful to look at the areaDetector documentation here:

<http://cars.uchicago.edu/software/epics/areaDetectorDoc.html>

Using areaDetector 2 (January 23)

The preferred method for the areaDetector hands-on laboratory is to install areaDetector on your laptop. Prebuilt versions of areaDetector/ADCore R2-1 for Linux and Windows can be obtained here:

<http://cars.uchicago.edu/software/pub/ADCore>

Alternatively you can build areaDetector/ADCore from source code as described here:

https://github.com/areaDetector/areaDetector/blob/master/INSTALL_GUIDE.md

You should install medm, EDM, CSS Boy, or caQtDM in order to have an OPI to control the detector. medm is preferred because it is the best tested and debugged.

You should also install ImageJ, and install the areaDetector plugin for ImageJ. This is documented here:

<http://cars.uchicago.edu/software/epics/areaDetectorViewers.html#ImageJViewer>

If possible you should test that you can run the simulation detector and display the images on your laptop before the class. But if you have difficulty with this you can get assistance at the class.

If you have a real detector, such as a Prosilica GigE camera or other portable device that would like to use instead of the simulation detector please bring it!

If you cannot get a local IOC running on your laptop then you will be able to run an IOC on corvette.cars.aps.anl.gov. Follow the instructions for the Asyn Driver 2 class below. You will be able to run the IOC, medm, and ImageJ on corvette, and display with X11 on your laptop. However, the performance will suffer, particularly if many students are doing this. Alternatively you can work in a small group with someone else who does have a local installation of areaDetector.

Asyn Driver 1 (January 26)

For the asyn class familiarity with C programming is needed. Familiarity with C++ is very helpful, at least with the basic ideas of classes, constructors and destructors, virtual functions, and template functions. A nice introduction to C++ can be found here: <http://www.cplusplus.com/doc/tutorial/>.

It would be helpful to look at the asyn documentation here:

<http://www.aps.anl.gov/epics/modules/soft/asyn/R4-25/asynDriver.html>

<http://www.aps.anl.gov/epics/modules/soft/asyn/R4-25/asynPortDriver.html>

Asyn Driver 2 (January 27)

This lab will involve writing a driver based on asynPortDriver. The device to be controlled will be an emulator of the CaenEls AH401D Electrometer, which is used for beam position monitoring. The development will be done on our Linux machine, corvette.cars.aps.anl.gov. There is one account for use by all students. The login is epics_class, password=epics_student. Each student has his/her own subdirectory where they will be working. This directory will have an EPICS development environment installed. Standard Linux editors (emacs, gedit, nedit, vi) are available. Students can either work alone or in small groups.

You should bring a laptop to the class with the following capabilities:

- X11 server
- ssh client configured to tunnel X11

The ssh/X11 connection should be tested prior to the class by logging into corvette using the credentials above and typing the command “xclock &”. If you see the xclock application then things are configured correctly.

If you have a real device that you want to bring to the lab for development, please bring it! You will need to bring a computer that has an EPICS development system installed, including asyn R4-25.

areaDetector Programming 1 and 2 (January 29 & 30)

You can install a development system for areaDetector/ADCore on your laptop as described here: https://github.com/areaDetector/areaDetector/blob/master/INSTALL_GUIDE.md

Alternatively you can work on corvette.cars.aps.anl.gov following the instructions for Asyn Driver 2 above. Since most time will be in programming rather than running the detector IOC the performance should be OK even if most students work on corvette.