OAK RIDGE National Laboratory

ASYN

Oct. 2018

Kay Kasemir, based on slides by
Mark Rivers, Eric Norum, Marty Kraimer

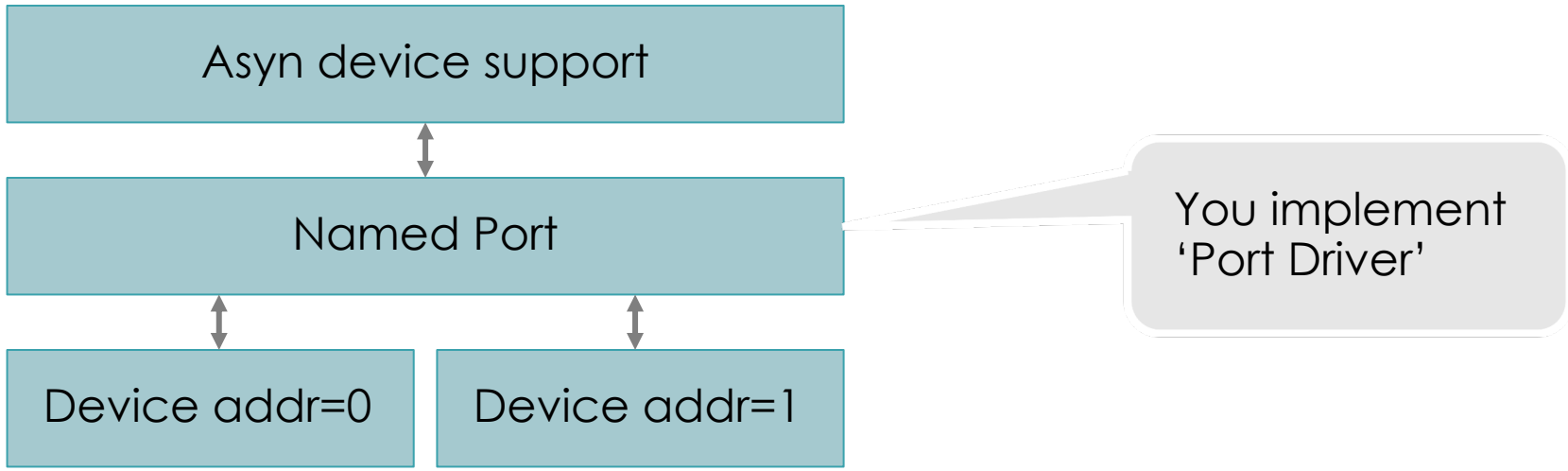
ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

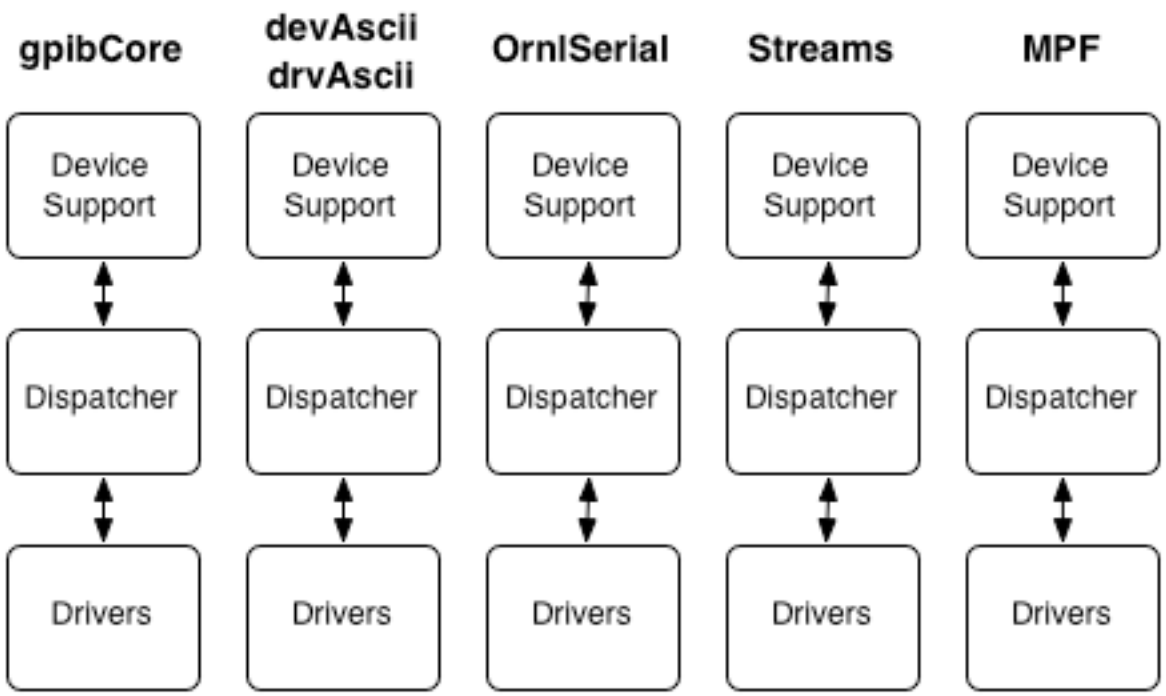
asyn: Helper for writing device support

Writing robust, non-blocking device support is hard.

- For string-based protocol, use ‘stream’ device
- Otherwise, consider using ‘asyn’

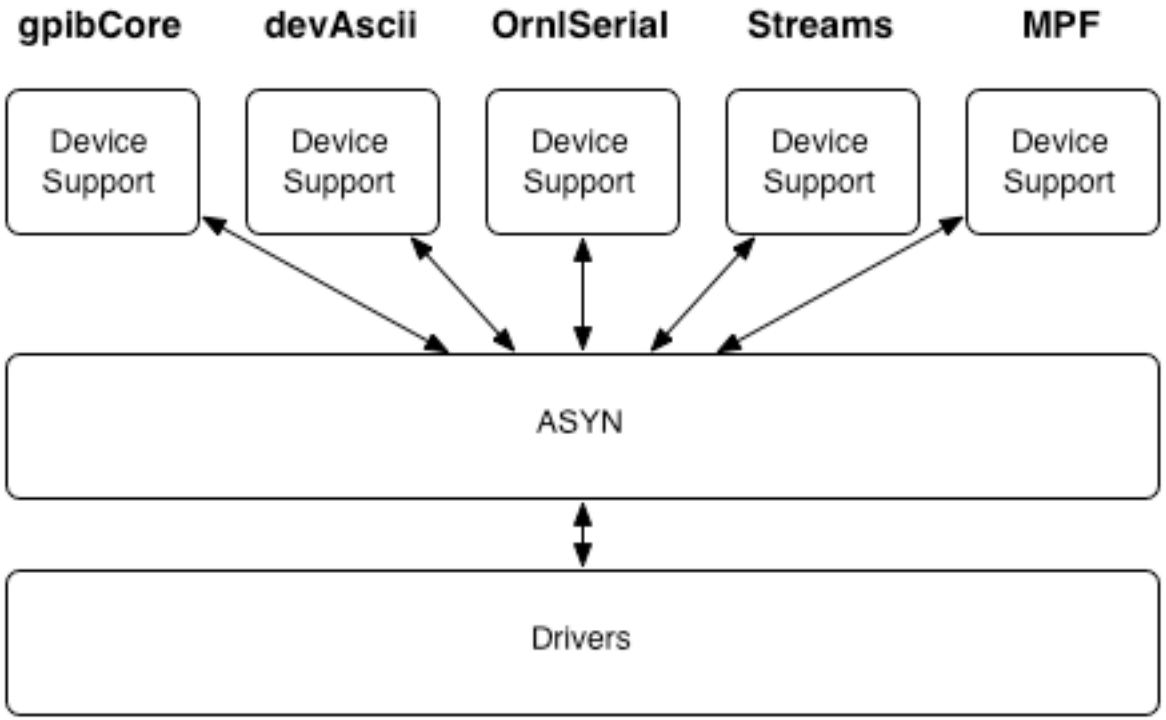


Motivation for asyn



- Separate (incompatible) sets of drivers and device supports.
- Much effort duplicated but different sets of features.

asyn goal



- Every device supports works with every driver.
- Much work went to ASYN, less work to do for drivers.

Vocabulary: Port

- Communication path (“bus”) with unique name.
- One or many devices can be connected.
- May have addresses to identify individual devices.
- May be blocking or non-blocking.
- Is configured in startup script.

```
drvAsynSerialPortConfigure "COM2", "/dev/sttyS1"
drvAsynIPPortConfigure "fooServer", "192.168.0.10:40000"
vxi11Configure "LanGpib1", "192.168.0.1", 1, 1000, "hpib"
myDeviceDriverConfigure "portname", parameters
```

Vocabulary: Interface

- API for a class of ports.
  - common, message based, register based, …
- Defines table of driver functions (“methods”)
- Does not implement driver methods.
- Every port has one or many interfaces.
- Clients talk to interfaces, not to drivers.

```
pasynCommon->connect()
pasynOctet->write()
```

Vocabulary: Driver

- Software to handle one type of ports.
- Implements one or many interfaces.
  - Provides method tables for interfaces.
  - Has internal knowledge about specific port hardware.
- Does not handle any specific device type!
- Examples:
  - serial bus, VXI-11, Green Springs IP488, …
- Configure function in startup script connects driver to port.

Vocabulary: asynUser

- Identifies the client.
- Each client needs one asynUser.
- From asynDriver's point of view, asynUser *is* the client.
- "Handle" to ports and everything else inside asynDriver.

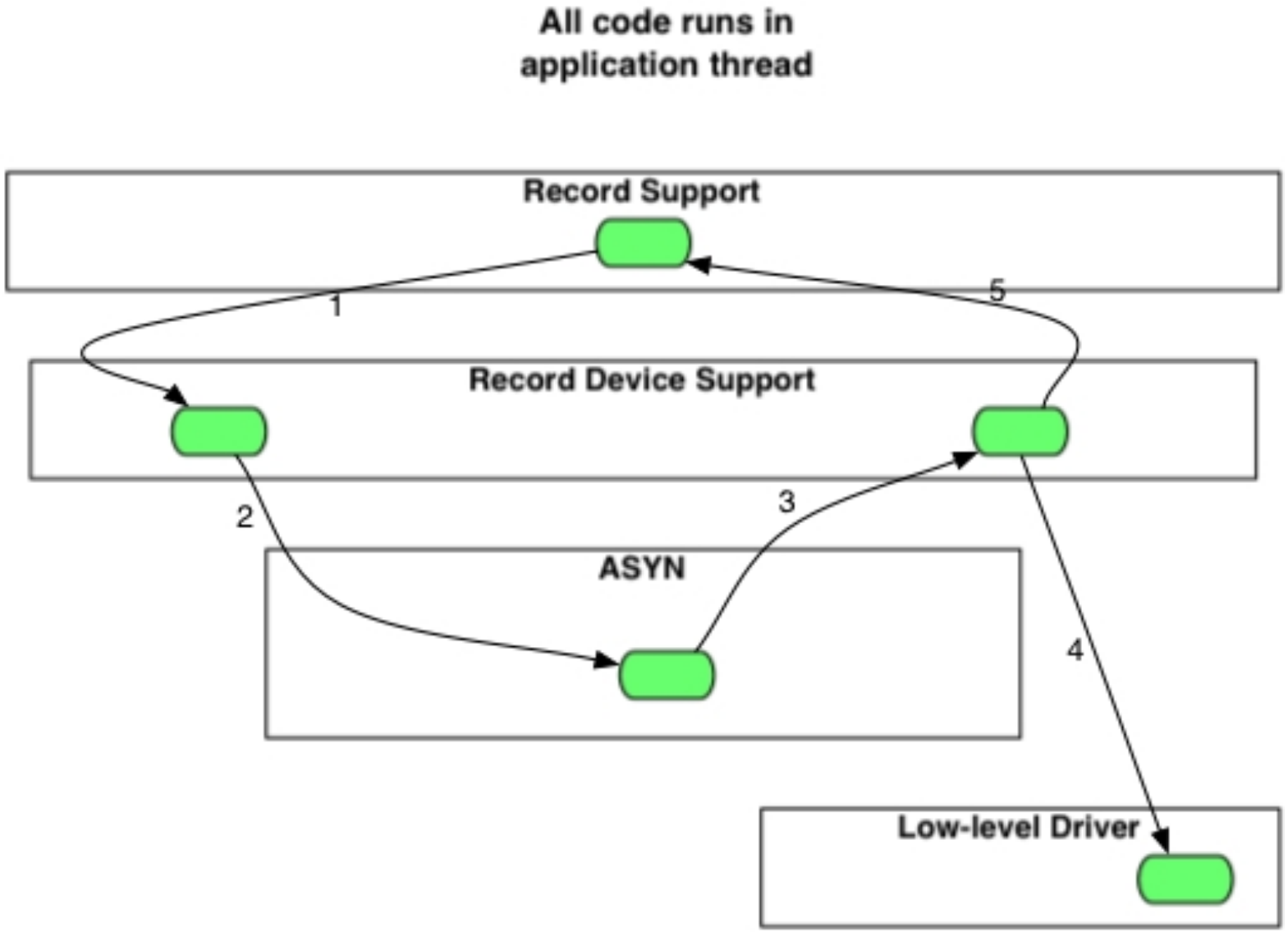
Vocabulary: asynManager

- Core of asynDriver.
- Creates threads for blocking ports.
- Registers and finds ports and interfaces.
- Schedules access to ports.
- There is exactly one global instance: `pasynManager`
- Clients ask asynManager for services

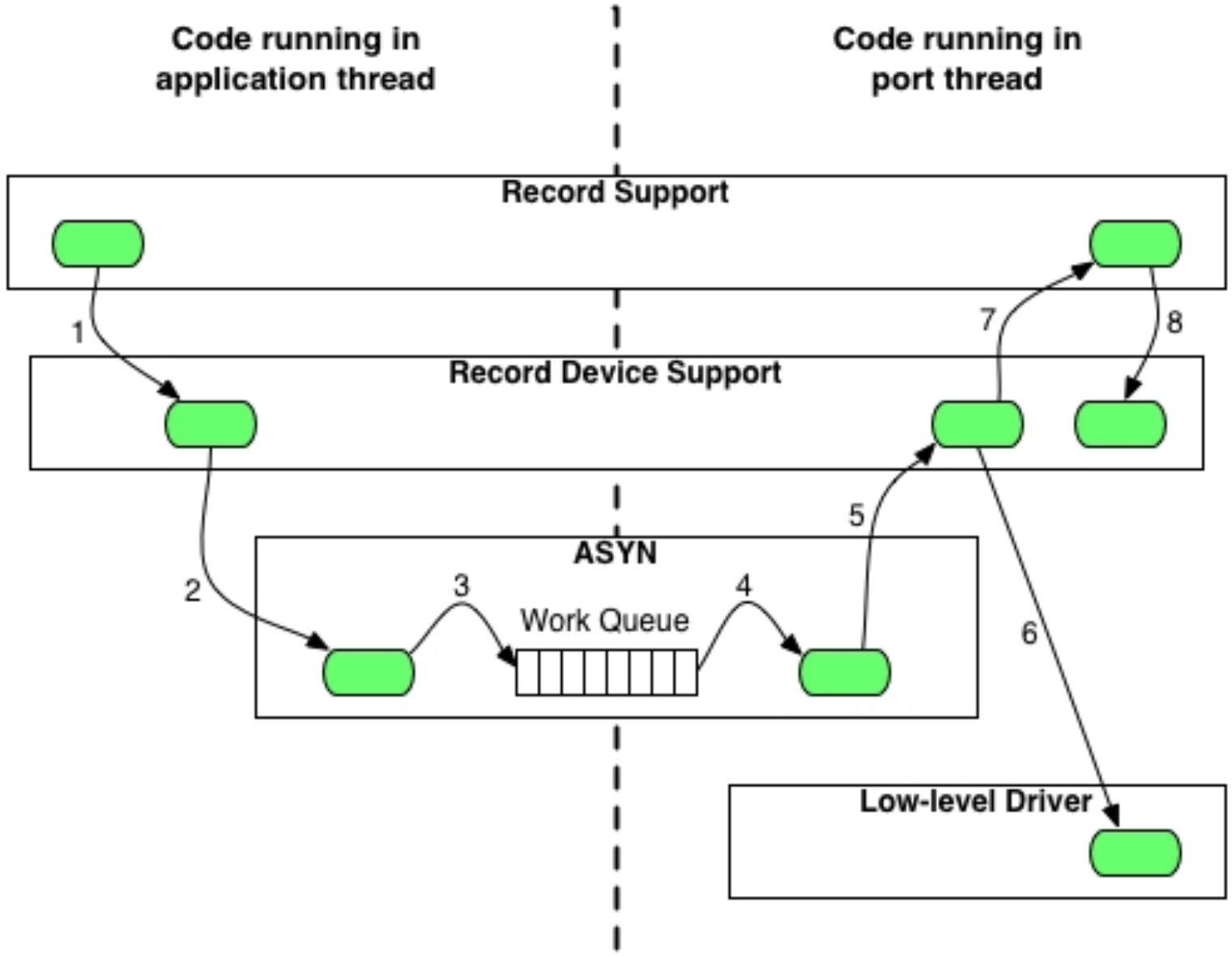
```
pasynManager->connectDevice(pasynUser , "portname", address)
pasynManager->findInterface(pasynUser, interfaceType, ...)
pasynManager->queueRequest(pasynUser, priority, timeout)
```

- Drivers inform asynManager about any important things.

Control Flow: ‘Synchronous’ for fast devices



Control Flow: ‘Asynchronous’ for slow devices



Example: “Port Driver”

epics-train/examples/asynDemoApp/src:
asynDemo.h and asynDemo.cpp

‘Port’ driver that creates a sawtooth-type ramp

- ‘RANGE’ to define upper limit
- ‘UPDATE_TIME’ to define update period in seconds
- ’VALUE’ for the value
- Thread that keeps updating the value

Example: Asyn Database & IOC

epics-train/examples/asynDemoApp/Db/asynDemo.db

Records that use ‘DTYP: asyn*’
to read/write port parameters

epics-train/examples/iocBoot/iocAsyn:

st.cmd to start IOC

Check ‘asynReport 10’

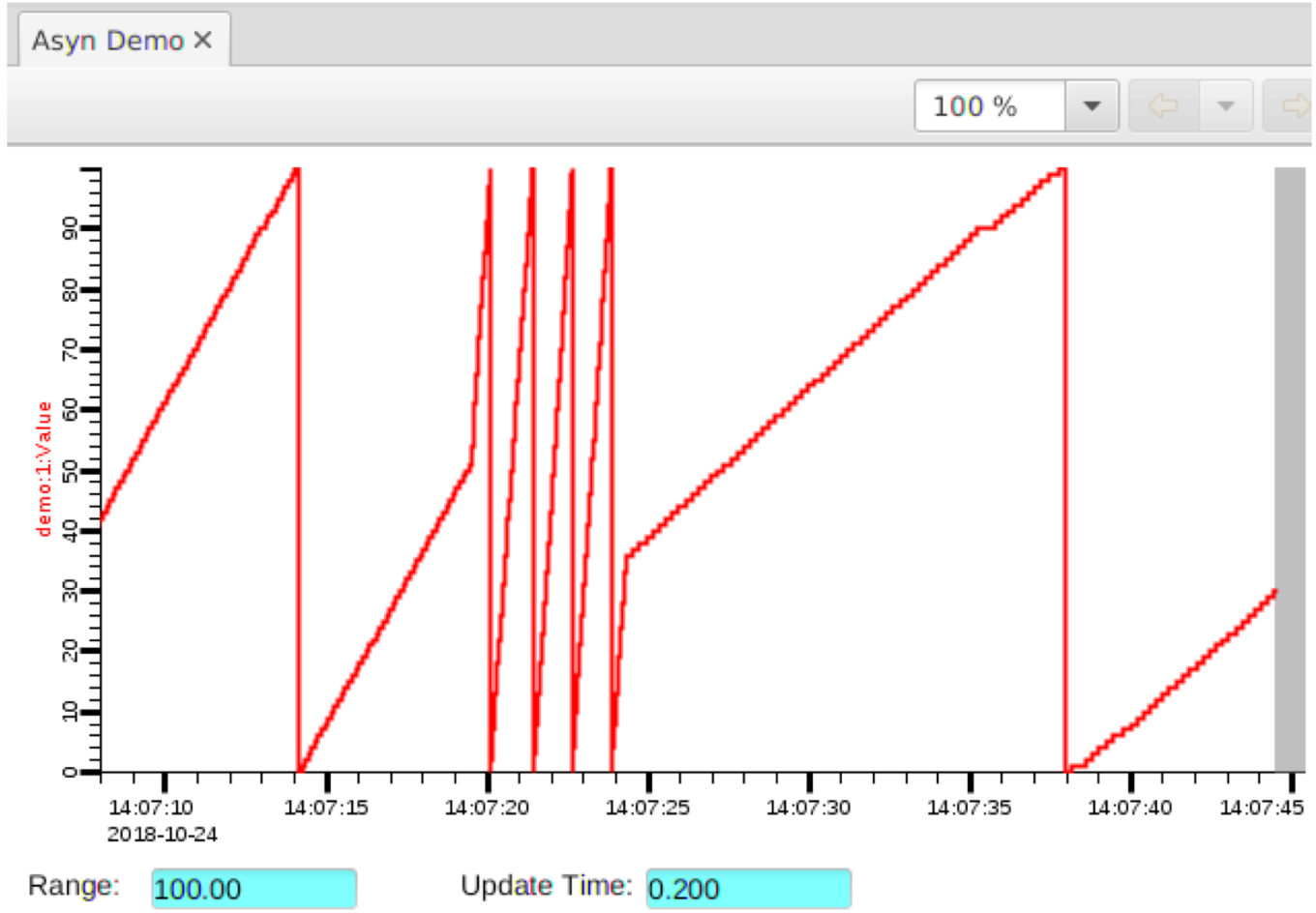
Example: Display

epics-train/examples/asynDemoApp/opi

Change range and

update time.

Update times
may be faster than 0.1 sec



There is much more

- Trace options to debug processing
- Support for serial ports, TCP, UDP, GPIB, Vxi11 (GPIB over Ethernet), USP Test-and-Measurement

Summary

- Many EPICS device support modules now use asyn
  - Motors, area detector, stream
- To interface new devices, consider Asyn