# An EPICS IOC Builder

M.G. Abbott, T. Cobb, Diamond Light Source, Oxfordshire, UK

The IOC builder assembles a complete IOC and related components, including IOC specific control screens, from a high level description.

The IOC description is either a Python script or an XML file specifying a list of components with instructions on how to integrate them.

The most important aspect of the IOC builder is the set of component descriptions: for each high level component ("module") there is a builder definition file specifying precisely how to initialise and use the interfaces provided by that module.

For example below we see module definitions for the procServ, eurotherm, Busy and 8515 modules, showing how template definitions and hardware initialisation instructions are specified using the IOC builder framework.

The IOC builder is particularly convenient for building large numbers of IOCs, particularly complex IOCs, or IOCs using components with complex initialisation requirements.

## Module definitions

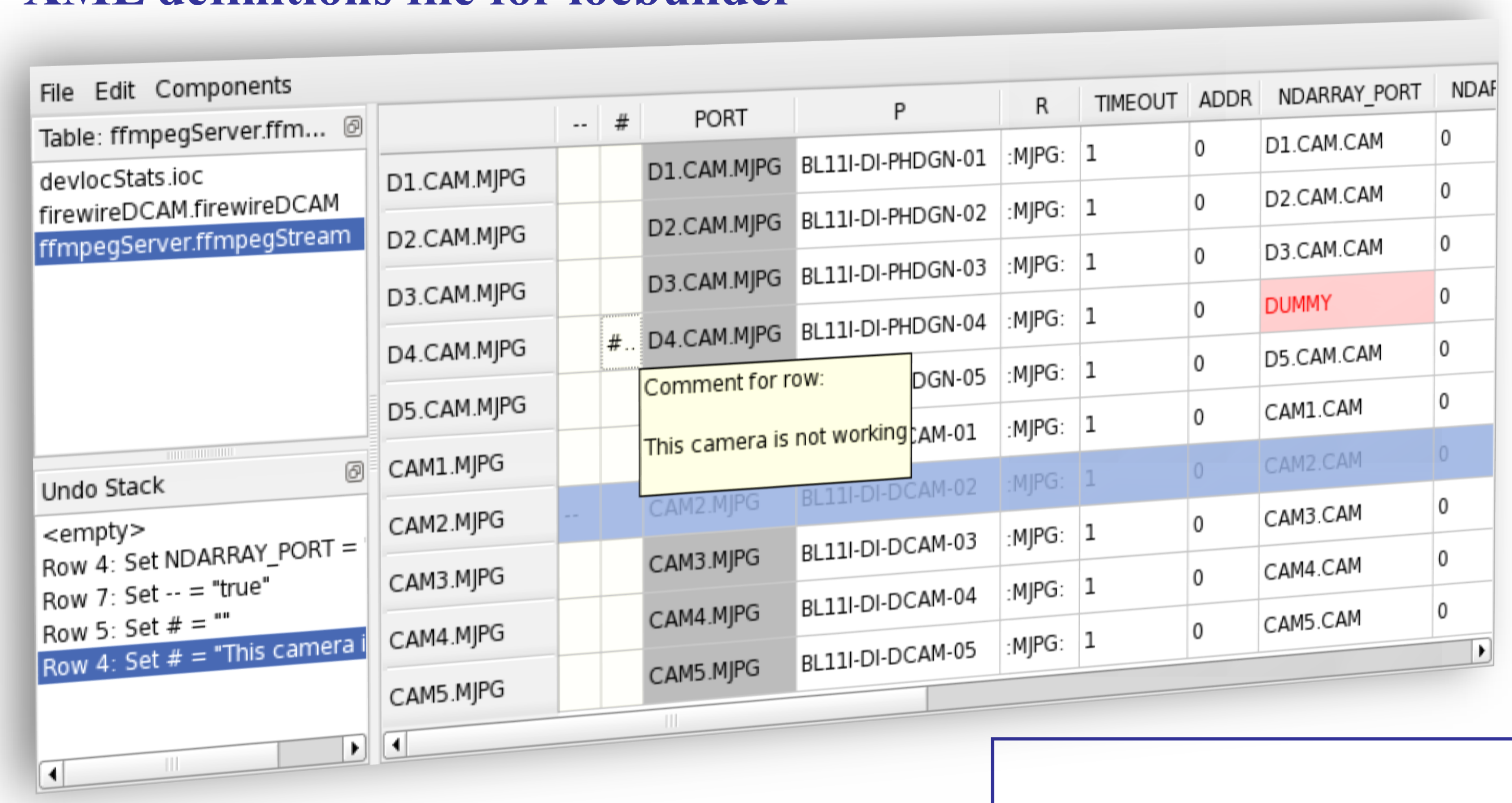
```python
from iocbuilder import AutoSubstitution, Device
from iocbuilder.modules.asyn import Asyn
from iocbuilder.modules.seq import Seq
from iocbuilder.modules.busy import Busy

class procServControl(AutoSubstitution, Device):
    Dependencies = (Asyn,Seq,Busy)
    LibFileList = ["procServControl"]
    DbdFileList = ["procServControl"]
    TemplateFile = 'procServControl.template'

    def PostIocInitialise(self):
        print 'seq(procServControl,"P=%(P)s")' % self.args
```

```python
from iocbuilder import Device

class Busy(Device):
    LibFileList = ['busy']
    DbdFileList = ['busySupport']
    AutoInstantiate = True
```

```python
from iocbuilder import AutoSubstitution
from iocbuilder.modules.streamDevice import AutoProtocol

class eurotherm2k(AutoSubstitution, AutoProtocol):
    TemplateFile = 'eurotherm2k.template'
    ProtocolFiles = ['eurotherm2k.proto']
```

```python
from iocbuilder import Device, SetSimulation
from iocbuilder.arginfo import *
from iocbuilder.modules.ipac import IpDevice, IpCarrier

class DLS8515(IpDevice):
    def __init__(self, carrier, ipslot, prefix="ty"):
        self.prefix = prefix
        self.configString = "DLS%sConfigure" % self.cardType
        self.__super.__init__(carrier, ipslot, None)
    ArgInfo = makeArgInfo(__init__,
        carrier = Ident ('Carrier card', IpCarrier),
        ipslot  = Simple('IP slot in carrier', int),
        prefix  = Simple('Prefix to create serial ports as, ' \
            'e.g. "ty" to create "/ty/70/0".'))
    def Initialise(self):
        print '%(configString)s(%(cardid)d, %(IPACid)s, %(vector)d, ' \
            '"%(prefix)s")' % self.__dict__
    def InitialiseOnce(self):
        print '# %(configString)s(card, carrier, vector, prefix)' \
            % self.__dict__
    LibFileList = ['DLS8515']
    DbdFileList = ['DLS8515']
    cardType = "8515"
```

## XML definitions file for iocbuilder



devIocStats.ioc
firewireDCAM.firewireDCAM
ffmpegServer.ffmpegStream

| | -- | # | PORT | P | R | TIMEOUT | ADDR | NDARRAY_PORT | NDAF |
|---|---|---|---|---|---|---|---|---|---|
| D1.CAM.MJPG | | | D1.CAM.MJPG | BL11I-DI-PHDGN-01 | :MJPG: | 1 | 0 | D1.CAM.CAM | 0 |
| D2.CAM.MJPG | | | D2.CAM.MJPG | BL11I-DI-PHDGN-02 | :MJPG: | 1 | 0 | D2.CAM.CAM | 0 |
| D3.CAM.MJPG | | | D3.CAM.MJPG | BL11I-DI-PHDGN-03 | :MJPG: | 1 | 0 | D3.CAM.CAM | 0 |
| D4.CAM.MJPG | | #.. | D4.CAM.MJPG | BL11I-DI-PHDGN-04 | :MJPG: | 1 | 0 | DUMMY | 0 |
| D5.CAM.MJPG | | | | DGN-05 | :MJPG: | 1 | 0 | D5.CAM.CAM | 0 |
| CAM1.MJPG | | | | AM-01 | :MJPG: | 1 | 0 | CAM1.CAM | 0 |
| CAM2.MJPG | -- | | CAM2.MJPG | BL11I-DI-DCAM-02 | :MJPG: | 1 | 0 | CAM2.CAM | 0 |
| CAM3.MJPG | | | CAM3.MJPG | BL11I-DI-DCAM-03 | :MJPG: | 1 | 0 | CAM3.CAM | 0 |
| CAM4.MJPG | | | CAM4.MJPG | BL11I-DI-DCAM-04 | :MJPG: | 1 | 0 | CAM4.CAM | 0 |
| CAM5.MJPG | | | CAM5.MJPG | BL11I-DI-DCAM-05 | :MJPG: | 1 | 0 | CAM5.CAM | 0 |

Comment for row:

This camera is not working

```
<empty>
Row 4: Set NDARRAY_PORT =
Row 7: Set -- = "true"
Row 5: Set # = ""
Row 4: Set # = "This camera i
```

## Python file representing an IOC

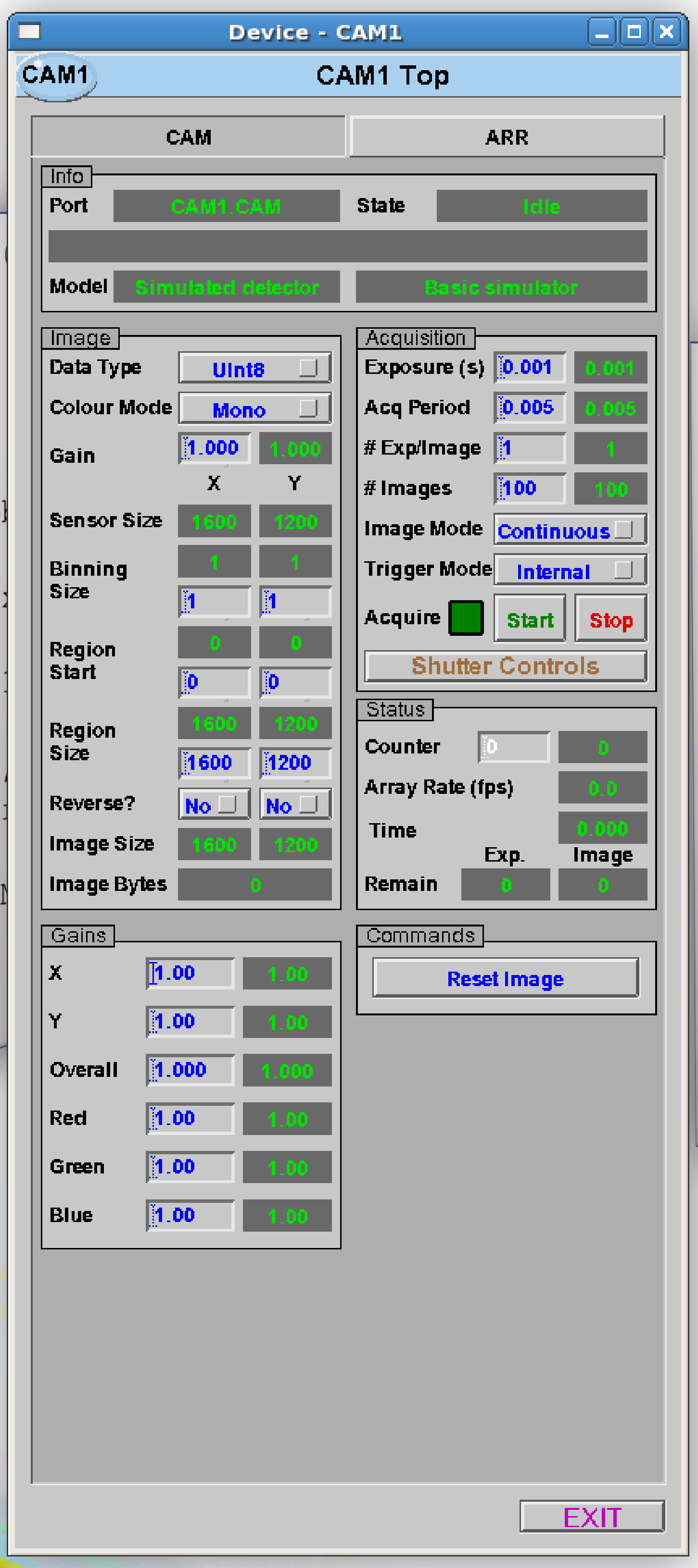
```python
card6 = ipac.Hy8002(6)
pmt_dac = card6.Hy8402(0)
pmt_adcs = [
    card6.Hy8401(i, intEnable=1, sampleSize=signals.PMTsampleSize)
    for i in (1,2)]
pmt_dac_channels = map(pmt_dac.channel, range(16))
pmt_adc_channels = [
    pmt_adcs[i].channel(j) for i in range(2) for j in range(8)]
card7 = ipac.Hy8001(7, ipac.DIRECTION_OUTPUT, invertout=1)
screen_outputs = card7.register(32, 16)
ict_controls = (
    card7.register(48, 8), card4.register(32, 8), card4.register(40, 8))
```

**iocbuilder**

## A full EPICS IOC

```
TOP = ../..
include $(TOP)/configure/CONFIG

PROD_IOC = simDetector
DBD += simDetector.dbd
simDetector_DBD += base.dbd
simDetector_DBD += asyn.dbd
simDetector_DBD += busySuppor
simDetector_DBD += ADSupport.
simDetector_DBD += NDPluginSu
simDetector_DBD += simDetecto
simDetector_LIBS += simDetect
simDetector_LIBS += NDPlugin
simDetector_LIBS += ADBase
simDetector_LIBS += netCDF
simDetector_LIBS += hdf5
simDetector_LIBS += PvAPI
simDetector_LIBS += GraphicsM
simDetector_LIBS += GraphicsM
simDetector_LIBS += GraphicsM
simDetector_LIBS += busy
simDetector_LIBS += asyn
simDetector_SYS_LIBS += tiff
simDetector_SYS_LIBS += jpeg
simDetector_SYS_LIBS += z
...
```

```
epicsEnvSet "EPICS_CA_MAX_ARRAY_BYTES", '4
epicsEnvSet "EPICS_TS_MIN_WEST", '0'

cd "$(INSTALL)"

dbLoadDatabase "dbd/simDetector.dbd"
simDetector_registerRecordDeviceDriver(pdbb

# simDetectorConfig(portName, maxSizeX, max
dataType, maxBuffers, maxMemory)
simDetectorConfig("CAM1.CAM", 2048, 1600, 1

# NDStdArraysConfigure(portName, queueSize,
blockingCallbacks, NDArrayPort, NDArrayAddr
maxMemory)
NDStdArraysConfigure("CAM1.ARR", 2, 0, "CAM
0, -1)

dbLoadRecords 'db/simDetector_expanded.db'
iocInit
```



```
Port          CAM1.CAM              State   Idle
Model         Simulated detector            Basic simulator

Data Type     UInt8                 Exposure (s)   0.001   0.001
Colour Mode   Mono                  Acq Period     0.005   0.005
Gain          1.000   1.000         # Exp/Image    1       1
              X       Y             # Images       100     100
Sensor Size   1600    1200          Image Mode     Continuous
Binning       1       1             Trigger Mode   Internal
Size          1       1             Acquire
Region        0       0
Start         0       0             Counter        0       0
Region        1600    1200          Array Rate (fps)       0.0
Size          1600    1200          Time                   0.000
Reverse?      No      No                           Exp.    Image
Image Size    1600    1200          Remain         0       0
Image Bytes   0

X             1.00    1.00
Y             1.00    1.00
Overall       1.000   1.000
Red           1.00    1.00
Green         1.00    1.00
Blue          1.00    1.00
```

```
SUPPORT = /dls_sw/prod/R3.14.11/support
WORK = /dls_sw/work/R3.14.11/support
AREADETECTOR = $(SUPPORT)/areaDetector/1-7beta1-dls3
ASYN = $(SUPPORT)/asyn/4-14
BUSY = $(SUPPORT)/busy/1-3dls3
# EPICS Base appears last
EPICS_BASE = /dls_sw/epics/R3.14.11/base
```

```
file $(AREADETECTOR)/db/ADBase.template
{
pattern { P, R, PORT, TIMEOUT, ADDR }
    { "SIMDETECTOR", ":CAM:", "CAM1.CAM", "1", "0" }
}

file $(AREADETECTOR)/db/simDetector.template
{
pattern { P, R, PORT, TIMEOUT, ADDR }
    { "SIMDETECTOR", ":CAM:", "CAM1.CAM", "1", "0" }
}

file $(AREADETECTOR)/db/NDPluginBase.template
{
pattern { P, R, PORT, TIMEOUT, ADDR, NDARRAY_PORT, NDARRAY_ADDR }
    { "SIMDETECTOR", ":ARR:", "CAM1.ARR", "1", "0", "CAM1.CAM", "0"}
}
...
```

For more information please contact michael.abbott@diamond.ac.uk