

# Enhancements to motor record device and driver support

Peter Denison  
Diamond Light Source



EPICS Collaboration, APS, June 2006



# Contents

- Recap of Aims
- Architecture
- Example
- Progress
- Future



# Recap of Aims

- Simpler to write motor controller driver
- Extensible API
- Allow trivial use with standard records in base
- Start with support for OMS, PMAC, MM4000

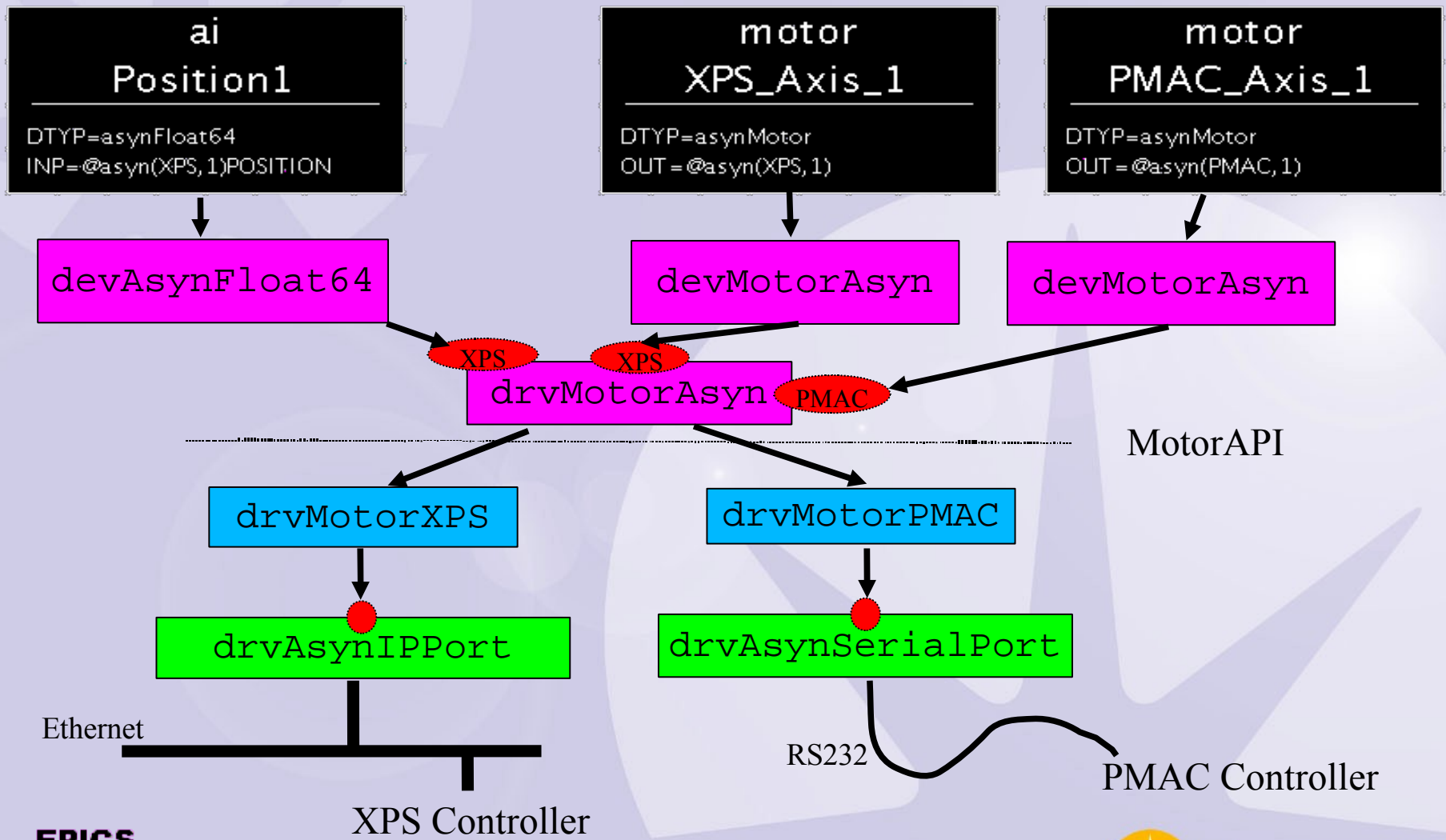
# Architecture

- Single device support for motor record
- Single Asyn-compatible driver interface layer
- Multiple drivers
  - separate modules
  - accessed by drvnet name



# Layers

 = Asyn Port



# Startup

```
### XPS Set Up
# cards (total
XPSSetup(2)
#
# card, IP, PORT
(ms), idle poll
XPSConfig(0, "172.23.1
# asyn port, driver name
drvAsynMotorConfigure("XPS", "MOTORXPS", 0, 2)
#
# card, axis, groupName, positionerName
XPSConfigAxis(0,0,"Z1_Z2_Z3.Z1_BASE", 10000)
XPSConfigAxis(0,1,"Z1_Z2_Z3.Z2_BASE", 10000)
```

Create top-level Asyn port for motor primitives

Attach

Assign each axis by name to  
an Asyn address  
(peculiar to this controller)



# Startup

```
# Configure the VME carrier card
```

```
IPAC4 = ipacEXTAddCarrier(&EXTHy8002, "4 2 192")
```

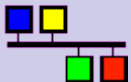
```
# Configure the stepper motor IP card
```

```
drvHy8601Configure(40, IPAC4, 0, 193, 0, 0)
```

```
# Setup the motor Asyn layer (portname, low-level driver drvnet  
name, card, number of axes on card, can_block)
```

```
drvAsynMotorConfigure("MOTOR", "drvHy8601Asyn", 40, 4, 0)
```

**EPICS**



EPICS Collaboration, APS, June 2006



# Startup

```
# Configure the VME card addresses
pmacVmeConfig (0, 0x7FA000, 0x700000, 0xc1, 2)
# Create the DRPAM ASCII buffer serial driver /dev/pmac/*/asc
pmacDrv()
# Setup the underlying Asyn port for comms
drvAsynSerialPortConfigure("p0", "/dev/pmac/0/asc", 0, 0, 0 )

# Create axes communicating via Asyn port "PMAC", addr 0
# on card 0, 6 axes
pmacAsynMotorCreate("p0", 0, 0, 6 )

# Setup the motor Asyn layer (portname, low-level driver drvname,
name, card, number of axes on card)
drvAsynMotorConfigure("PMAC", "pmacAsynMotor", 0, 6)
```



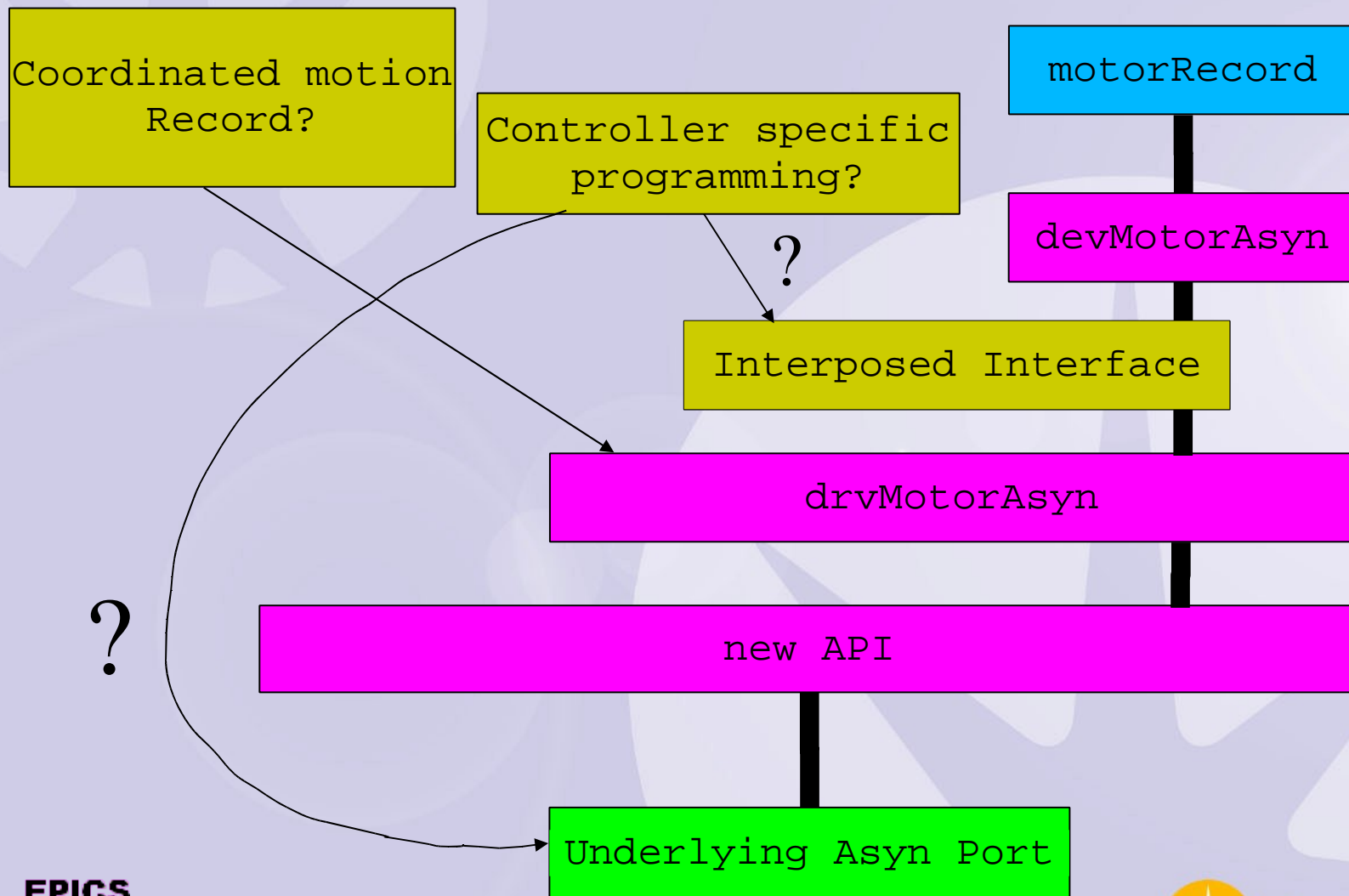


# Progress

- Common device support for motor record
- Common Asyn driver interface layer
- Simulated motor written to help testing
  - both 'old-style' and Asyn dev support
- Support for 3 controllers
  - 2 more under development
- Pre-release part of motor R5-9



# Layers (2)



# Future

- Lots more testing
- Profiled moves
- Coordinated motion?
- Extending interfaces
- Convert more controllers – do you want it?



# Thank you

- Nick Rees, Diamond – new API, Sim, PMAC
- Mark Rivers, APS / CARS – MM4000, XPS
- Ron Sluiter, APS – OMS

