# Adding and Using Motors

Kevin Peterson

2015-02-16

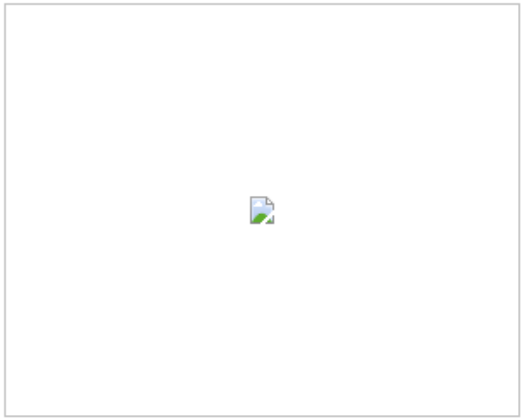
The Advanced Photon Source is an Office of Science User Facility operated for the U.S. Department of Energy Office of Science by Argonne National Laboratory

# Assumptions

- The IOC already exists
- The motor driver has already been written
- Additional motors need to be added to the IOC

# Software used during this demo

- Prebuilt IOC (Windows, OS X, Linux)
- Virtual Motor Controller (requires Python 2.7)
  - Provides 8 axes (400 steps per EGU)



- Software setup instructions:
  - http://www.xray.aps.anl.gov/~kpetersn/motorClass.html

# Adding motors to an IOC: Where to look for examples?

- Adding motors records for an existing controller
  - Refer to IOC's configuration files
- Adding support for an additional controller
  - Refer to IOC's configuration files
- Adding support for a new controller
  - Refer to example configurations in the motor module
    - motor/iocBoot/iocWithAsyn
  - Refer to example configurations in the xxx module
    - xxx/iocBoot/ioc{vxWorks,Linux}

# Adding support for a new controller: How to include the motor driver

For IOCs based on the xxx synApps module, many drivers are included by default.

vmc/vmcApp/src/Makefile:

Makefile

```
ifdef MOTOR
        vmc_Common_DBD += motorSupport.dbd devAcsMotor.dbd devAerotech.dbd
        vmc_Common_DBD += devAttocube.dbd devFaulhaberMotor.dbd devImsMotor.dbd
        vmc_Common_DBD += devKohzuMotor.dbd devMclennanMotor.dbd devMicos.dbd
        vmc_Common_DBD += devMicroMo.dbd devNewFocus.dbd devNewport.dbd
        vmc_Common_DBD += devOriel.dbd devPC6K.dbd devPIJena.dbd devPIMotor.dbd
        vmc_Common_DBD += devSPiiPlus.dbd devSmartMotorMotor.dbd devSoftMotor.dbd
        vmc_Common_DBD += devThorLabs.dbd motorRecord.dbd motorSimSupport.dbd
        vmc_Common_DBD += VirtualMotorDriver.dbd

        vmc_SRCS += VirtualMotorDriver.cpp

        vmc_Common_LIBS += Acs acsTech80 Aerotech Attocube Faulhaber Ims KohzuMotor
        vmc_Common_LIBS += Mclennan Micos MicroMo NewFocus Newport Oriel PI PIJena
        vmc_Common_LIBS += Parker SmartMotor ThorLabs softMotor motorSimSupport
        vmc_Common_LIBS += motor
endif
```

# Adding support for a new controller: Communication setup

- Ethernet communication
  - Increasingly common feature of motor controllers
  - Used with serial servers to provide many serial ports to soft IOCs

vmc.cmd

```
### Virtual motor controller

drvAsynIPPortConfigure("VMC_ETH","127.0.0.1:$(VMC_PORT1)", 0, 0, 0)

# Show communication
#!asynSetTraceMask("VMC_ETH", 0, 3)
# Only show errors
asynSetTraceMask("VMC_ETH", 0, 1)
# Leave ascii selected so traces can be turned on with a single click
asynSetTraceIOMask("VMC_ETH", 0, 1)

# Set end-of-string terminators
asynOctetSetInputEos("VMC_ETH",0,"\r\n")
asynOctetSetOutputEos("VMC_ETH",0,"\r")
```

- Serial communication
  - Used for local serial ports
  - Used for USB devices that present a serial interface to the OS

serial.cmd

```
#drvAsynSerialPortConfigure("portName","ttyName",priority,noAutoConnect,
#                           noProcessEos)
drvAsynSerialPortConfigure("serial1", "/dev/ttyS0", 0, 0, 0)
asynSetOption(serial1,0,baud,57600)
asynSetOption(serial1,0,bits,"8")
asynSetOption(serial1,0,parity,"None")
asynSetOption(serial1,0,stop,"2")
asynSetOption(serial1,0,clocal,"Y")
asynSetOption(serial1,0,crtscts,"N")
asynOctetSetInputEos("serial1",0,"\r\n")
asynOctetSetOutputEos("serial1",0,"\r\n")
# Make port available from the iocsh command line
asynOctetConnect("serial1", "serial1")
```

- For VxWorks serial examples, refer to xxx/iocBoot/iocvxWorks/serial.cmd

# Adding support for a new controller: Motor driver configuration

- The driver for the Virtual Motor Controller is a simple, model 3 driver.

vmc.cmd

```
23 # VirtualMotorController(
24 #    portName          The name of the asyn port that will be created for this driver
25 #    VirtualMotorPortName     The name of the drvAsynSerialPort that was created previously
26 #    numAxes           The number of axes that this controller supports
27 #    movingPollPeriod  The time between polls when any axis is moving
28 #    idlePollPeriod    The time between polls when no axis is moving
29
30 # 1-second idle polling
31 #!VirtualMotorCreateController("VMC1", "VMC_ETH", 3, 250, 1000)
32 # 10-second idle polling
33 VirtualMotorCreateController("VMC1", "VMC_ETH", 3, 250, 10000)
34 # No idle polling
35 #!VirtualMotorCreateController("VMC1", "VMC_ETH", 3, 250, 0)
36 # Extra axes, 10-second idle polling
37 #!VirtualMotorCreateController("VMC1", "VMC_ETH", 8, 250, 10000)
```

- The model 3 driver for the Newport XPS controller is a complex driver.
  - motor/iocBoot/iocWithAsyn/st.cmd.xps5

# Adding support for a new controller: Record configuration

- A substitutions file is used to simplify the loading of many records.

vmc.cmd

```
22 # If a substitutions file is used, the "P" macro needs to be modified by hand
23 #!dbLoadTemplate("vmc.substitutions")
```

- All of the fields in the database being loaded that do not have default values must be defined in the substitutions file.

vmc.substitutions

```
 1 file "$(TOP)/db/asyn_motor.db"
 2 {
 3 pattern
 4 {P,     N,   M,       DTYP,         PORT,  ADDR,  DESC,          EGU, DIR,  VELO,  VBAS,  ACCL,  BDST,  BVEL,  BACC,  MRES,  PREC,  DHLM,  DLLM,  INIT}
 5 {vmc:,  1,  "m$(N)",  "asynMotor",  VMC1,  0,     "motor $(N)",  mm,  Pos,  1,     .1,    .2,    0,     1,     .2,    0.0025,  4,     100,   -100,  ""}
 6 {vmc:,  2,  "m$(N)",  "asynMotor",  VMC1,  1,     "motor $(N)",  mm,  Pos,  1,     .1,    .2,    0,     1,     .2,    0.0025,  4,     100,   -100,  ""}
 7 {vmc:,  3,  "m$(N)",  "asynMotor",  VMC1,  2,     "motor $(N)",  mm,  Pos,  1,     .1,    .2,    0,     1,     .2,    0.0025,  4,     100,   -100,  ""}
 8 {vmc:,  4,  "m$(N)",  "asynMotor",  VMC1,  3,     "motor $(N)",  deg, Pos,  1,     .1,    .2,    0,     1,     .2,    0.01,    4,     100,   -100,  ""}
 9 {vmc:,  5,  "m$(N)",  "asynMotor",  VMC1,  4,     "motor $(N)",  deg, Pos,  1,     .1,    .2,    0,     1,     .2,    0.01,    4,     100,   -100,  ""}
10 {vmc:,  6,  "m$(N)",  "asynMotor",  VMC1,  5,     "motor $(N)",  deg, Pos,  1,     .1,    .2,    0,     1,     .2,    0.01,    4,     100,   -100,  ""}
11 {vmc:,  7,  "m$(N)",  "asynMotor",  VMC1,  6,     "motor $(N)",  deg, Pos,  1,     .1,    .2,    0,     1,     .2,    0.01,    4,     100,   -100,  ""}
12 {vmc:,  8,  "m$(N)",  "asynMotor",  VMC1,  7,     "motor $(N)",  deg, Pos,  1,     .1,    .2,    0,     1,     .2,    0.01,    4,     100,   -100,  ""}
13 }
```

- Values specified in the substitutions file will be overwritten at iocInit by autosave.

# Adding support for a new controller: Autosave configuration

- Important motor values are usually autosaved
  - The .req files can be found in vmc/iocBoot/iocvmc

auto_positions.req

```
1 $(P)m1.DVAL
2 $(P)m2.DVAL
3 $(P)m3.DVAL
4 #!$(P)m4.DVAL
5 #!$(P)m5.DVAL
6 #!$(P)m6.DVAL
7 #!$(P)m7.DVAL
8 #!$(P)m8.DVAL
```

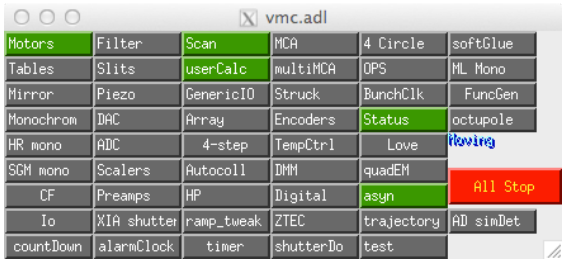
auto_settings.req

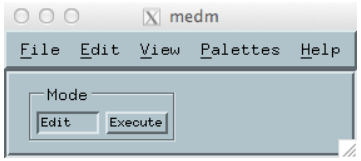
```
1 file motor_settings.req P=$(P),M=m1
2 file motor_settings.req P=$(P),M=m2
3 file motor_settings.req P=$(P),M=m3
4 #!file motor_settings.req P=$(P),M=m4
5 #!file motor_settings.req P=$(P),M=m5
6 #!file motor_settings.req P=$(P),M=m6
7 #!file motor_settings.req P=$(P),M=m7
8 #!file motor_settings.req P=$(P),M=m8
```

- If a motor controller has a non-zero position, the autosaved DVAL is NOT restored!

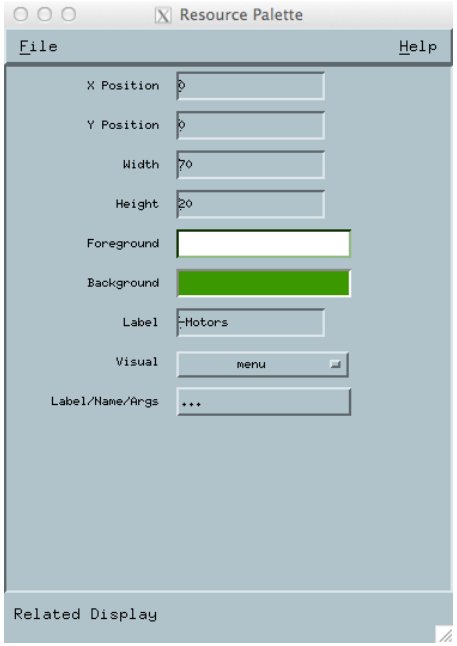
# Adding support for a new controller: GUI configuration

$ medm vmc/vmcApp/op/adl/vmc.adl &







# Soft-channel device support (Soft motors)

- The motor record documentation contains a simple example
  - http://www.aps.anl.gov/bcda/synApps/motor/R6-9/motorRecord.html#Examples
- A slightly-modified version of this example is included in the IOC for this class
  - Constant in calculation reduced from 1000 to 100 to better match motor range

st.cmd

```
22 # Virtual Motor Controller
23 < vmc.cmd
24
25 # Soft Channel Example
26 #!dbLoadRecords("$(TOP)/db/softChannelExample.db", "P=$(PREFIX),M=m1,SM=sm1")
27
28 ### Allstop, alldone
29 dbLoadRecords("$(TOP)/db/motorUtil.db", "P=$(PREFIX)")
```

# Questions?

???