Motion control and EPICS

Torsten Bögershausen

Motion Control and Automation Group

Plans for today

EUROPEAN SPALLATION SOURCE

- Overview
- EPICS abstraction
- Future development
- Demo of demo system
- Hands on

EPICS Training 2016 – short version

Using Motors

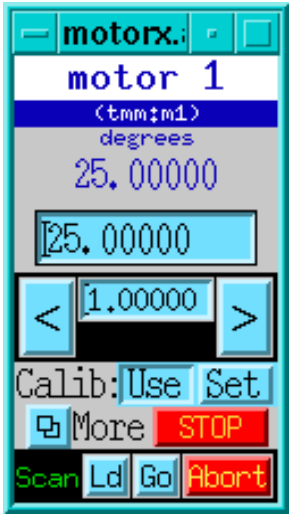
Stolen from Ronald L. Sluiter

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

What's the Motor module for?

- Device independence – motor hardware is transparent to users.
  - Same user displays and motor motion behavior, for all devices.



Available operations from this display;

Make absolute or incremental moves.

Define the current position.

Stop the current move.

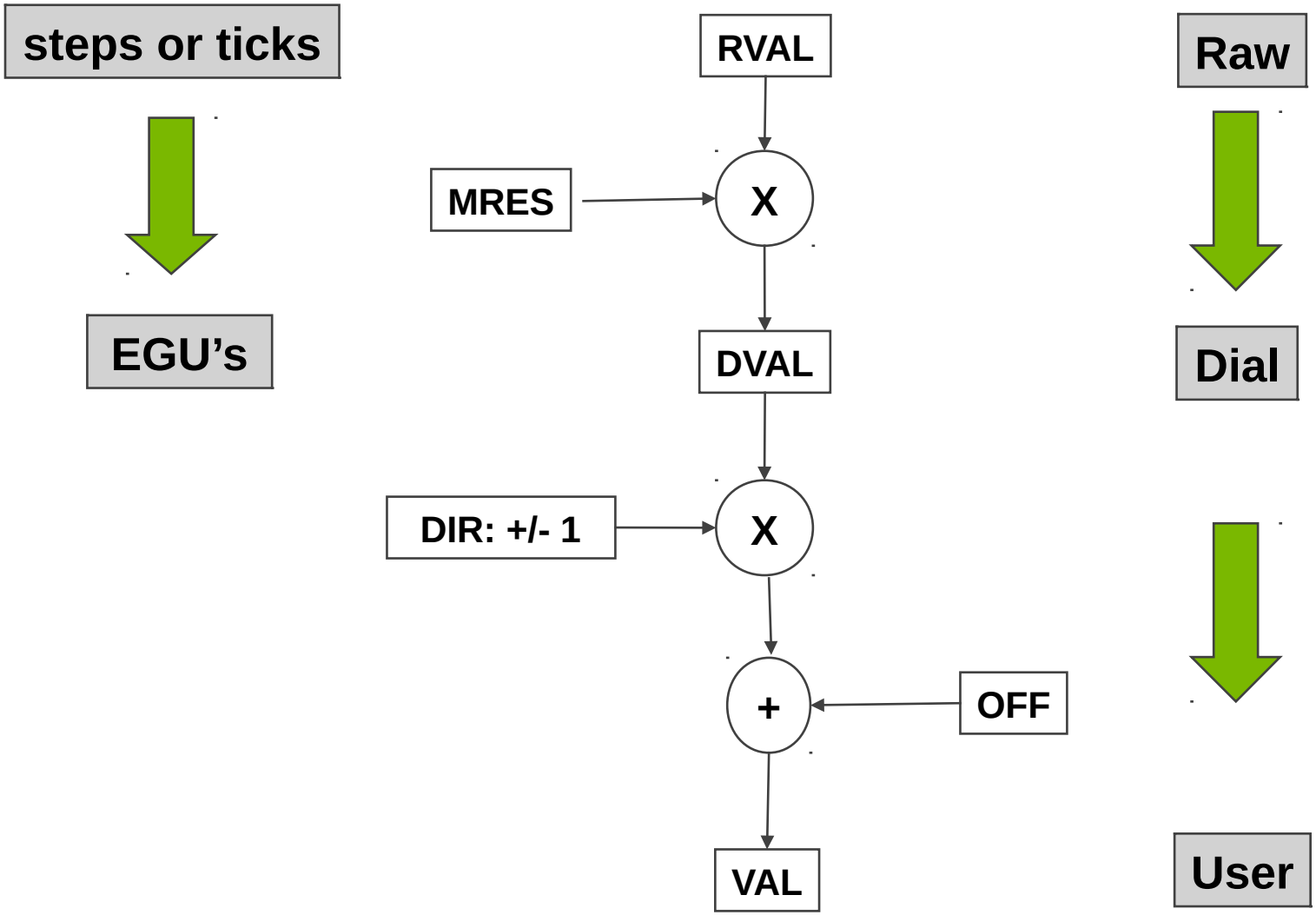
without any controller specific information.

Features - scope

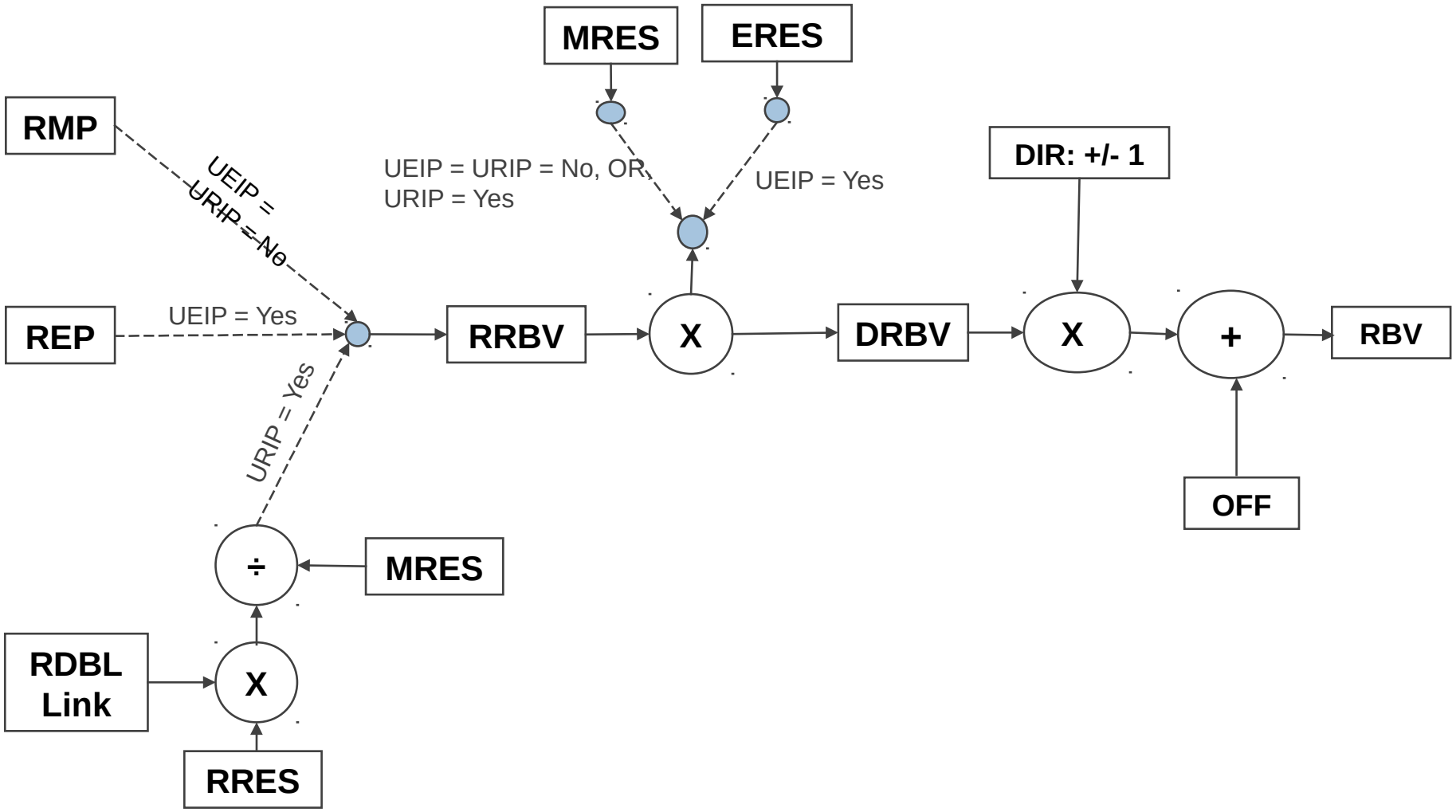
- The scope of the MR is limited to single axis, non-coordinated, point to point moves.
- Including homing, status & error reporting
- Including the configuration of the axis (MRES, RDBD, HLM, LLM)
- Config good enough for “simple” motion controllers.
- Advanced motion controllers have many, many more parameters
- My plan is to change the configuration for ESS:
  read values from controller and put them into the MR

- The scope for the model 3 driver is larger:
- For multi-axis coordinated motion and trajectory scans see…
  - https://subversion.xray.aps.anl.gov/synApps/motor/trunk/documentation/trajectoryScan.html

Features - coordinate systems.



Feedback data flow



Features - Move types

- Absolute (VAL, DVAL, RVAL)
- Relative (RLV)
- Incremental (TWF, TWR, TWV)

MotorRecord @ ESS

EUROPEAN SPALLATION SOURCE

- Abstracts the motor into an EPICS PV
  `LabS-ESSIIP:MC-Pos-01`
  `LabS-MCAG:MC-Pos-TS01`
- Converts writing to a field into commands to the controller
- Retrieves the status from the controller
- EPICS PV can be reached via:
- C/C++/Java/Python/sh

Fields used in motorRecord

- VAL: Where the motor should be
- RBV: Where the motor is
- MOVN: Moving
- DMOV: Done moving
- STOP: stop moving
- LVIO: (soft) limit violation
- STAT: Alarm status
- MISS: ”Missed” (see RDBD, ”retry deadband”)

Limitations in motorRecord

- Stepper motor abstraction
  - No good support for microstepping
  - No good support for controllers using EQU, e.g. Millimeter
- Features, which are considered a ”bug” today: motor stopped when going into the wrong direction
- But: a usable abstraction of a motor

MotorRecord today

- Maintained at APS.
- Issue(s):
  Change one line of code -> something breaks on another beamline
- Solution:
  a) Take out a new branch to work on
  b) Clean up, refactor & develop
  c) Keep most EPICS fields, remove some
  d) Result is no more 100% ”the motorRecord”
  e) As agreed with APS, call it ”axisRecord”

AxisRecord

EUROPEAN SPALLATION SOURCE

- Ongoing development under 2017++
- Deployed at ESS early 2017
- caget(motor.RBV) works as before.

Status as used at SNS

- 3 States in control screen:
  IDLE Moving ERROR

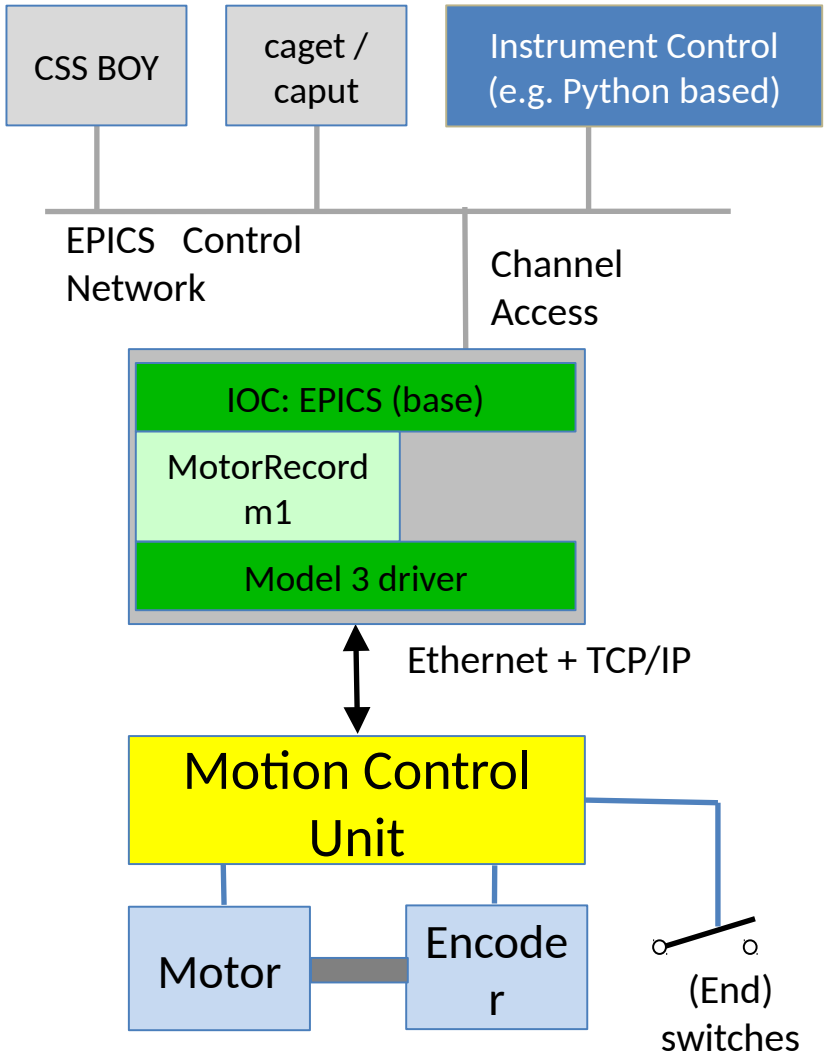
```
if (!DMOV || MOVN)
  status = Moving
else if (STAT != 0 || MISS)
  status = ERROR
else
  IDLE
```

Status as used in Nicos ?

- 4 States in control screen:
  OK BUSY WARN ERROR

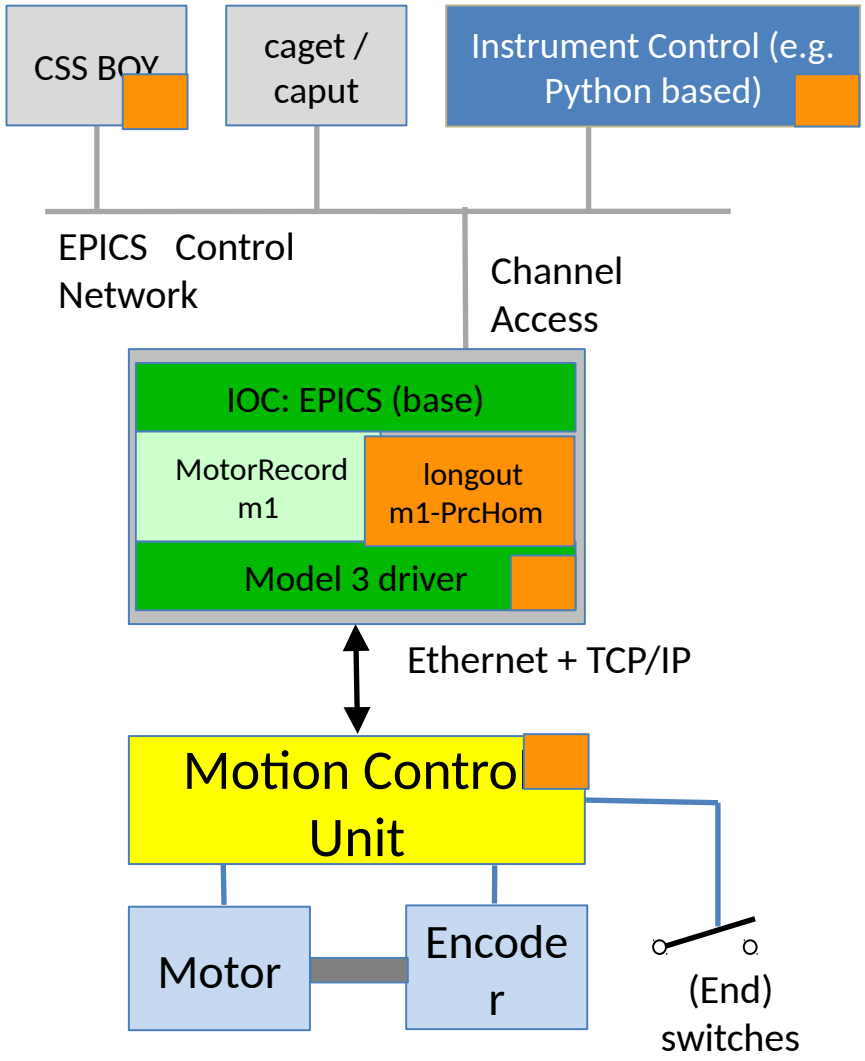
```
if (!DMOV || MOVN)
  status = BUSY
else if (LVIO)
  status = WARN
 else if (STAT != 0 || MISS)
  status = ERROR
else
  status = OK
```

Motion Control Integration (today)



- 3 layers in control box:
  - EPICS IOC
  - Motor record (single axis)
  - Model 3 driver
- Basic functionalities for point-to-point movements
- Future extensions of functionality in additional records

Motion Control Integration (tomorrow)



- 3 layers in IOC:
  - Motor record (single axis)
  - longout record (homing)
  - Model 3 driver
- Basic functionalities for point-to-point movements
- Future extensions of functionality in additional records
- Current development includes a.o. jerk, homing procedures
- Extensions need to be implemented in user interfaces, model 3 driver and motion control unit as well (strong coordination necessary)

Move & monitor a motor (done on a simulator, Mac or raspi)

```
caput IOC:m1.VAL 60.1

Old : IOC:m1.VAL
              24.9

New : IOC:m1.VAL
              60.1
```

#Notes:

```
#caput:   Channelaccessput
#camonitor:   Channelaccessmonitor
#IOC:m1..DMOV: Field"donemoving"
#IOC:m1.RBV:  FieldReadbackvalue
```

- camonitor IOC:m1.RBV IOC:m1.DMOV
- IOC:m1.RBV 2016-02-02 10:58:16.157350 24.87
- IOC:m1.DMOV 2016-02-02 10:58:16.658745 1
- IOC:m1.DMOV 2016-02-02 10:58:16.658745 0
- IOC:m1.RBV 2016-02-02 10:58:30.124368 49.86
- IOC:m1.RBV 2016-02-02 10:58:31.138087 60.06
- IOC:m1.DMOV 2016-02-02 10:58:31.639072 1

Automated tests

EUROPEAN SPALLATION SOURCE

- Find many bugs
- Python based
- With real hardware
- Today: Part of TwinCAT,
  Tomorrow: Part of axisRecord.

python

EUROPEAN SPALLATION SOURCE

- `motor = 'LabS-ESSIIP:MC-Pos-01'`
- `epics.caput(motor, destination, wait=True)`
- `stat = epics.caget(motor + '.STAT', use_monitor=False)`

Future plans

EUROPEAN SPALLATION SOURCE

- Tomorrow: EPICS driver part of axisRecord
- Today: Configuration MR → controller
- Tomorrow: Configuration controller → MR
- Slits as virtual axes
- Scans as waveforms

Motion at in kind partners

- ISIS ?
- JCNS ?
- PSI ?
- Other ?
-

Requirements

- Positioner – single axis
- Coordinated slit
- (wire) scan & position capture
-
- What more ?

Tools

- Git – version control
- Jira – issue tracker
- Naming convention !
- CCDB – IOC factory
- How do you store low level motor configs ? Mark ? Fabian ? ISIS ? Others ?
  /home/motion/BEAMLINE1/motor1
  `LabS-ESSIIP:MC-Pos-01`

SW platforms

EUROPEAN SPALLATION SOURCE

- TwinCAT
- Linux open source
- Simulator in C (volonteers for python ?)
- Test code in python
  Function test for the Record
  Commisioning real hardware

EPICS interface

- motor Record ->axis Record
- Profile move
- Position readback (arrays == waveforms)
- RPC with ”EPICS7” ?

That's it – in theory

-

Questions ?

Hands on

- Start your VM
-

That's it

EUROPEAN SPALLATION SOURCE

- Thanks for listening