Next generation motor record ?

Torsten Bögershausen

Motion Control and Automation Group

MotorRecord Status

- (2015 – late 2016):

  Re-write from scratch ?

- - Fix bugs
  - Implement features we need
  Follow up in 2018 @ APS

Axis Record - birth

- Fork of motor in late 2016

  - Call it axis rather than motor
  - Read config from controller
  - Bug fixes (state machine)
  - Read back with encoder > 32 bits
  - Improved sync when IOC is (re)started

Parameters in motorRecord

- Can configure a stepper quite well:
  SREV/UREV or MRES
  VELO, VMAX, ACCL, RDBD, DLY, SDBD
  DLLM, DLLM
  JVEL, JAR, HVEL

  HVEL

Parameters in controller

- Scaling, dynamics, soft limits, monitoring
  SREV/UREV
  VELO, VMAX, ACCL, JERK, RDBD/SDBD, DLY, DLLM, DLLM, JVEL
- Homing parameters
- Because: encoders allow better dynamics with closed loop:
- Lots of (optional) dynamic parameters

Configuration: Where is the truth ?

- Motion control engineers
- Configure the machine (safe, no burning motors):

#Driver reads controller:
out=HLenabled?;Hlvalue?
in=1;169 CHLM_En=1 CHLM=169

#Record gets value from driver:
motorRecord.cc:3943 IOC:m1 pmr->dhlm=169 maxValue=169

Improvements

EUROPEAN SPALLATION SOURCE

- RMP: 32 bit not enough, float64 better
  motorRecord.html says ”double”
  motor/issues/8
- Autopower 2:
  - ”Blind timeout” may be shortened.
  - If wanted: leave power on forever
- Re-calc of LVIO when soft limits changed (ongoing)

Improvements

EUROPEAN SPALLATION SOURCE

- Flags can configure the Record (Controllers need different behaviour) MFLG field:
- - HomeOnLS
- - NoStopProblem
- - LSrampDown

Software refactor

- State machine, dev support, model 3 driver: move(), moveVelocity(), home(), stop()
- Record today:
  doStartMoving()
  doStartJogging()
  doStartHoming()
  devSupStop()

Upgrade, Migration

EUROPEAN SPALLATION SOURCE

- We (at ESS) converted into axis good - but:
  Lost connection to community.
  Lost soft motor
- Could use axis and motor on the same IOC:
  Never needed, never used.

Axis Record – the end

- December 2017:
  -Submodule axis, axisCore, drivers/
  - All improvements
  - Remove warnings here and there.
- Never really used
- December 2017, Instrument control WS:
  Soft motor & community had been lost.
- Re-integrated important stuff into motor

Added fields

- MISV: MISS alarm serVerity
- SDBD: Set point DeadBand
- MFLG: Flags from driver
- PRIV: Private data

  - info from driver
- Removed fields
  - LVAL, LDVL, LRVL, ALST, MLST)

MsgTxt

- Controllers have useful information:
  `Error "4650"`
- State machine has useful information:
  `E: Axis not homed`
- Drivers have useful information:
  `I: Moving ABS`

MsgTxt

EUROPEAN SPALLATION SOURCE

- People like it
- Common messages are in asynMotorAxis
- Driver-specific in driver

Improvements axis, not in motor

EUROPEAN SPALLATION SOURCE

- Done in axis:
  Generate field documentation from dbd (axisRecord.dbd.pod)
- Various compiler warnings (code not tested)

Problems, headaches

EUROPEAN SPALLATION SOURCE

- Loose list of
  - Problems
  - Headaches
  - Observations

Problems, headaches

- ”Problem” reported from controller (”MCU”)
- Limit switches
- Homing against limit switches
- NTM field
- MRES
- Autosave & Restore ?
- IOC started before controller

MCU reports a problem

- Motor is not homed → Problem
- HW failure → Problem
- Out of range value (acceleration) → Problem
- Motor out of control (?) → Problem

Stop on problem

Stop on problem

- Record sends a stop
- But the problem does not go away.

Problems & Alarm state

- How to put Record into alarm state ?
  PROBLEM bit !
  - MotorRecord < 6.10: PROBLEM means alarm
  - MotorRecord > 6.10: PROBLEM means stop.
  (Which driver(s) needs stop?)
  (How do drivers signal alarm / problem)

NoStopProblem

EUROPEAN SPALLATION SOURCE

- MotorRecord @ ESS: setIntegerParam(pC_->motorFlagsNoStopProblem_, 1);

Limit switches

- Limit switches

Limit switches, general

- Running into a limit switch: should stop
- Must allow to move away from a limit switch
- Should not stop when limit switch is used as ”home switch”
- Current ”stop” from Record to Driver confuses the controller.

Limit switches, move away

- Today: Commanded direction, CDIR
  - Does not work with 3rd party movements
- HOMF, HOMR vs LLS HLS
  - Must use appropriate
  HOMF when LLS or HOMR when HLS
- Record does nothing - No feedback at all.
  not user friendly, not script friendly
- Calc/transform record to improve?

Homing against a limit switch

- 1st Problem:
  We have ”homing sequences” in the MCU
  Sitting on LLS and homing against LLS works:
  Motor is moved away, and then homed.
- But: motorRecord blocks HOMR @ LLS,
  no feedback to the user.

Homing against LS, workaround 1

- Workaround 1:
  Don’t report ”LLS active” to record when MCU reports that it is homing.

- Good: Homing sequence is not aborted
- Bad: Can’t see the LLS in user interface

Homing against LS, workaround 2

- Workaround 2, current state
  setIntegerParam(
  pC_->motorFlagsHomeOnLs_, 1);
  Ignore LLS in the record for HOMR
- Good: Can home from record.
  LLS is displayed in CSS
- Bad: Interrupts homing when IOC is running and homed by engineering tool (or other IOC)

Homing against LS, HW damage

- Homing switches had been mixed up
  Trying to home:
  - No stop from Record
  - No stop from Driver
- worst possible accident, (GAU in german)
- Suggestions ?

Limit switches in Record

- What should happen: ”Wrong LS” is activated
  - Stop ? (the safe way)
  - Ignore ?

- And: How do you move away from LS?
  Best practice anyone ?

NTM field

- Stops motor when traveling into the wrong direction.
- Surprises users of servo motors
  stops on overshoot

What to do with MRES

- Needed for stepper motor with controller using steps:
  Controller counts in steps.
  Record can EGU or steps (VAL, RVAL)
- What happens if the controller uses EGU?

  Controller is configured in ”mm”, set scaling:
  - Steps/revolution
  - mm/revolution
  - sign bit for direction

MRES, 1st approach

- 1st approach:
  Set MRES = 1.0
- Setpoint:
  Record: VAL = 12.34 mm → devSupport 12.34
  → model 3 driver 12.34 →
  [over the cable] → controller 12.34
- Readback:
  controller 12.34 → [cable] driver 12.34 →
  devSupport 12.00 → record 12.00 RBV

MRES, 2nd approach

- 2nd approach
  Dummy MRES in record e.g. 1um.
  Scale factor in driver 1/1000.
- Works better.
- But: People looking at REP/RMP: no relation to steps.
- Why can’t I position better than 30um ?

MRES, 3rd approach

- 3rd approach – not yet finished
  Use EGU everywhere – Record, devSup, driver, wire, controller.
- Needs a flag to be enabled.
- Read MRES/ERES from controller
- MRES/ERES used to calculate RMP/REP.

MRES, more comments

- More comments on MRES ?
- How do YOU do ?

Save & Restore

EUROPEAN SPALLATION SOURCE

- Seems to be used and useful
  APS, BNL
- Seems to be problematic
  https://github.com/epics-modules/motor/issues/85
- Experiences, solutions ?

IOC started before controller

- Problem: An IOC is started before the controller is started.
- Nice to have in testing/commissioning phase
- Solution:
  a) Don’t do that, be happy
  b) Record stays in UDF - postpone init() (work ongoing)

That's it

- Thanks for listening