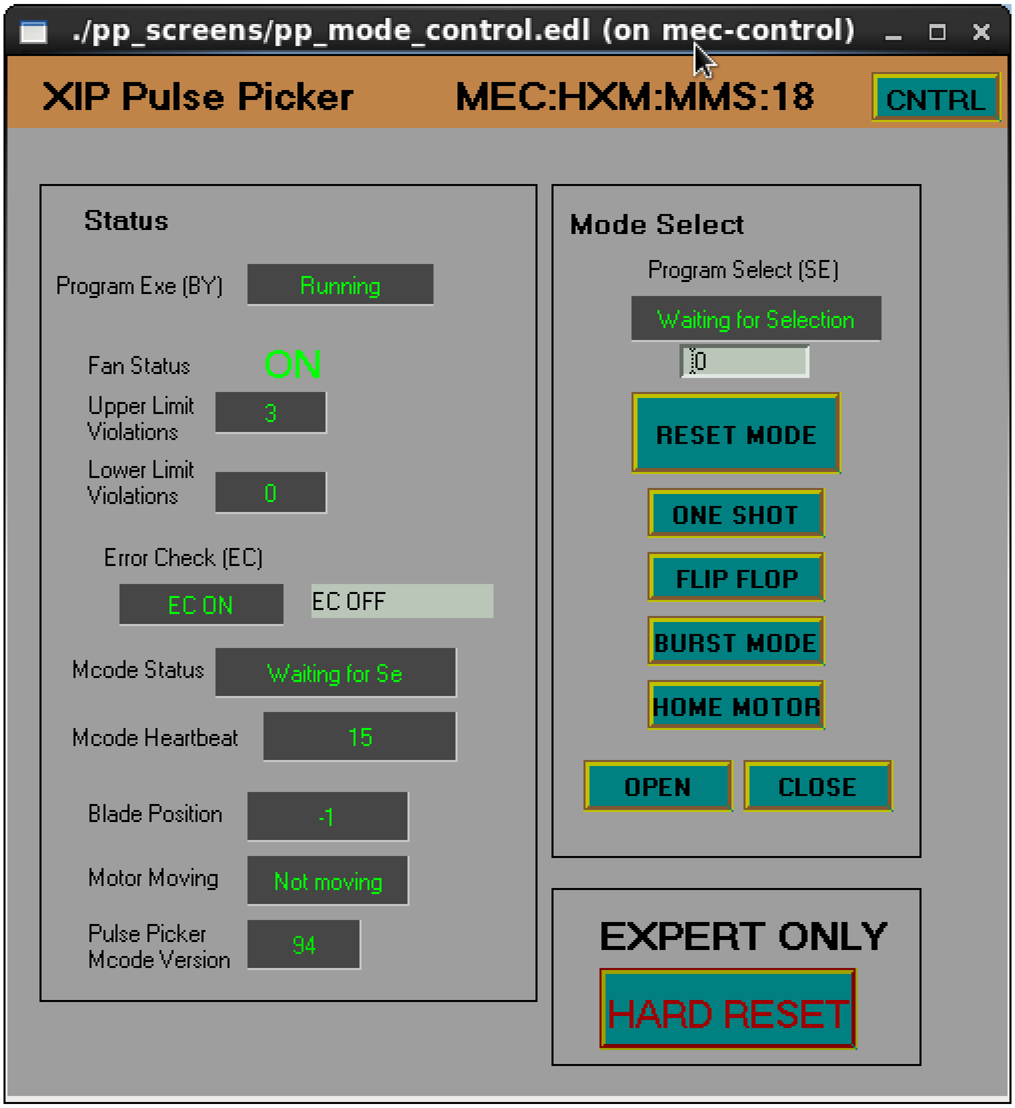
XIP Pulse Picker EPICS Ioc and Edm Screens

Pulse Picker Edm Screens



./pp_screens/pp_mode_control.edl (on mec-control)

XIP Pulse Picker MEC:HXM:MMS:18

Status

Program Exe (BY) Running

Fan Status ON

Upper Limit Violations 3

Lower Limit Violations 0

Error Check (EC) EC ON EC OFF

Mcode Status Waiting for Se

Mcode Heartbeat 15

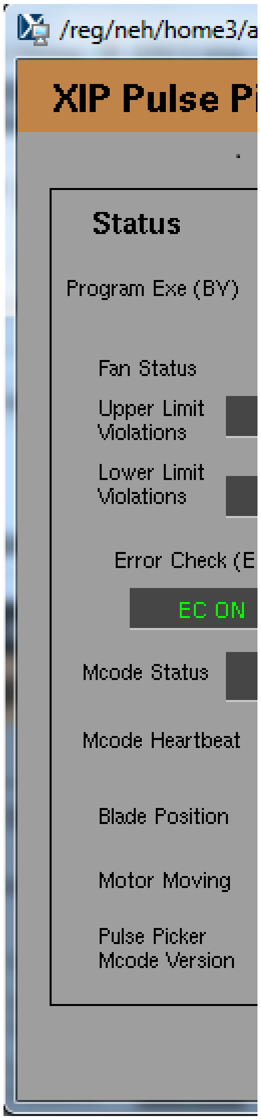
Blade Position -1

Motor Moving Not moving

Pulse Picker Mcode Version 94

Mode Select

Program Select (SE) Waiting for Selection 0



/reg/neh/home3/a

XIP Pulse Pi

Status

Program Exe (BY)

Fan Status

Upper Limit Violations

Lower Limit Violations

Error Check (E

EC ON

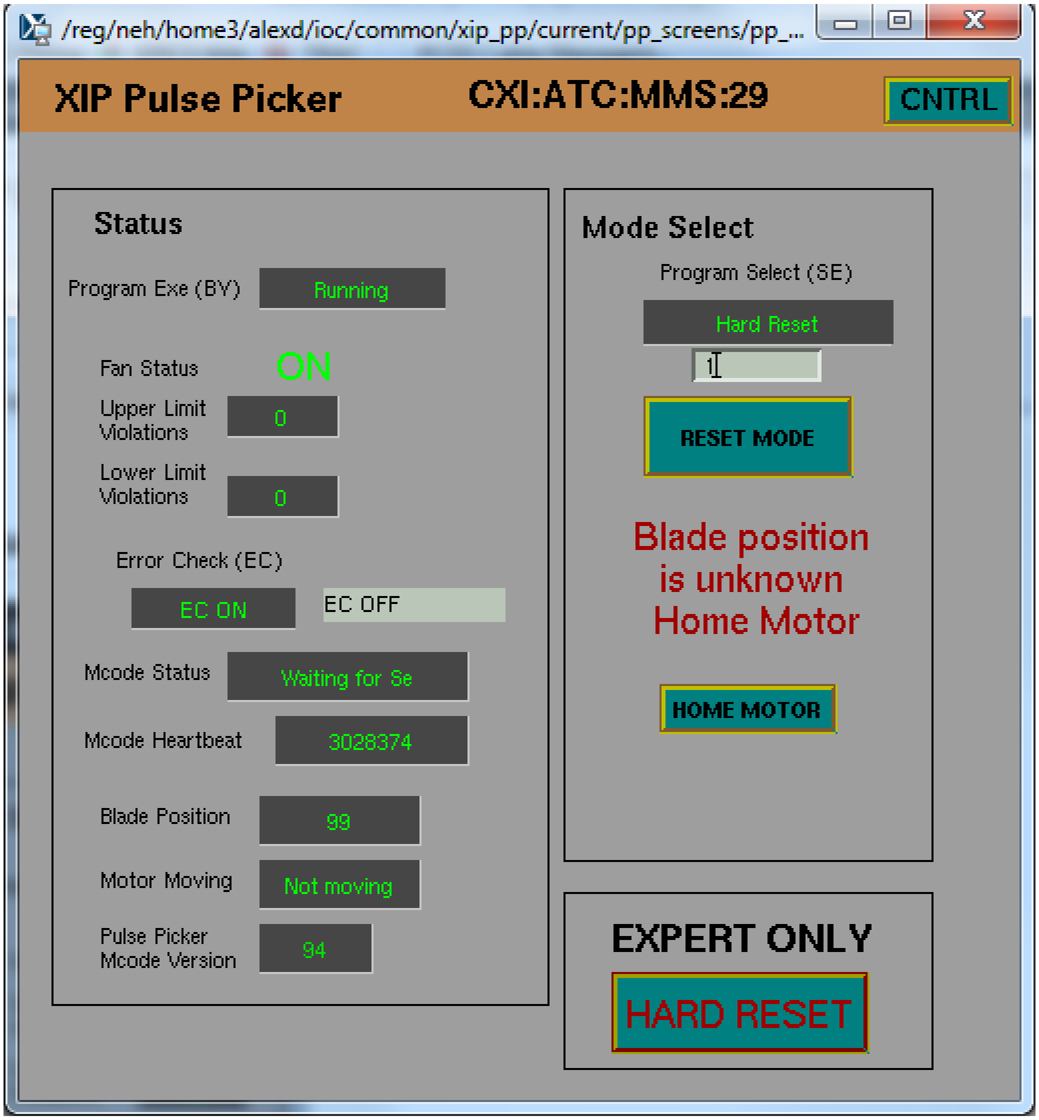
Mcode Status

Mcode Heartbeat

Blade Position

Motor Moving

Pulse Picker Mcode Version



/reg/neh/home3/alexd/ioc/common/xip_pp/current/pp_screens/pp_...

XIP Pulse Picker CXI:ATC:MMS:29

Status

Program Exe (BY): Running

Fan Status: ON

Upper Limit Violations: 0

Lower Limit Violations: 0

Error Check (EC)

Mcode Status: Waiting for Se

Mcode Heartbeat: 3028374

Blade Position: 99

Motor Moving: Not moving

Pulse Picker Mcode Version: 94

Mode Select

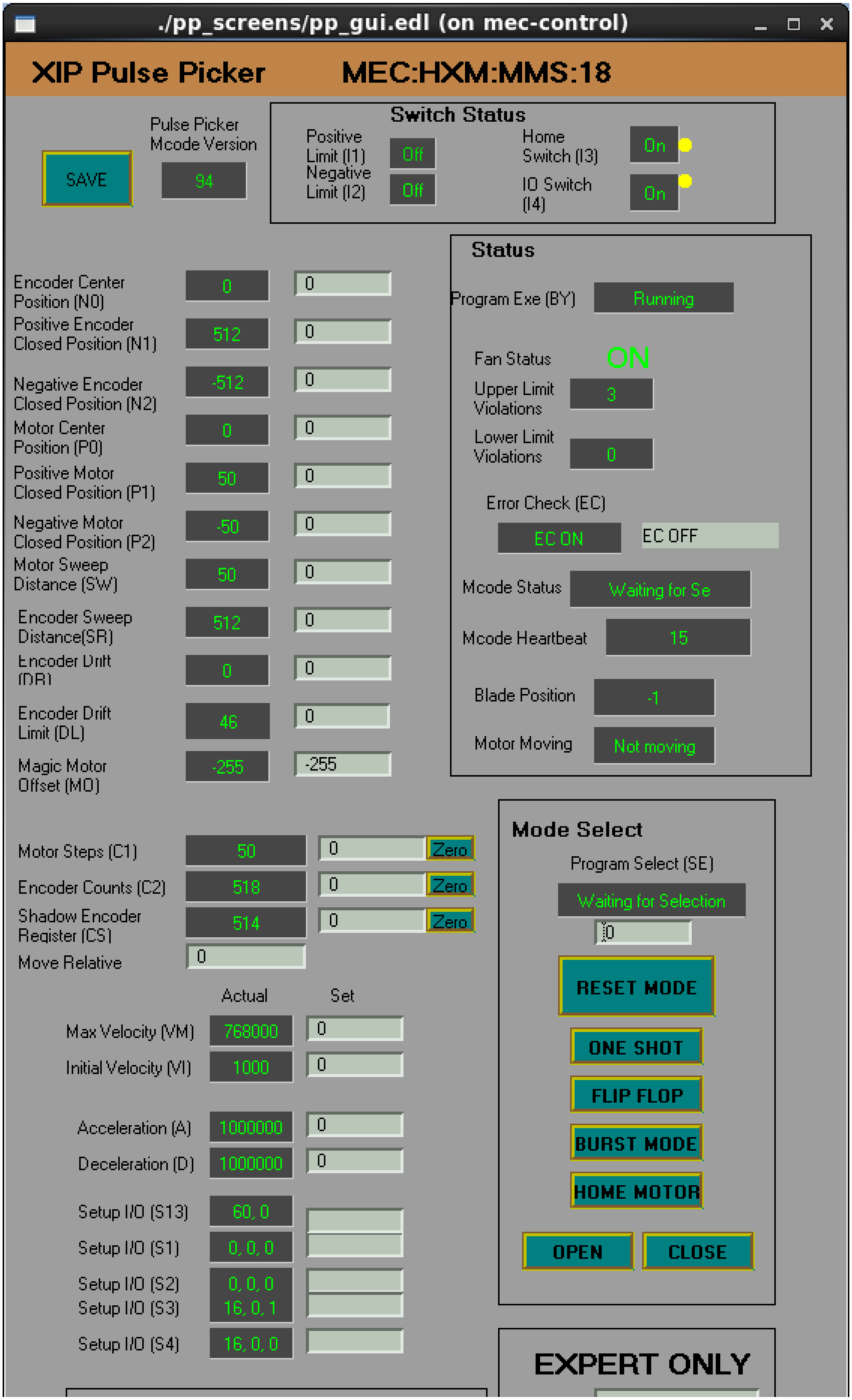
Program Select (SE): Hard Reset

1

Blade position is unknown Home Motor

EXPERT ONLY

Fig 1. pp_mode_control.edl waiting for mode selection Fig 2. pp_mode_control.edl, Pulse picking mode has been selected Fig 3. pp_mode_control.edl, Blade position unknown motor, homing required



./pp_screens/pp_gui.edl (on mec-control)

XIP Pulse Picker MEC:HXM:MMS:18

Pulse Picker Mcode Version: 94

Switch Status

Positive Limit (I1)	Off	Home Switch (I3)	On ●
Negative Limit (I2)	Off	IO Switch (I4)	On ●

Encoder Center Position (N0)	0	0
Positive Encoder Closed Position (N1)	512	0
Negative Encoder Closed Position (N2)	-512	0
Motor Center Position (P0)	0	0
Positive Motor Closed Position (P1)	50	0
Negative Motor Closed Position (P2)	-50	0
Motor Sweep Distance (SW)	50	0
Encoder Sweep Distance(SR)	512	0
Encoder Drift (DR)	0	0
Encoder Drift Limit (DL)	46	0
Magic Motor Offset (MO)	-255	-255

Motor Steps (C1)	50	0
Encoder Counts (C2)	518	0
Shadow Encoder Register (CS)	514	0
Move Relative	0	

	Actual	Set
Max Velocity (VM)	768000	0
Initial Velocity (VI)	1000	0
Acceleration (A)	1000000	0
Deceleration (D)	1000000	0
Setup I/O (S13)	60, 0	
Setup I/O (S1)	0, 0, 0	
Setup I/O (S2)	0, 0, 0	
Setup I/O (S3)	16, 0, 1	
Setup I/O (S4)	16, 0, 0	

Status

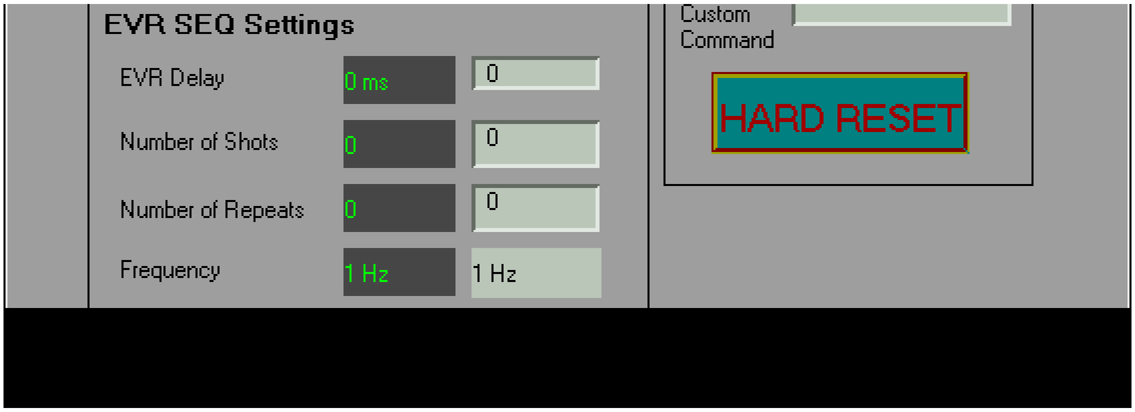
Program Exe (BY)	Running
Fan Status	ON
Upper Limit Violations	3
Lower Limit Violations	0
Error Check (EC)	EC ON
Mcode Status	Waiting for Se
Mcode Heartbeat	15
Blade Position	-1
Motor Moving	Not moving

Mode Select

Program Select (SE): Waiting for Selection

0

EXPERT ONLY



EVR SEQ Settings

EVR Delay 0 ms 0

Number of Shots 0 0

Number of Repeats 0 0

Frequency 1 Hz 1 Hz

Custom Command

Fig 4. pp_gui.edl

- **pp_mode_control.edl**
  - Shown in Fig 1. through Fig 3.
  - User screen for pulse picker motor
  - Displays current motor status including: blade position, pulse picker mcode status and version, fan status, limit violations, error mode, mcode heartbeat, and selected pulse picking mode.
  - Contains buttons to select pulse picking mode and perform a hard reset of the pulse picker mcode.
  - Contains button to open to open pp_gui.edl, labeled CNTRL.
  - The edm macro for the screen is MOTOR=<Motor Base PV>, ie edm -x -eolc -m "MOTOR=CXI:ATC:MMS:29" pp_screens /pp_mode_control.edl &
- **pp_gui.edl**
  - Shown in Fig 4.
  - Expert screen for pulse picker motor
  - Contains all displays and controls contained the in pp_mode_control.edl screen
  - Contains displays and controls for all variables for pulse picker Mcode.
- **Pulse picker screen logic**
  - Both screens have built in logic to assist the user in the correct operation of the pulse picker, the logic is the same for both screens.
  - If the user has selected a mode for the pulse picking motor all of the mode selecting buttons will become invisible until the motor has been returned to 'Waiting for Mode Select'. This logic is show in Fig 2.
  - If the position of the pluse picker blades is unknown, DF=99, only the home motor button will be shown to the user and the user will be shown a message stating that the motor needs to be homed. Once the blade position returns to a know position the other buttons will become visible. The homing process takes a few seconds. This logic is show in Fig 3.
- **Pulse Picker Screen Launch Script**
  - The script edm-pp-screens.cmd can be used to launch the pp_mode_control.edl screen with the correct motor name for each hutch
  - To launch screen for a specific hutch pass the three letter acronym of the hutch to the script, the script will accept upper, lower and mixed case arguments.
  - Currently the script is located in: /reg/neh/home/alexd/ioc/common/xip_pp/current, the script will be moved to a release at the next update to the pulse picker ioc.

edm-pp-screens.cmd

```
$ ./edm-pp-screens.cmd mec
```

Additional Screens

There are additional screens located in the /pp_screens directory, these screens are only meant to be used by MEC, they have PVs hard coded in to them.

One-Shot

One-shot mode select button has been removed from pp_mode_select.edl and pp_gui.edl as of R1.4.1. The one-shot option has not been removed from the mcode or IOC. The mode is redundant as Burst Mode uses the same functions but does not require resetting after each use.

Epics IOC

Motor Interlocks

- If pules picking motor is in pulse picking mode, $(MOTOR):SE_L != 0, X and Y stepper motors are disabled
- If Y motor position is greater than 10mm X motor is disabled
- If X motor position is greater then 5mm or less then -5mm Y motor is disabled.
- Interlock records can be found in the 2-axis-control.sub-db file.

Records located in xip.template

$(MOTOR):SE

- Pulse picking mode select
- Reads Se from mcode
- Returns string corresponding to selected pulse picking mode

$(MOTOR):SE_L

- Raw pulse picking mode select number
- Reads Se from Mcode
- Returns number corresponding to selected pulse picking mode
- See Pulse Picker MCode Pseudocode & Reference for number description

$(MOTOR):SET_SE

- Pulse picking mode select
- Writes Se to Mcode, values 0 to 100 are allowed

$(MOTOR):MCODE:INIT

- Sends a hard reset to the Mcode
- Sets Se to 99
- Executes program 1
- Most variable used in the pulse picking mcode will be reset to standard values.
- Motor will need to be re-homed if this is called

$(MOTOR):RESET_PG

- Resets pulse picking mode
- Sets Se to 0
- Mcode will see Se=0, close the blades then return to mode select

> **Reset Program**
>
> The reset process is completed when the SD is "Waiting for Se", SD_L=0

$(MOTOR):SD_L

- Raw Mcode Program status
- Returns number corresponding to mcode program status
- scanned 1 second
- See Pulse Picker MCode Pseudocode & Reference for number description
- Forward link to $(MOTOR):SD

$(MOTOR):SD

- Mcode program status
- Returns string corresponding to status of mcode program

$(MOTOR):RUN_ONESHOT

- Selects One Shot pulse picking mode
- Sets SET_SE to 1
- Turns output 4 on
- Read Se back from Mcode

$(MOTOR):RUN_FLIPFLOP

- Selects Flip Flop pulse picking mode Mode
- Sets SET_SE to 2
- Turns output 4 on
- Reads Se back from mcode

$(MOTOR):RUN_BURSTMODE

- Selects Burst Mode
- Sets SET_SE to 3
- Turns output 4 on
- Reads Se back from mcode

$(MOTOR):RUN_FLIPFLOP

- Selects Flip Flop Mode
- Sets SET_SE to 2
- Turns output 4 on
- Reads Se back from mcode

$(MOTOR):S_OPEN

- Moves shutter to open position
- Sets Se to 4
- Reads Se from Mcode

$(MOTOR):S_CLOSE

- Moves shutter to close position
- Sets Se to 5
- Reads Se from Mcode

$(MOTOR):HOME:MOTOR

- Selects automatic motor homing and offset
- Sets Se to 7
- Reads Se from Mcode

> ⓘ **Mode Selecting**
>
> To change pulse picking modes, ie processing RUN_<MODE> PV, RESET_PG must be processed before selecting the new mode

$(MOTOR):LD

- Lower limit drift violation counter
- Scanned at 1 second
- Minor alarm at 100

$(MOTOR):UD

- Upper limit violation
- Scanned at 1 second
- Minor alarm at 100

Records that use xip_readwrite.template called by pulse-motor-ims.sub-db

$(MOTOR):N0

- Center position of motor in Encoder Counts
- Reads N0 from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_N0

- Center position of motor in Encoder Counts
- Writes N0 to mcode

$(MOTOR):N1

- Positive motor closed position in Encoder Counts
- Reads N1 from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_N1

- Positive motor closed position in Encoder Counts
- Writes N1 to mcode

$(MOTOR):N2

- Negative closed position of motor in Encoder Counts
- Reads N2 from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_N2

- Negative closed position of motor in Encoder Counts
- Writes N2 to mcode

$(MOTOR): P0

- Center motor position in motor microsteps
- Reads P0 from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_P0

- Center motor position in motor microsteps
- Writes P0 to mcode

$(MOTOR): P1

- Positive closed motor position in motor microsteps
- Reads P1 from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_P1

- Positive closed motor position in motor microsteps
- Writes P1 to mcode

$(MOTOR): P2

- Negative closed motor position in motor microsteps
- Reads P2 from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_P2

- Negative closed motor position in motor microsteps
- Writes P2 to mcode

$(MOTOR): SW

- Sweep distance from closed position to open position in microsteps
- Reads SW from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_SW

- Sweep distance from closed position to open position in microsteps
- Writes SW to mcode

$(MOTOR):SR

- Sweep distance from closed position to open position in encoder counts
- Reads SR from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_SR

- Sweep distance from closed position to open position in microsteps
- Writes SR to mcode

$(MOTOR):DR

- Position drift at closed positions in encoder counts
- Reads DR from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_DR

- Position drift at closed positions in encoder counts
- Writes DR to mcode

$(MOTOR):DL

- Position drift limit in encoder counts
- Reads DL from mcode
- Passive Scanning
- Processed at IOC start

$(MOTOR):SET_DL

- Position drift limit in encoder counts
- Writes DL to mcode

Records that use xip_read.template called by pulse-motor-ims.sub-db

$(MOTOR):DF

- Current motor position
- Reads DR from mcode
- Scanned at 1 second

$(MOTOR):HB

- Mcode Heartbeat
- Reads HB from mcode
- Passive scanning
- Scanned at IOC start

$(MOTOR):NS

- Pulse picker mcode version
- Reads NS from mcode
- Passive Scanning
- Scanned at IOC start

Records contained in xip_evr_seq.template

ⓘ **EVR SEQ Records**

All records in template do not have inputs or outputs. They are read by python module to select proper pulse selecting mode

$(MOTOR):PP_EVRDELAY

- EVR Delay for pulse picker
- Engineering units 'ms'

$(MOTOR):PP_NSHOTS

- Number of shots for picker to select

$(MOTOR):PP_REPEAT

- Number of repetitions of pulse selecting

$(MOTOR):PP_FREQ

- Frequency of selected pulses

$(MOTOR):PP_SEQ

- Use Sequencer <Yes,No>

$(MOTOR):PP_EVRDELAY

- EVR Delay for pulse picker

##

- Tasks to complete for PP ioc
  - Add records to write to desired sequencer and evr records and hold desired information
  - Edit EDM screen to reflect changes to ioc
  - look in to turning on scanning so motion is displayed on screens, may be covered by MV scanning
  - Update confluence page