XCD Sample Scripts

XCD Sample Script for a Rotary Stage to Home on Index with NanoCommander

```
////////////////////////////////////////////////////////////////////////////////
// XCD sample homing script for rotary stage with no hard stops to home at index.
// Please make sure that the stage is tuned and configured properly per the tuning
// instructions shown in the XCD Firmware manual before running this script.
// The script assumes that the ENR units are in Degrees/Count and all related
// units are in degrees (i.e. VEL in deg/s).
////////////////////////////////////////////////////////////////////////////////

delay 2000

// Set slow VEL and ACC for homing
VEL = 60          // deg/sec
ACC = 600         // deg/sec^2
MTL = 20000 // Allow enough time for motion to complete

set FPOS = 0      // Set current position to 0
set S_IND = 0     // Zero index flag
enable            // Turn controller output on
delay 100                 // Small delay to allow to stabilize

// We need to move at least one full rotation
// to make sure we will cross over index
nmove 370

while (S_IND = 0 & FPOS <= 370)
end
// If the index is not found, the S_IND flag will remain at zero and
// the while loop will exit when FPOS > 370 degrees. Motion will stop
// 370 degrees and the position below will not be reset to zeo.

disable

// Reset FPOS to zero only if index is found
if S_IND = 1
 set FPOS = FPOS - POSI
end

delay 1000
enable
delay 100
move 0
delay 2000        // Give it some time to settle in position
kill
```

disable

1. **XCD Sample Script for a Rotary Stage Demo Routine with NanoCommander**

```
////////////////////////////////////////////////////////////////////////////////
// XCD rotary stage demo script for stage with index pulse and no hard stops.
// Please make sure that the stage is tuned and configured properly per the tuning
// instructions shown in the XCD Firmware manual before running this script.
// The script assumes that the ENR units are in Degrees/Count and all related
// units are in degrees (i.e. VEL in deg/s).
////////////////////////////////////////////////////////////////////////////////

delay 2000

// Slow VEL and ACC for homing
VEL = 60          // deg/sec
ACC = 600         // deg/sec^2

enable
delay 100

////////////////////////////////////////////////////
// Home to index pulse
////////////////////////////////////////////////////
set FPOS = 0      // Set current position to 0
enable            // Turn controller output on
delay 100                 // Small delay to allow to stabilize


// Start moving towards other hard stop
// until we find index

set S_IND = 0     // Zero index flag
enable
delay 100

// Make sure we move at least one full rotation
// to make sure we will cross over index
nmove 370

while (S_IND = 0 & FPOS <= 370)
end

disable

// Reset FPOS to zero only if index is found
if S_IND = 1
```

```
 set FPOS = FPOS - POSI
end

delay 1000

enable
delay 100
move 0
////////////////////////////////////
// End homing
////////////////////////////////////

delay 2000

////////////////////////////////////////
// Initialize variables
////////////////////////////////////////

// V0 = Loop state control variable
// V0 = 1: Fast moves between travel points
// V0 = 2: Step moves between travel points
// V0 = 3: Slow moves between travel points
V0 = 1

V1 = 360          // First position to move to in deg
V2 = 0   // Second position to move to

V10 = 360         // Fast speed in deg/s
V11 = 30          // Slow speed in deg/s
V12 = 10          // Setp size in deg

PEL = 0
////////////////////////////////////////

// Infinite Loop
while 1
        delay 10

        // Back/Forth between positions fast
        // Uses the move command for motion
        if V0 = 1
                V0 = 2
                VEL = V10         // Fast speed
                ACC = 1000
                enable
                delay 100
                for V5 = 1 to 3
                        move V1
```

```
                move V2
        end
        delay 1000
end

// Back/Forth to limits at V12 steps
if V0 = 2
        V0 = 3
        V6 = 20 // Step size in deg

        // Fast VEL and ACC
        VEL = V10
        ACC = 1200

        // V6 steps in (+) rotation
        V7 = FPOS       // Remember current position
        while V7 <= (V1 - 1)
                delay 100
                move (V7 + V12)
                V7 = FPOS
        end
        delay 1000

        // V6 steps in (-) rotation
        V7 = FPOS       // Remember current position
        while V7 >= (V2 + 1)
                delay 100
                move V7 - V12
                V7 = FPOS
        end
        delay 1000
end

// Back/Forth between positions slow
// Uses the nmove/while S_BUSY technique
// to go to position
if V0 = 3
        V0 = 1
        VEL = V11
        MTL = 60000
        for V5 = 1 to 1
                nmove V1
                while S_BUSY
                end
                nmove V2
                while S_BUSY
                end
        end
```

```
            delay 1000
        end

        delay 2000

end

kill
disable
```