

XCD Sample Scripts

XCD Sample Homing Script for a Linear Stage to Home on Index Pulse with NanoCommander

```
////////////////////////////////////  
// XCD sample homing script for a Linear Stage to home at index pulse.  
// 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.  
////////////////////////////////////  
  
delay 2000  
  
// Set slow VEL and ACC for homing  
VEL = 10      // mm/s  
ACC = 100    // mm/sec^2  
MTL = 20000 // Allow enough time for motion to complete  
  
set FPOS = 0  // Set current position to 0  
enable      // Turn controller output on  
delay 100    // Small delay to allow to stabilize  
  
// Start moving to (-) hard stop but continue  
// with script execuion after motion start.  
nmove -1000  
// Keep checking Position Error. When PE > 0.5 mm  
// we have found the hard stop.  
while abs(PE) < 0.5  
end  
// We found hard stop  
disable  
  
delay 500  
  
// Start moving towards other hard stop  
// until we find index  
  
set S_IND = 0 // Zero index flag  
nmove 1000   // Start moving towards index  
  
while (S_IND = 0)  
end  
disable  
// We found index.  
// Caution: if index is not found, (due to a hardware issue for example),  
// the stage will continue to move towards hard stop and an error will be  
// triggered when PE > PEL or MTL is exceeded. At the final system configuration  
// it is recommended to set PE and MTL accordingly and should not be set at zero
```

// (disabled) in order to prevent motor or controller damage.

set FPOS = FPOS - POSI

```
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 Linear Stage to Home at Hard Stop with NanoCommander

```
////////////////////////////////////
// XCD sample homing script for a linear stage to home at hard stop.
// 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.
////////////////////////////////////
```

delay 2000

```
// Set slow VEL and ACC for homing
VEL = 10 // mm/s
ACC = 100 // mm/sec^2
MTL = 20000 // Allow enough time for motion to complete
```

set FPOS = 0

```
// V1 = Small buffer region to home away from hard stop.
// (This is used as a safety mechanism so we are not positioned
// (right against the hard stop).
V1 = 0.5
```

```
enable // Turn output on
delay 100 // Allow some time to settle
nmove -1000 // start moving towards (-) hard stop
while abs(PE) < 0.5
end
disable
```

```
// Set current position to -V1 so the
// zero position is V1 away from hard stop
set FPOS = -V1
```

delay 500

```
// Move forward to the zero position
enable
delay 100
move 0 // Move to origin
delay 2000 // Allow some time to settle into position
kill
disable
```

2. XCD Sample Script for a Linear Stage Demo Routine with NanoCommander

```
////////////////////////////////////
// XCD linear stage demo script with encoder index pulse.
// 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.
////////////////////////////////////
```

```
delay 2000
```

```
VEL = 10
ACC = 100
```

```
enable
delay 100
```

```
// Use the XCD automated homing routine
// to home at index pulse
home 60
```

```
delay 2000
```

```
//*****
// Find travel limits of stage
//*****
nmove 1000 // Go to positive hard stop
while abs(PE) < 0.5
end
disable
// Set V1 = Positive travel limit 0.5 mm from hard stop
V1 = FPOS - 0.5
```

```
delay 1000
```

```
enable
delay 100
nmove -1000 // Go to negative hard stop
while abs(PE) < 0.5
end
```

```

disable
// Set V2 = Negative travel limit 0.5 mm from hard stop
V2 = FPOS + 0.5
//*****

delay 1000

enable
delay 100
move 0

delay 2000

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

V0 = 1

// Infinite Loop
while 1
    delay 10

    // Fast back/forth moves between travel limits (V1 and V2)
    // Uses the move command for motion
    if V0 = 1
        V0 = 2

        // Fast VEL and ACC
        VEL = 100
        ACC = 1000
        //enable
        //delay 100
        for V5 = 1 to 5
            move V1
            move V2
        end
        delay 1000
    end

    // Step moves between travel limits at V6 size steps
    if V0 = 2
        V0 = 3
        V6 = 1 // Step size in mm

        // Fast VEL and ACC
        VEL = 100

```

```

ACC = 1000

// 1mm steps toward (+) travel limit
V7 = FPOS // Remember current position
while V7 <= (V1 - 1)
    delay 100
    move (V7 + V6)
    V7 = FPOS
end
delay 1000

// 1mm steps toward (-) travel limit
V7 = FPOS // Remember current position
while V7 >= (V2 + 1)
    delay 100
    move V7 - V6
    V7 = FPOS
end
delay 1000
end

// Slow back/forth moves between travel limits (V1 and V2)
// Uses the nmove/while S_BUSY technique to go to position
if V0 = 3
    V0 = 1
    VEL = 5 // Slow velocity
    MTL = 60000 // Increase motion time limit
    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

```