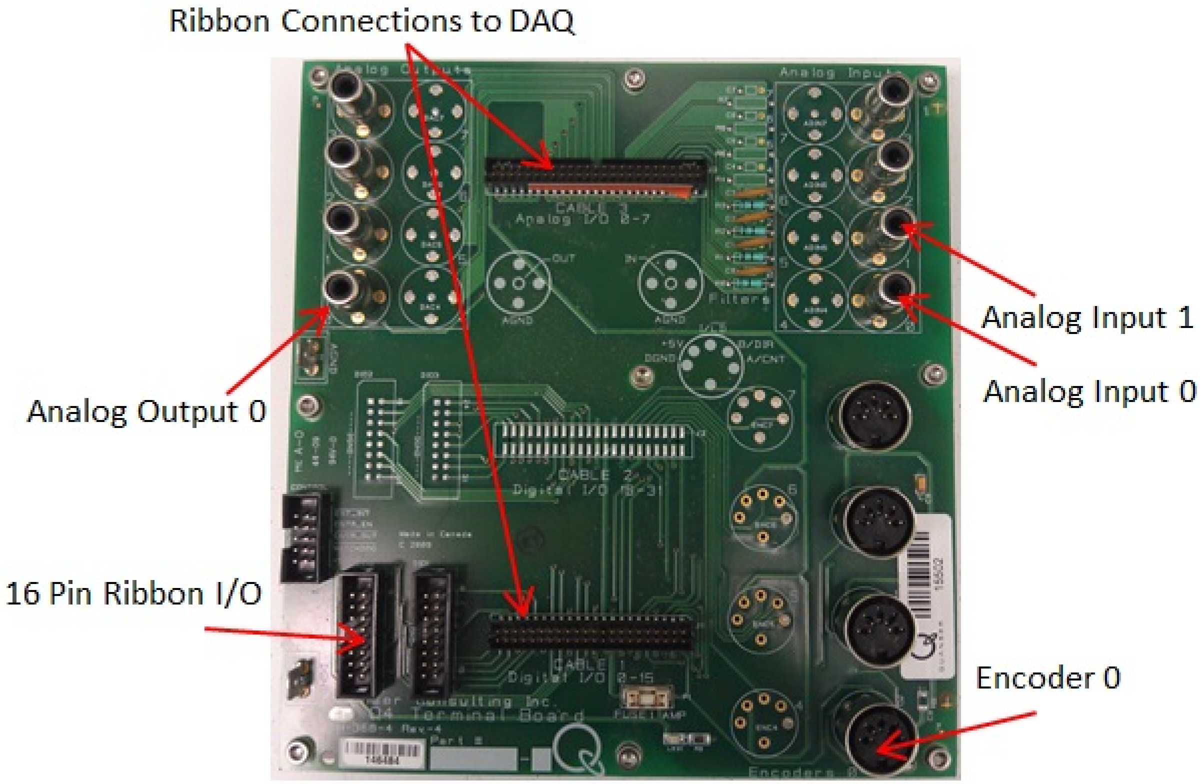
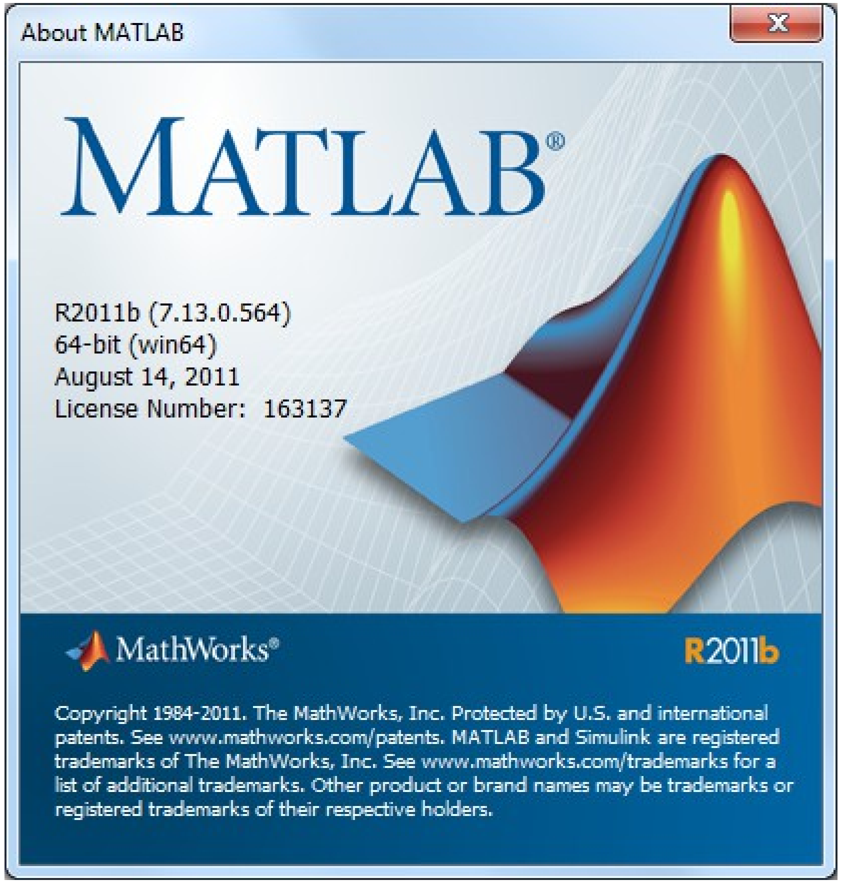
ECE 147C
Getting started with the hardware & software

Justin Pearson

2014.04.07

Our current setup: Windows 7,
MATLAB R2011B,
Quanser Q4 terminal board



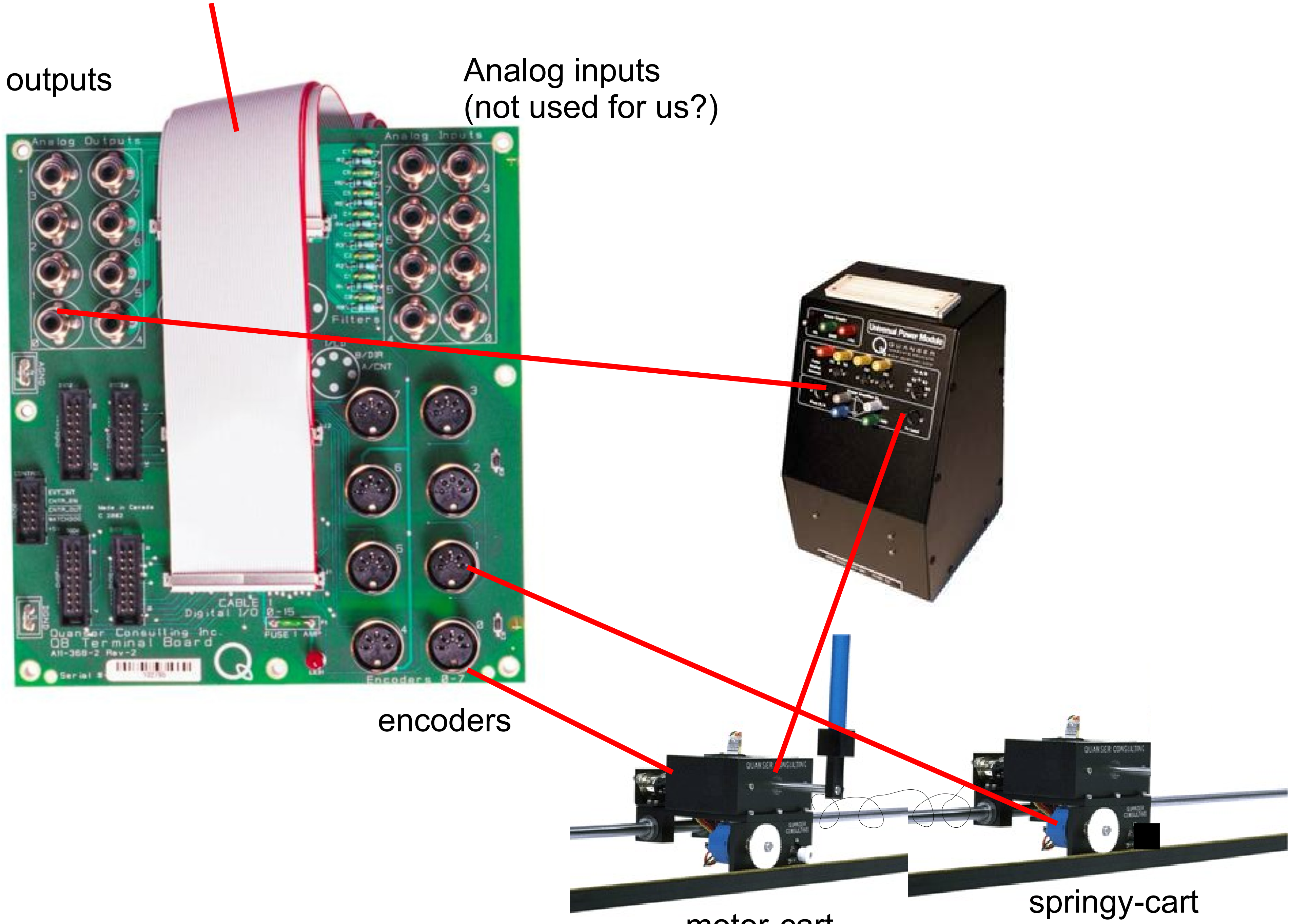


R2011b (7.13.0.564)
64-bit (win64)
August 14, 2011
License Number: 163137

Copyright 1984-2011. The MathWorks, Inc. Protected by U.S. and international patents. See www.mathworks.com/patents. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

http://www.clemson.edu/ces/crb/ece495/past_projects/MagnaDoodle/Fall2013Websites/Team%2017/Images/P1_F2_Quanser%20Q4%20HIL%20Board%20Connections.jpg

Hardware setup



Win 7 computer

Analog outputs

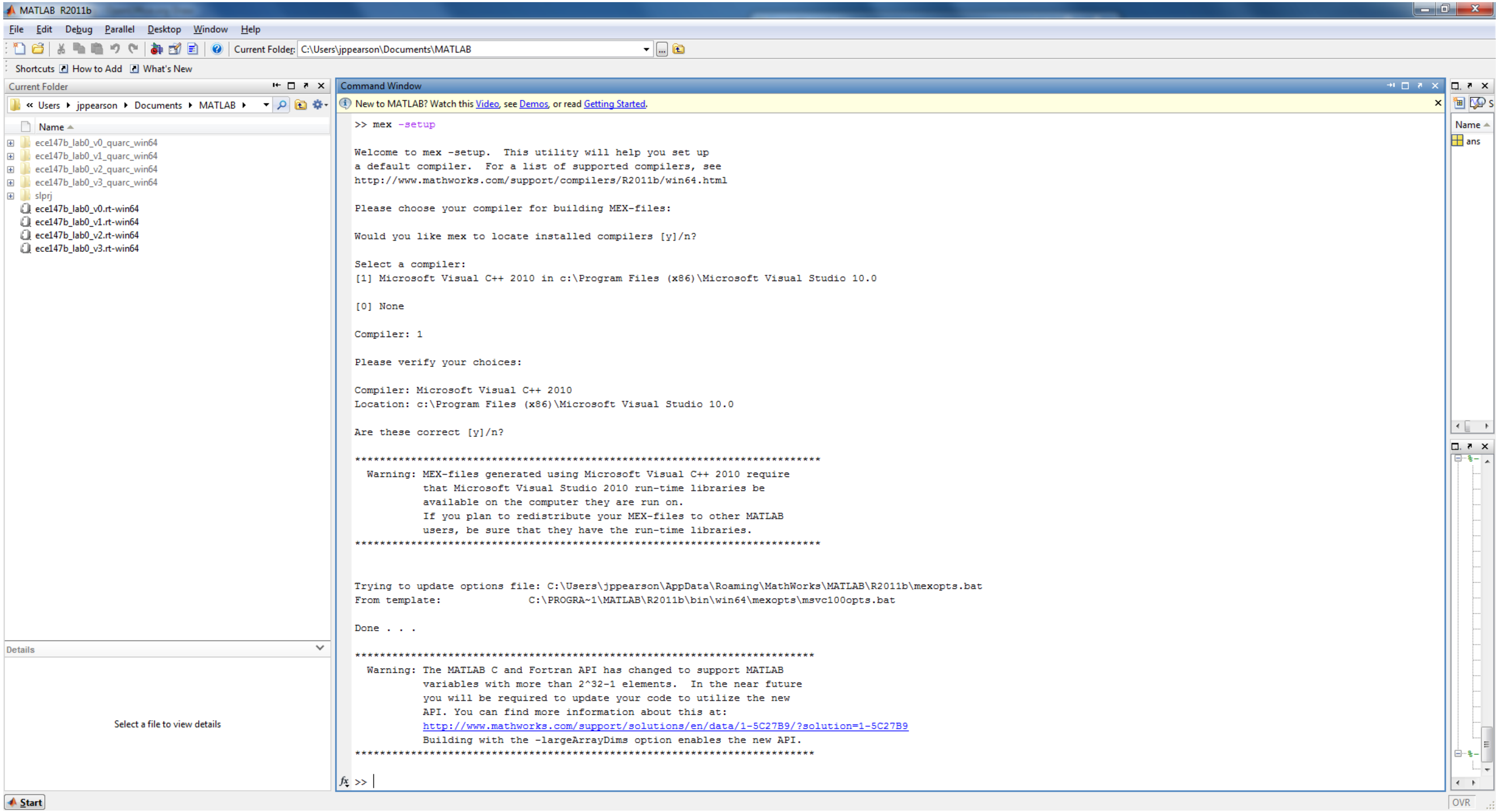
Analog inputs
(not used for us?)

encoders

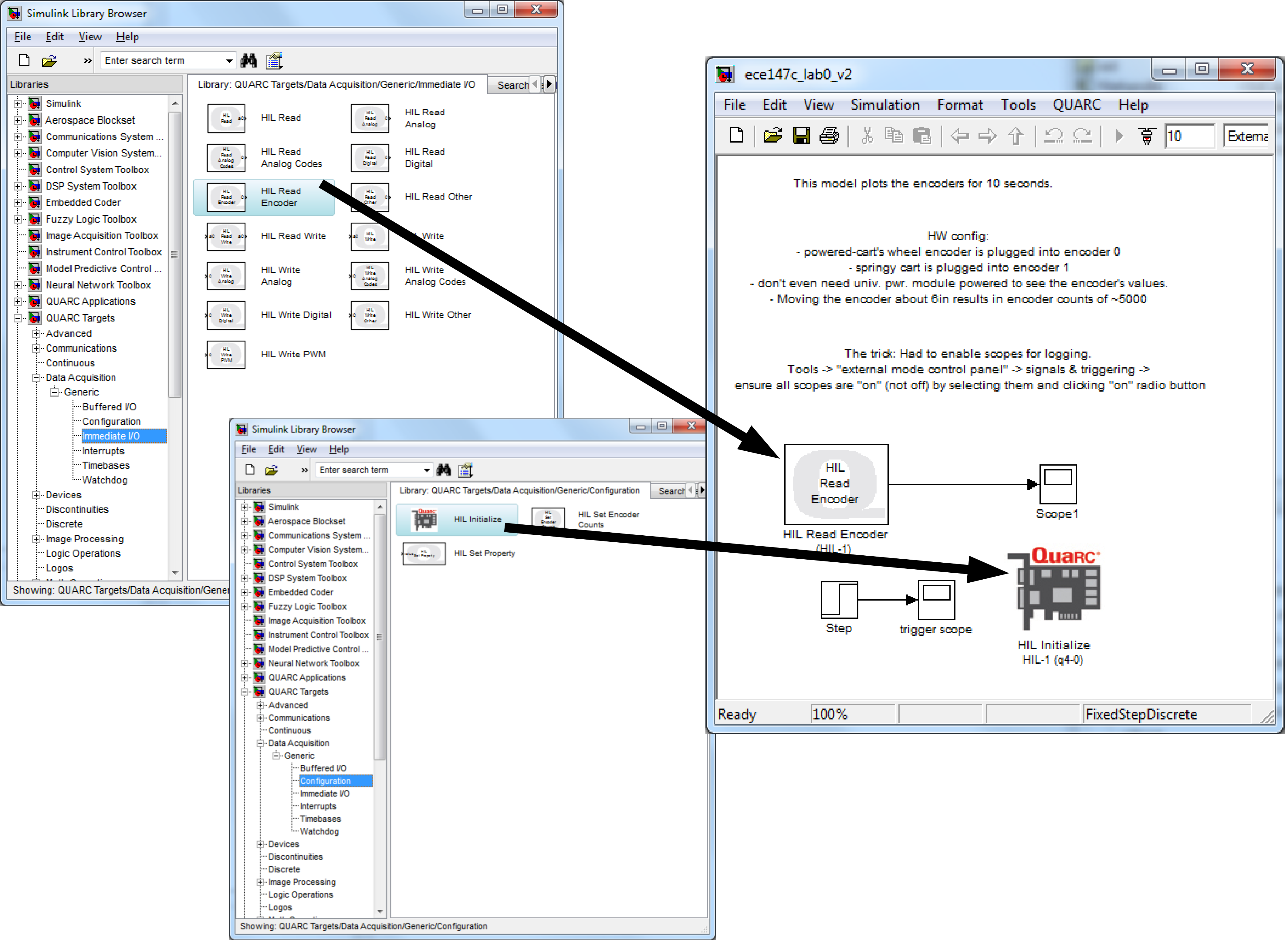
motor-cart

springy-cart

How to build a simple Simulink model to read the encoders


MATLAB R2011b
File
Edit
Debug
Parallel
Desktop
Window
Help
Current Folder: C:\Users\jppearson\Documents\MATLAB
Shortcuts
How to Add
What's New
Current Folder
« Users ▸ jppearson ▸ Documents ▸ MATLAB ▸
Name
ece147b_lab0_v0_quarc_win64
ece147b_lab0_v1_quarc_win64
ece147b_lab0_v2_quarc_win64
ece147b_lab0_v3_quarc_win64
slprj
ece147b_lab0_v0.rt-win64
ece147b_lab0_v1.rt-win64
ece147b_lab0_v2.rt-win64
ece147b_lab0_v3.rt-win64
Details
Select a file to view details
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Name
ans
Start
OVR

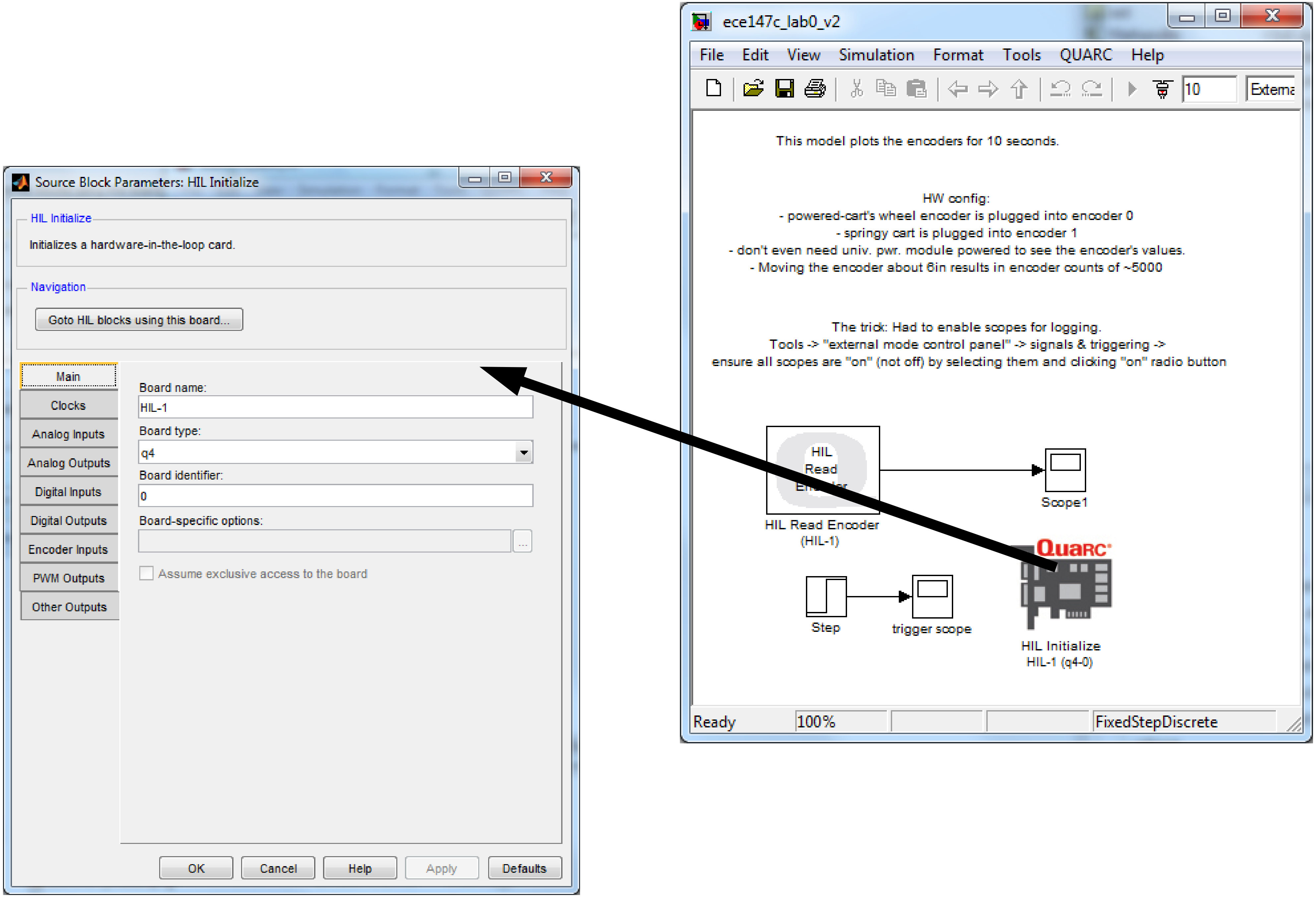
```
>> mex -setup

Welcome to mex -setup.  This utility will help you set up
a default compiler.  For a list of supported compilers, see
http://www.mathworks.com/support/compilers/R2011b/win64.html

Please choose your compiler for building MEX-files:

Would you like mex to locate installed compilers [y]/n?

Select a compiler:
[1] Microsoft Visual C++ 2010 in c:\Program Files (x86)\Microsoft Visual Studio 10.0

[0] None

Compiler: 1

Please verify your choices:

Compiler: Microsoft Visual C++ 2010
Location: c:\Program Files (x86)\Microsoft Visual Studio 10.0

Are these correct [y]/n?

***************************************************************************
  Warning: MEX-files generated using Microsoft Visual C++ 2010 require
           that Microsoft Visual Studio 2010 run-time libraries be
           available on the computer they are run on.
           If you plan to redistribute your MEX-files to other MATLAB
           users, be sure that they have the run-time libraries.
***************************************************************************


Trying to update options file: C:\Users\jppearson\AppData\Roaming\MathWorks\MATLAB\R2011b\mexopts.bat
From template:              C:\PROGRA~1\MATLAB\R2011b\bin\win64\mexopts\msvc100opts.bat

Done . . .

**************************************************************************
  Warning: The MATLAB C and Fortran API has changed to support MATLAB
           variables with more than 2^32-1 elements.  In the near future
           you will be required to update your code to utilize the new
           API. You can find more information about this at:
           http://www.mathworks.com/support/solutions/en/data/1-5C27B9/?solution=1-5C27B9
           Building with the -largeArrayDims option enables the new API.
**************************************************************************

>> 
```



This model plots the encoders for 10 seconds.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button



Source Block Parameters: HIL Initialize

HIL Initialize

Initializes a hardware-in-the-loop card.

Navigation

Board name: HIL-1

Board type: q4

Board identifier: 0

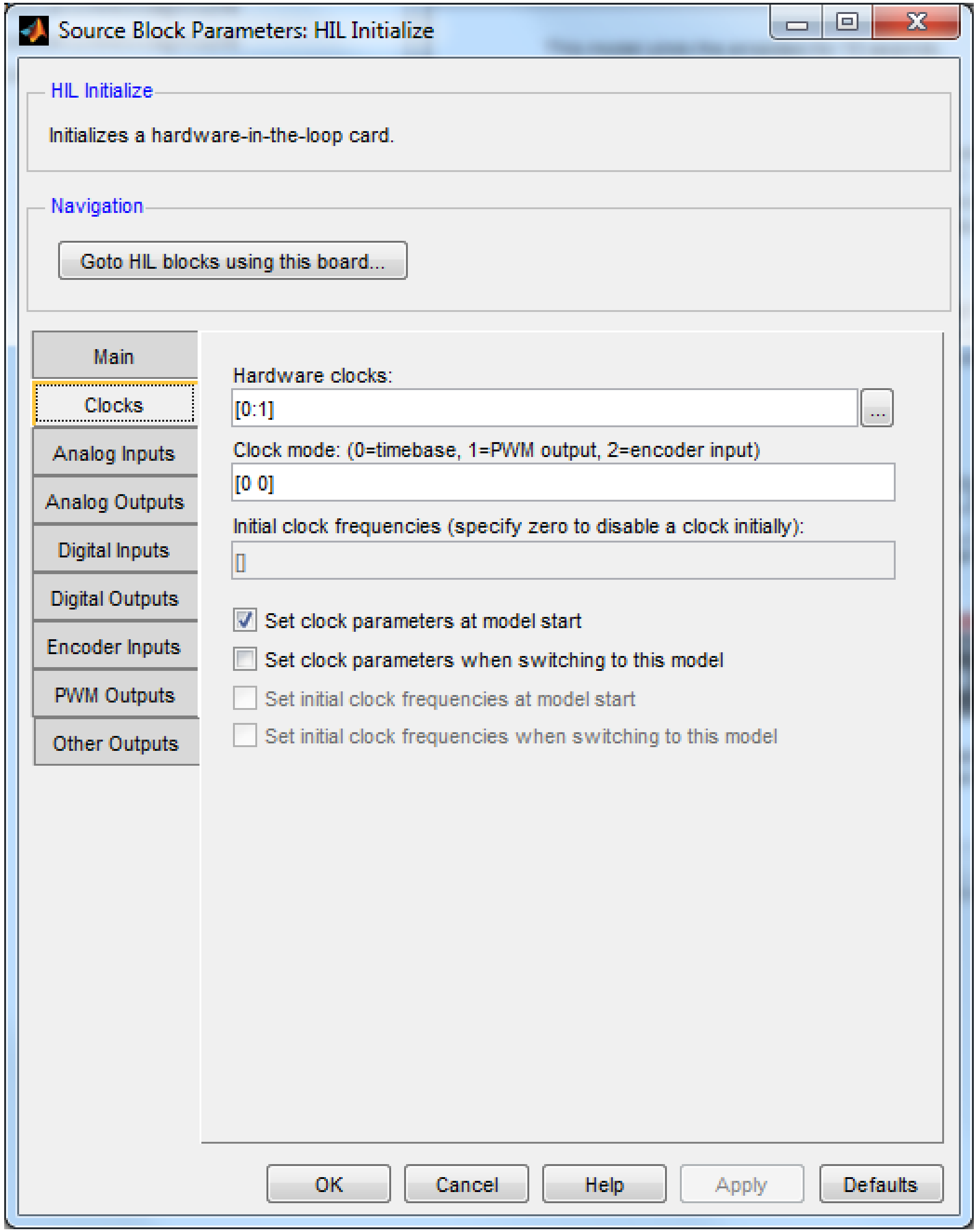
Board-specific options:

☐ Assume exclusive access to the board

This model plots the encoders for 10 seconds.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button



HIL Initialize

Initializes a hardware-in-the-loop card.

Navigation

Hardware clocks: [0:1]

Clock mode: (0=timebase, 1=PWM output, 2=encoder input) [0 0]

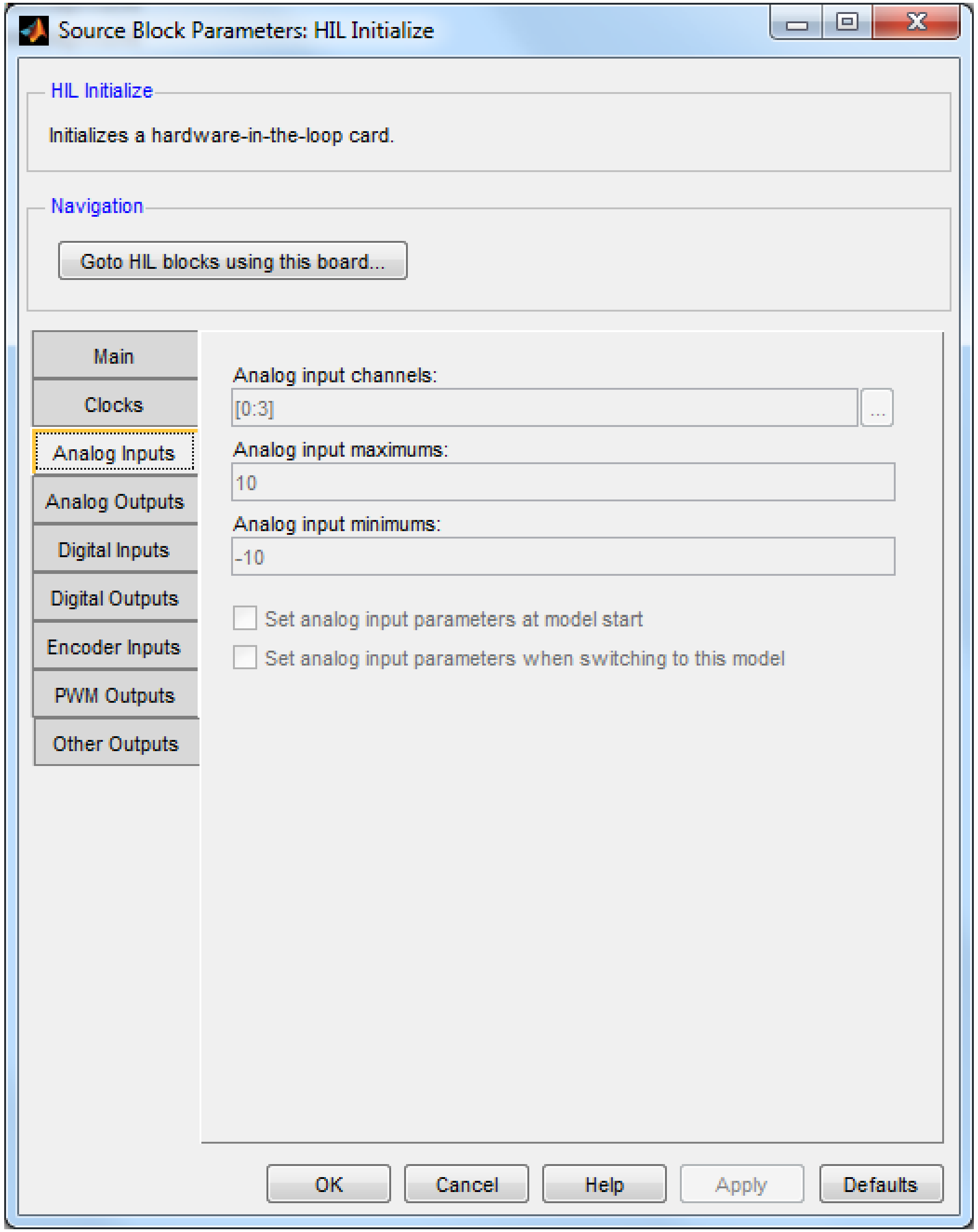
Initial clock frequencies (specify zero to disable a clock initially): []

☑ Set clock parameters at model start

☐ Set clock parameters when switching to this model

☐ Set initial clock frequencies at model start

☐ Set initial clock frequencies when switching to this model



HIL Initialize

Initializes a hardware-in-the-loop card.

Navigation

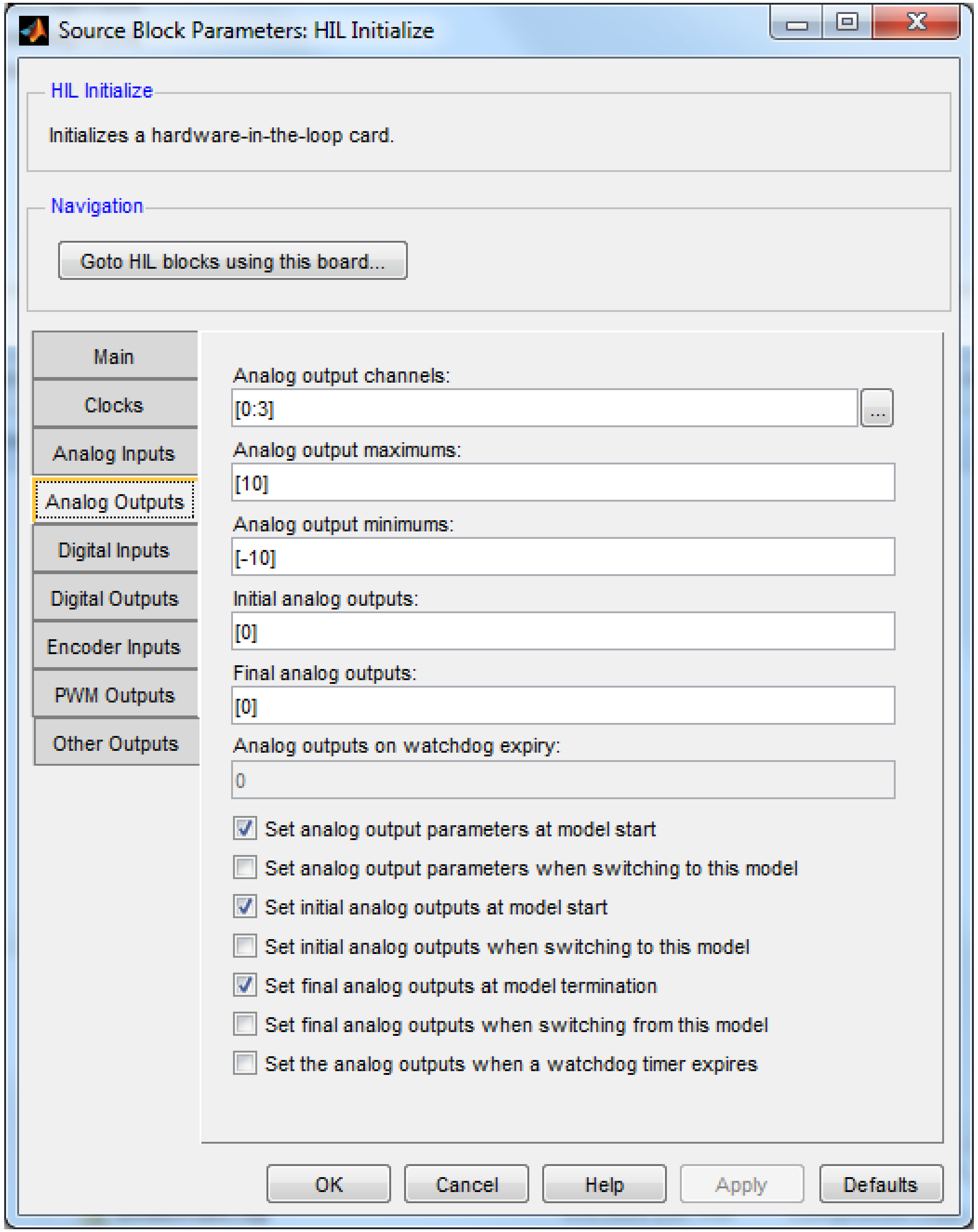
Analog input channels: [0:3]

Analog input maximums: 10

Analog input minimums: -10

☐ Set analog input parameters at model start

☐ Set analog input parameters when switching to this model



Initializes a hardware-in-the-loop card.

Analog output channels: [0:3]

Analog output maximums: [10]

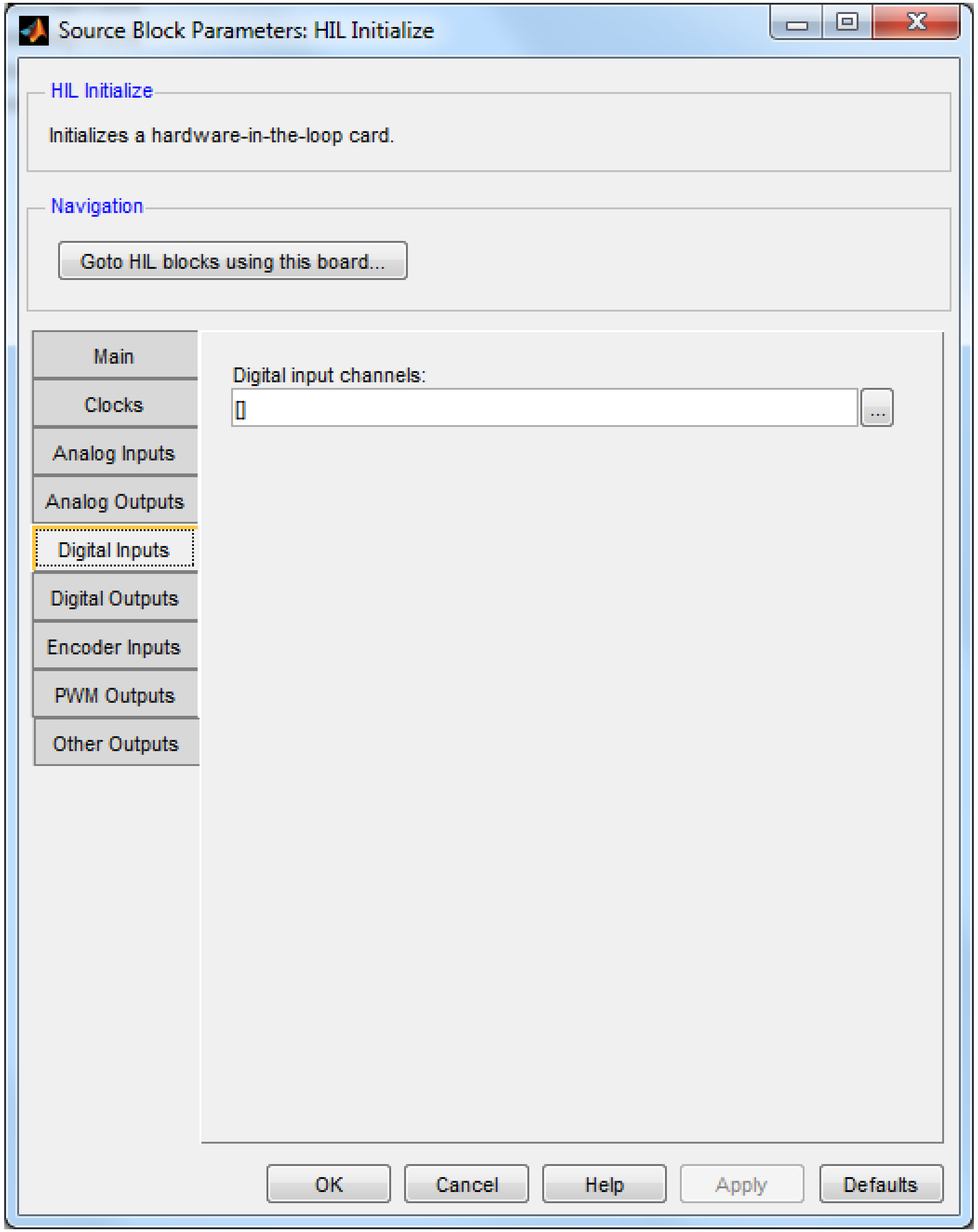
Analog output minimums: [-10]

Initial analog outputs: [0]

Final analog outputs: [0]

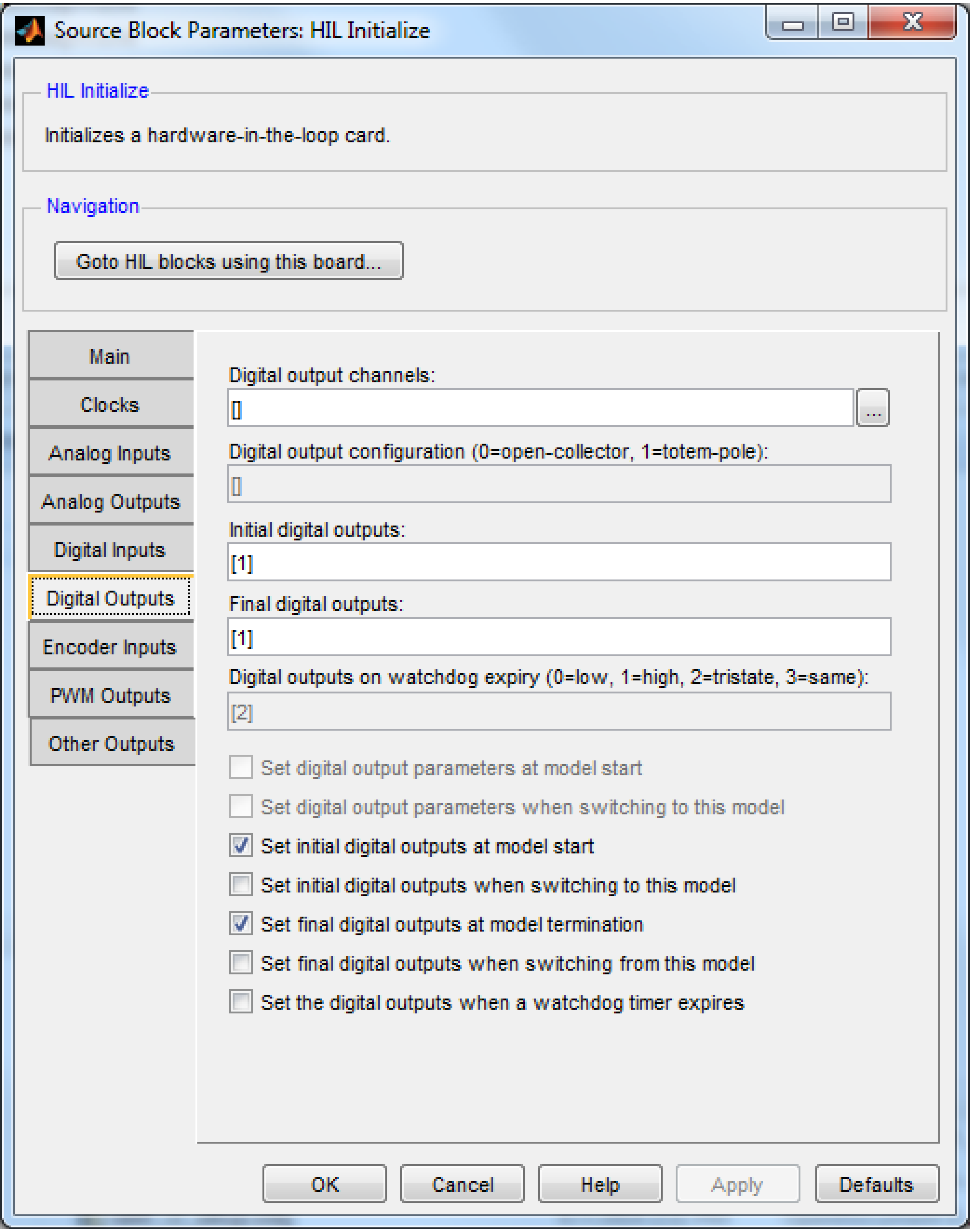
Analog outputs on watchdog expiry: 0

☑ Set analog output parameters at model start
☐ Set analog output parameters when switching to this model
☑ Set initial analog outputs at model start
☐ Set initial analog outputs when switching to this model
☑ Set final analog outputs at model termination
☐ Set final analog outputs when switching from this model
☐ Set the analog outputs when a watchdog timer expires



Initializes a hardware-in-the-loop card.

Digital input channels: []



HIL Initialize

Initializes a hardware-in-the-loop card.

Navigation

Digital output channels:
[]

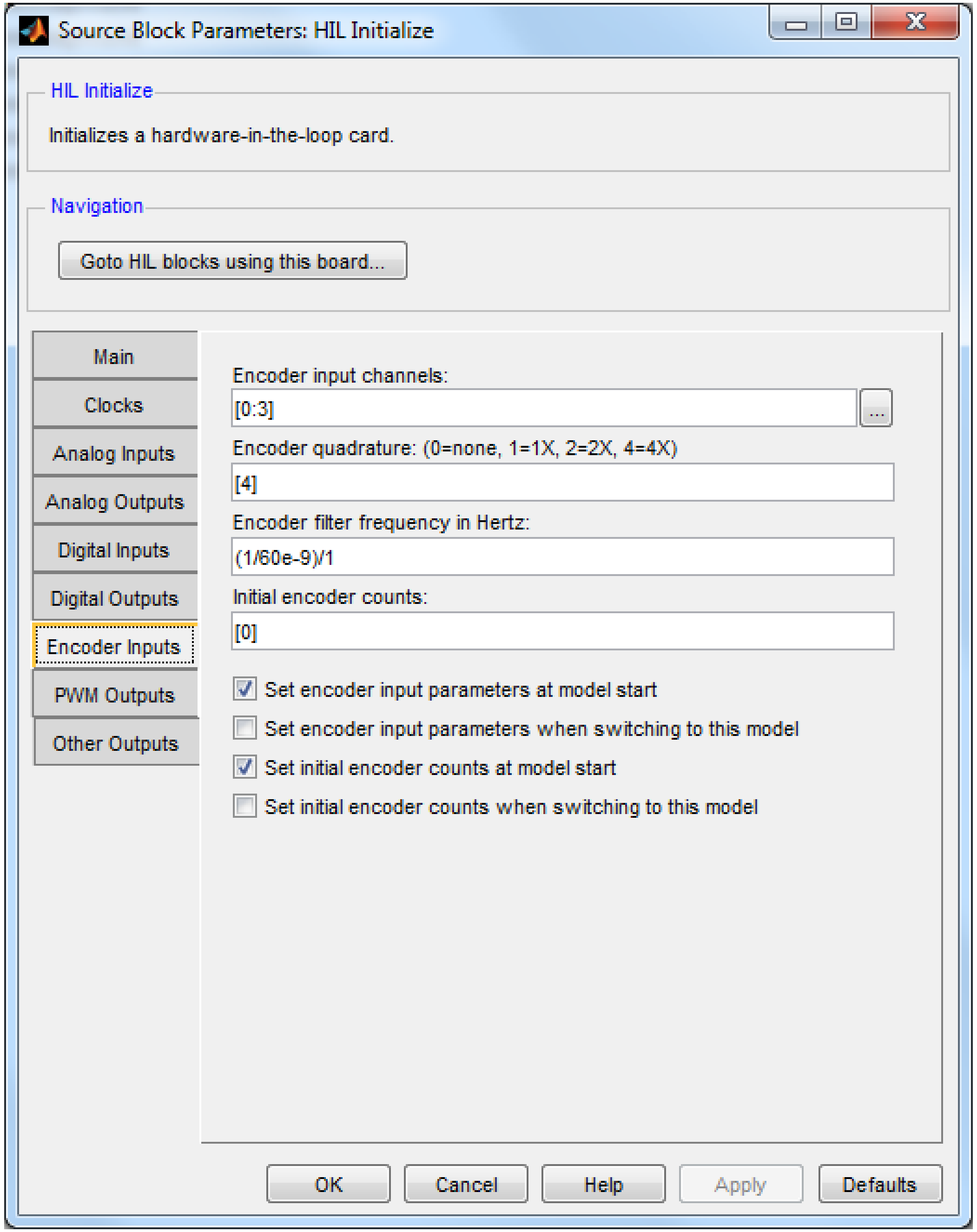
Digital output configuration (0=open-collector, 1=totem-pole):
[]

Initial digital outputs:
[1]

Final digital outputs:
[1]

Digital outputs on watchdog expiry (0=low, 1=high, 2=tristate, 3=same):
[2]

□ Set digital output parameters at model start
□ Set digital output parameters when switching to this model
☑ Set initial digital outputs at model start
□ Set initial digital outputs when switching to this model
☑ Set final digital outputs at model termination
□ Set final digital outputs when switching from this model
□ Set the digital outputs when a watchdog timer expires



HIL Initialize

Initializes a hardware-in-the-loop card.

Navigation

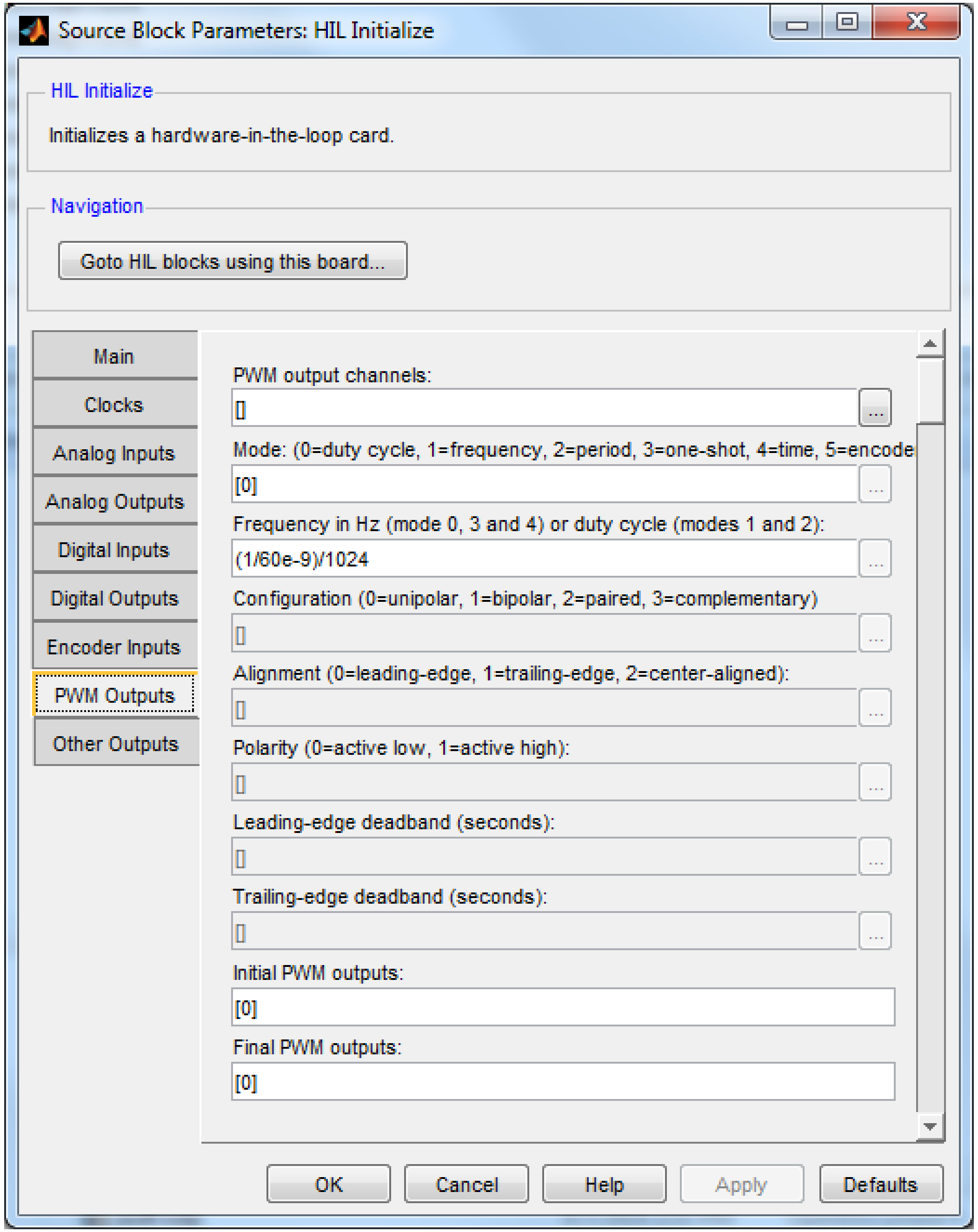
Encoder input channels:
[0:3]

Encoder quadrature: (0=none, 1=1X, 2=2X, 4=4X)
[4]

Encoder filter frequency in Hertz:
(1/60e-9)/1

Initial encoder counts:
[0]

☑ Set encoder input parameters at model start
□ Set encoder input parameters when switching to this model
☑ Set initial encoder counts at model start
□ Set initial encoder counts when switching to this model



Source Block Parameters: HIL Initialize

HIL Initialize

Initializes a hardware-in-the-loop card.

Navigation

PWM output channels: []

Mode: (0=duty cycle, 1=frequency, 2=period, 3=one-shot, 4=time, 5=encode [0]

Frequency in Hz (mode 0, 3 and 4) or duty cycle (modes 1 and 2): (1/60e-9)/1024

Configuration (0=unipolar, 1=bipolar, 2=paired, 3=complementary) []

Alignment (0=leading-edge, 1=trailing-edge, 2=center-aligned): []

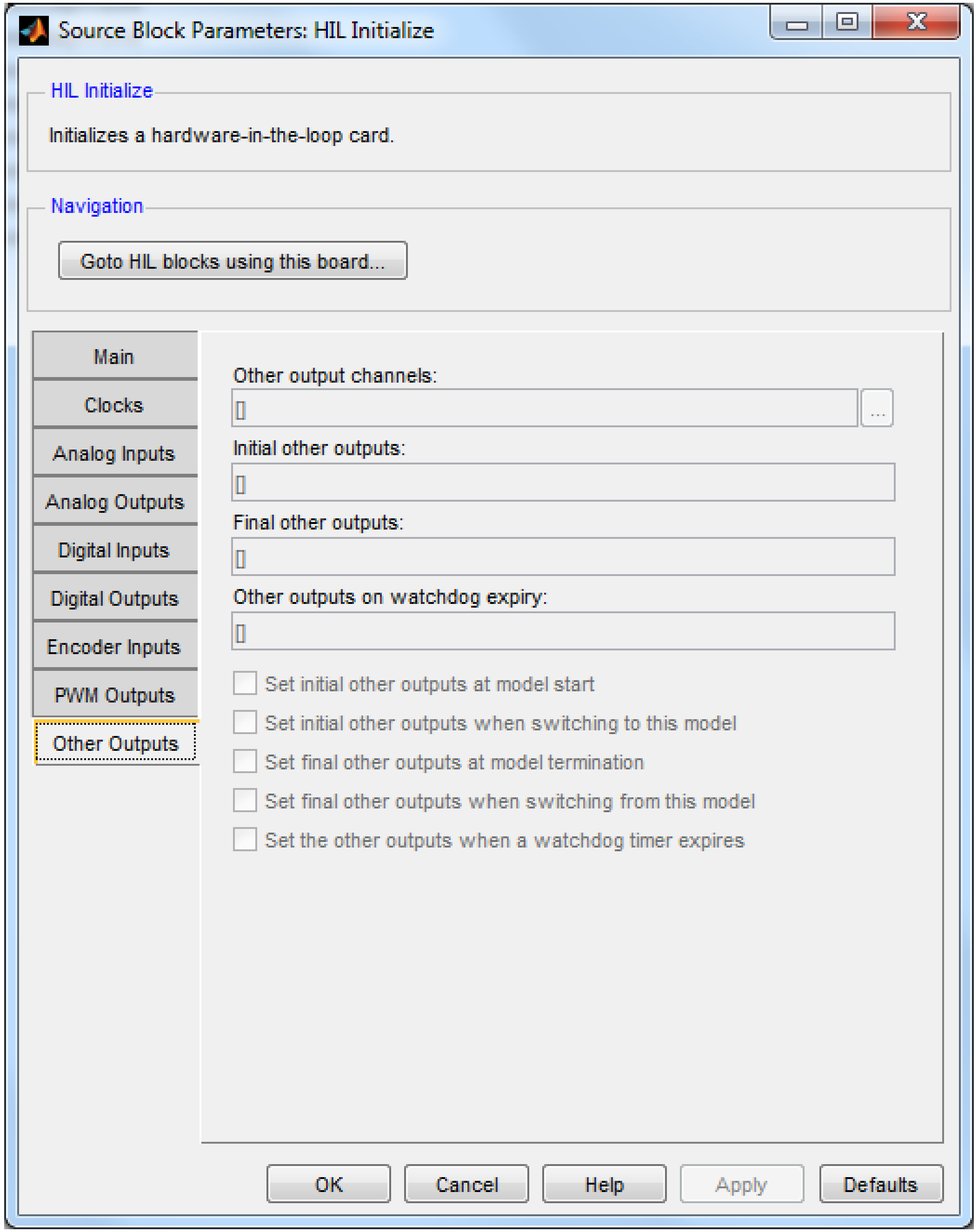
Polarity (0=active low, 1=active high): []

Leading-edge deadband (seconds): []

Trailing-edge deadband (seconds): []

Initial PWM outputs: [0]

Final PWM outputs: [0]



Source Block Parameters: HIL Initialize

HIL Initialize

Initializes a hardware-in-the-loop card.

Navigation

Other output channels: []

Initial other outputs: []

Final other outputs: []

Other outputs on watchdog expiry: []

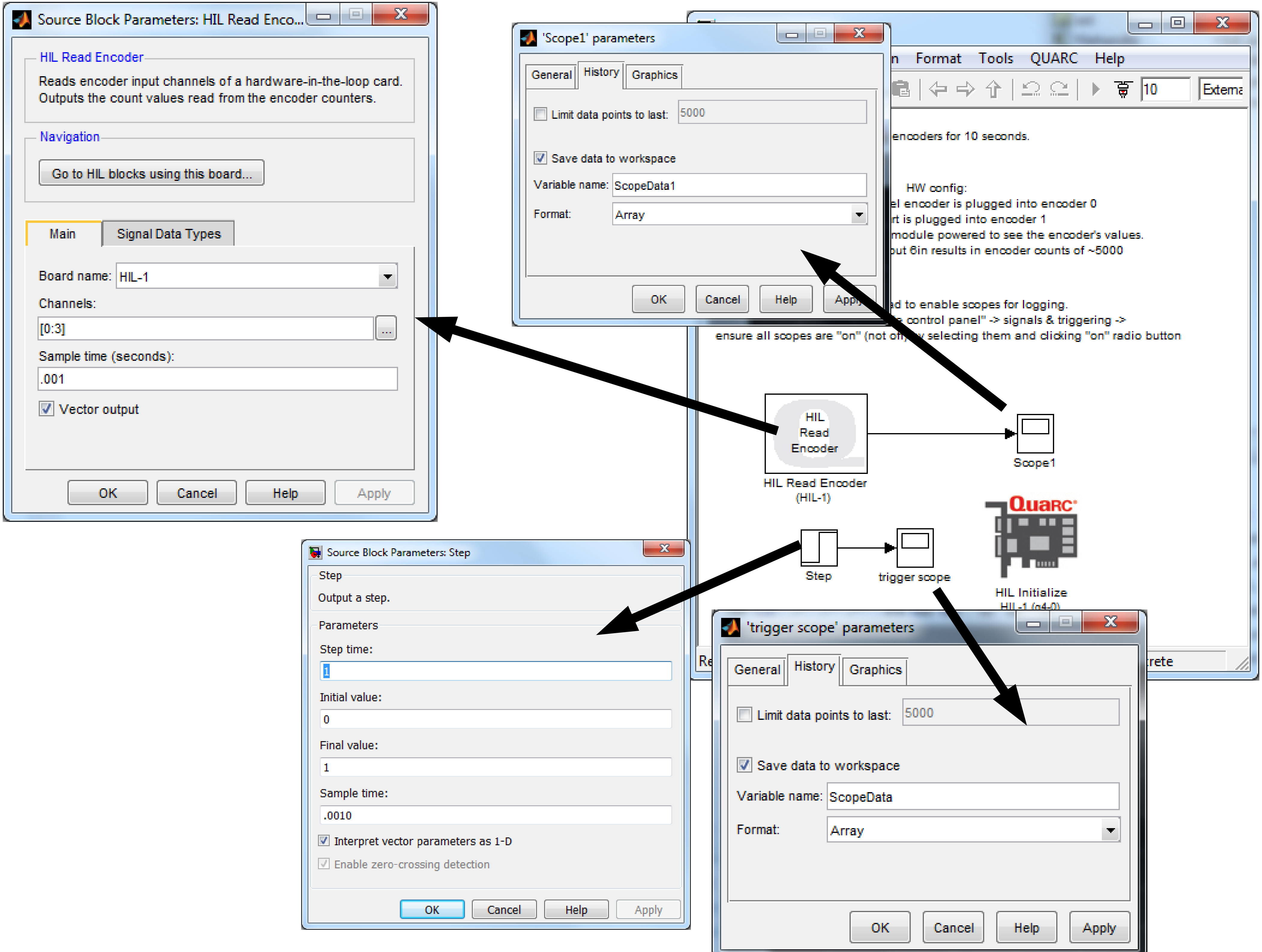
□ Set initial other outputs at model start

□ Set initial other outputs when switching to this model

□ Set final other outputs at model termination

□ Set final other outputs when switching from this model

□ Set the other outputs when a watchdog timer expires



Source Block Parameters: HIL Read Enco...

HIL Read Encoder

Reads encoder input channels of a hardware-in-the-loop card. Outputs the count values read from the encoder counters.

Navigation

Go to HIL blocks using this board...

Main | Signal Data Types

Board name: HIL-1

Channels: [0:3]

Sample time (seconds): .001

☑ Vector output

OK | Cancel | Help | Apply

'Scope1' parameters

General | History | Graphics

☐ Limit data points to last: 5000

☑ Save data to workspace

Variable name: ScopeData1

Format: Array

OK | Cancel | Help | Apply

encoders for 10 seconds.

HW config:

el encoder is plugged into encoder 0

rt is plugged into encoder 1

module powered to see the encoder's values.

put 6in results in encoder counts of ~5000

d to enable scopes for logging.

control panel" -> signals & triggering ->

ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

Source Block Parameters: Step

Step

Output a step.

Parameters

Step time: 1

Initial value: 0

Final value: 1

Sample time: .0010

☑ Interpret vector parameters as 1-D

☑ Enable zero-crossing detection

OK | Cancel | Help | Apply

'trigger scope' parameters

General | History | Graphics

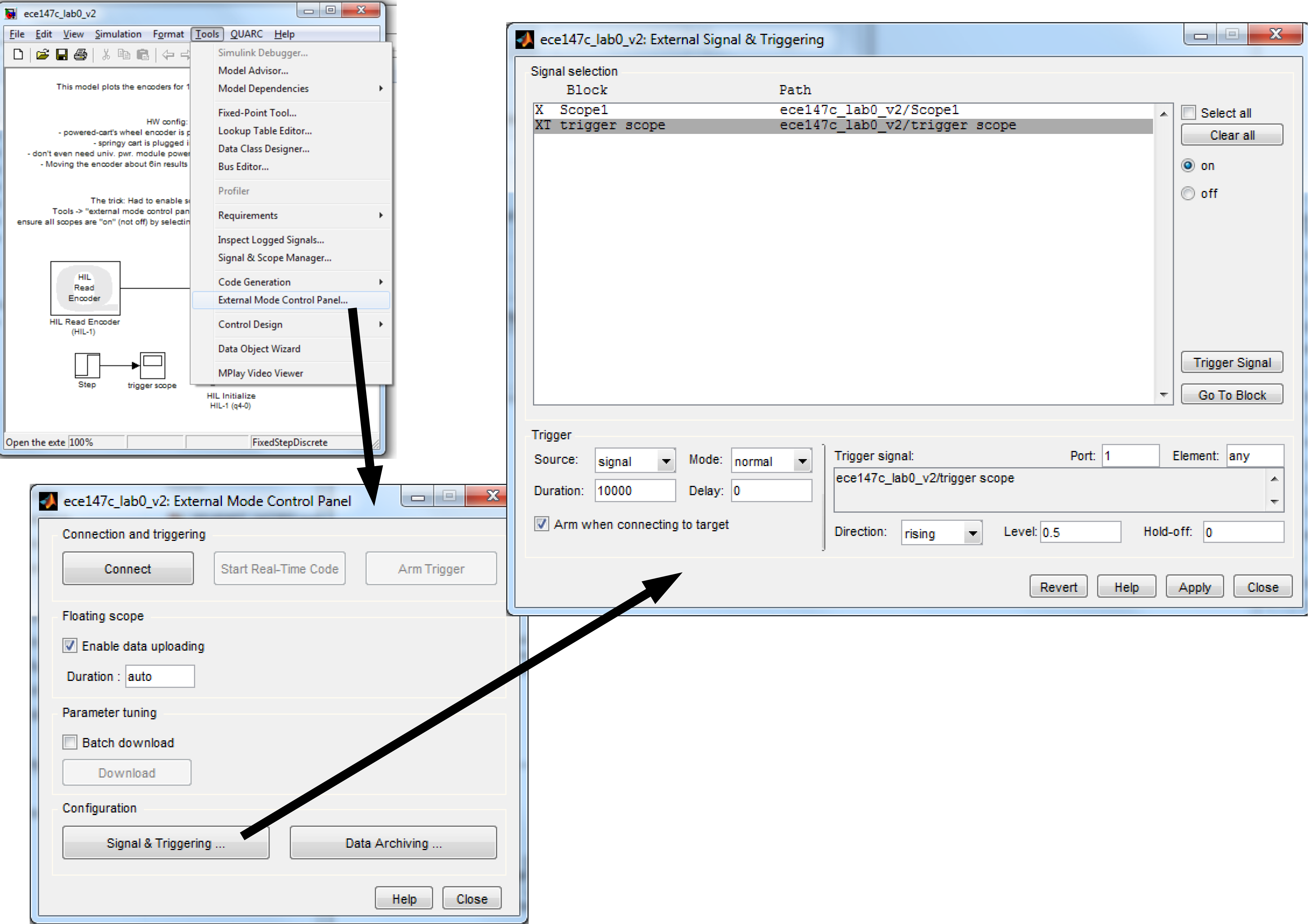
☐ Limit data points to last: 5000

☑ Save data to workspace

Variable name: ScopeData

Format: Array

OK | Cancel | Help | Apply



This model plots the encoders for 1

HW config:
- powered-cart's wheel encoder is p
- springy cart is plugged i
- don't even need univ. pwr. module powe
- Moving the encoder about 6in results

The trick: Had to enable s
Tools -> "external mode control pan
ensure all scopes are "on" (not off) by selectin

ece147c_lab0_v2: External Mode Control Panel
Connection and triggering
Connect
Start Real-Time Code
Arm Trigger
Floating scope

☑ Enable data uploading

Duration : auto

□ Batch download

	Block	Path
X	Scope1	ece147c_lab0_v2/Scope1
XT	trigger scope	ece147c_lab0_v2/trigger scope

□ Select all

● on

○ off

Source: signal Mode: normal

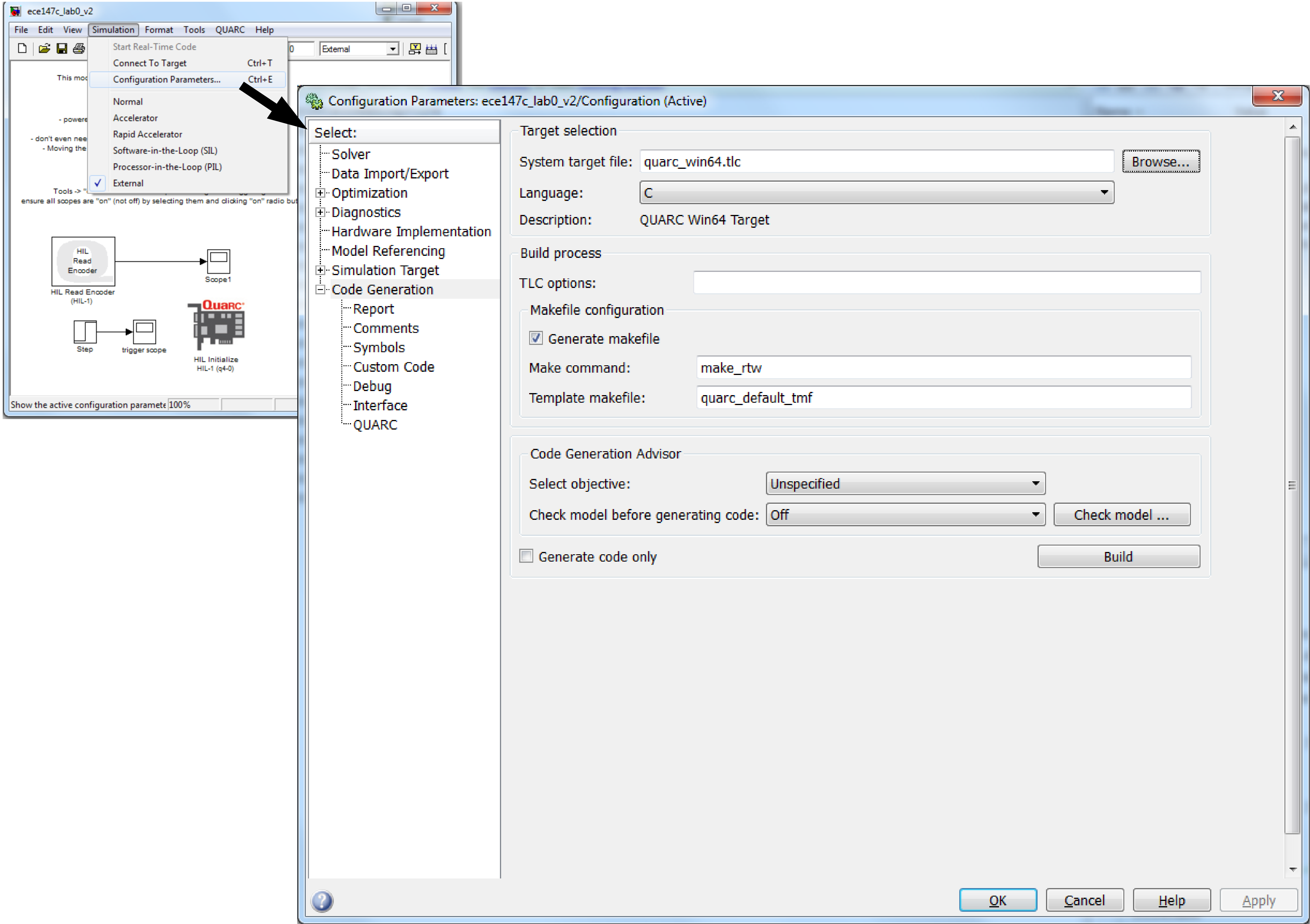
Duration: 10000 Delay: 0

☑ Arm when connecting to target

Trigger signal: Port: 1 Element: any

ece147c_lab0_v2/trigger scope

Direction: rising Level: 0.5 Hold-off: 0



Configuration Parameters: ece147c_lab0_v2/Configuration (Active)

Target selection

System target file: quarc_win64.tlc

Language: C

Description: QUARC Win64 Target

Build process

TLC options:

Makefile configuration

☑ Generate makefile

Make command: make_rtw

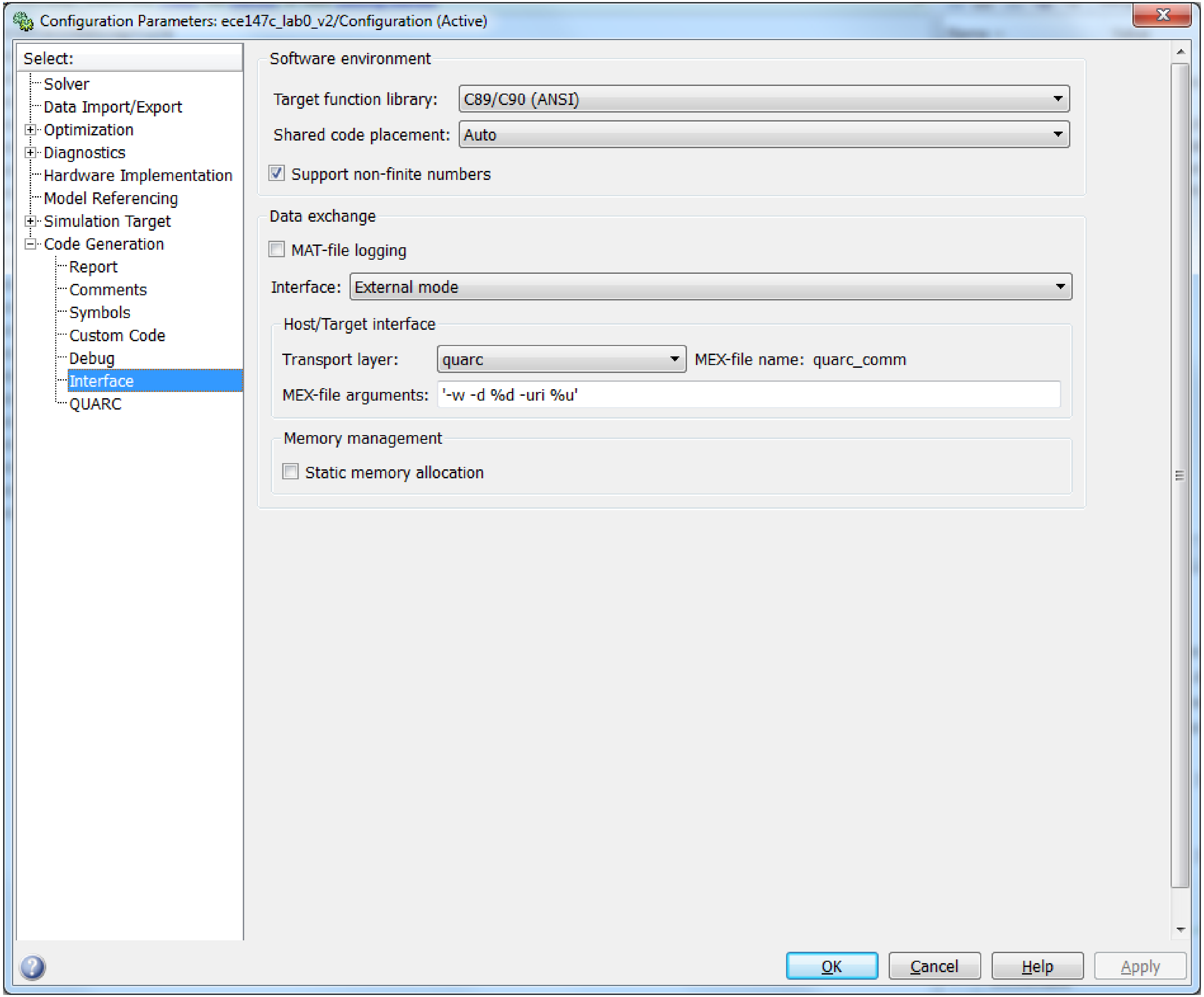
Template makefile: quarc_default_tmf

Code Generation Advisor

Select objective: Unspecified

Check model before generating code: Off

□ Generate code only



Software environment

Target function library: C89/C90 (ANSI)

Shared code placement: Auto

☑ Support non-finite numbers

Data exchange

☐ MAT-file logging

Interface: External mode

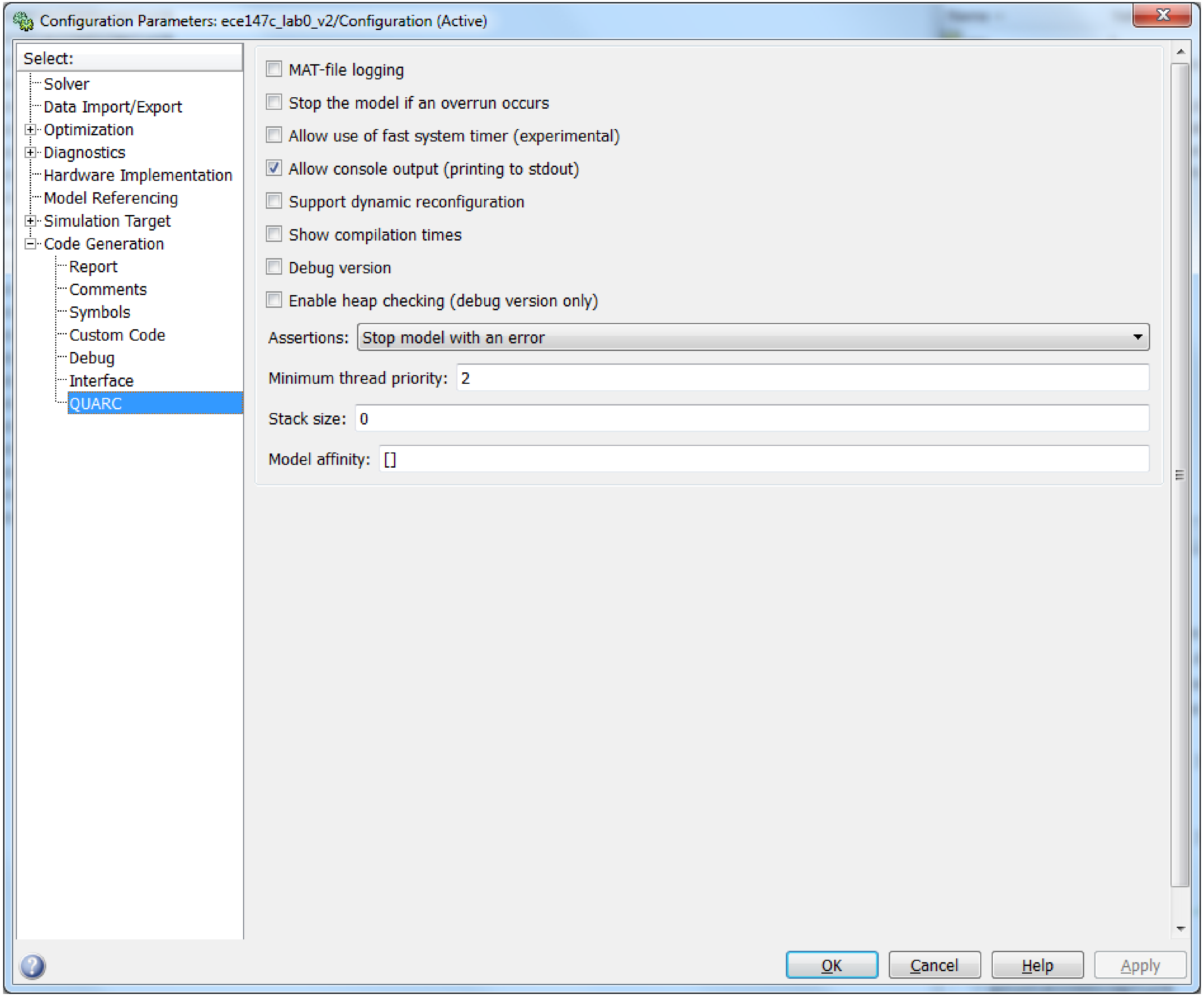
Host/Target interface

Transport layer: quarc MEX-file name: quarc_comm

MEX-file arguments: '-w -d %d -uri %u'

Memory management

☐ Static memory allocation



Configuration Parameters: ece147c_lab0_v2/Configuration (Active)

□ MAT-file logging

□ Stop the model if an overrun occurs

□ Allow use of fast system timer (experimental)

☑ Allow console output (printing to stdout)

□ Support dynamic reconfiguration

□ Show compilation times

□ Debug version

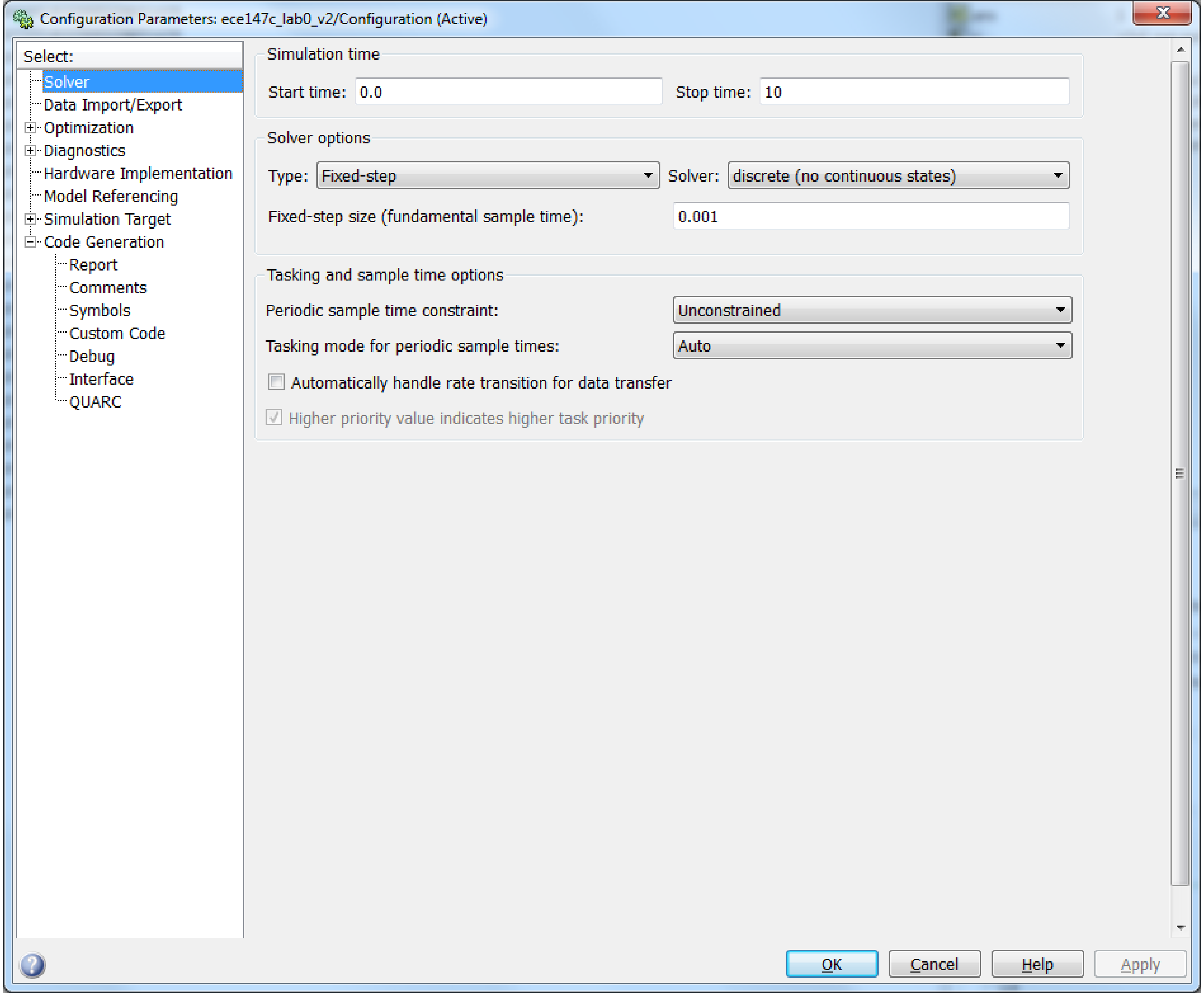
□ Enable heap checking (debug version only)

Assertions: Stop model with an error

Minimum thread priority: 2

Stack size: 0

Model affinity: []



Configuration Parameters: ece147c_lab0_v2/Configuration (Active)

Simulation time

Start time: 0.0 Stop time: 10

Solver options

Type: Fixed-step Solver: discrete (no continuous states)

Fixed-step size (fundamental sample time): 0.001

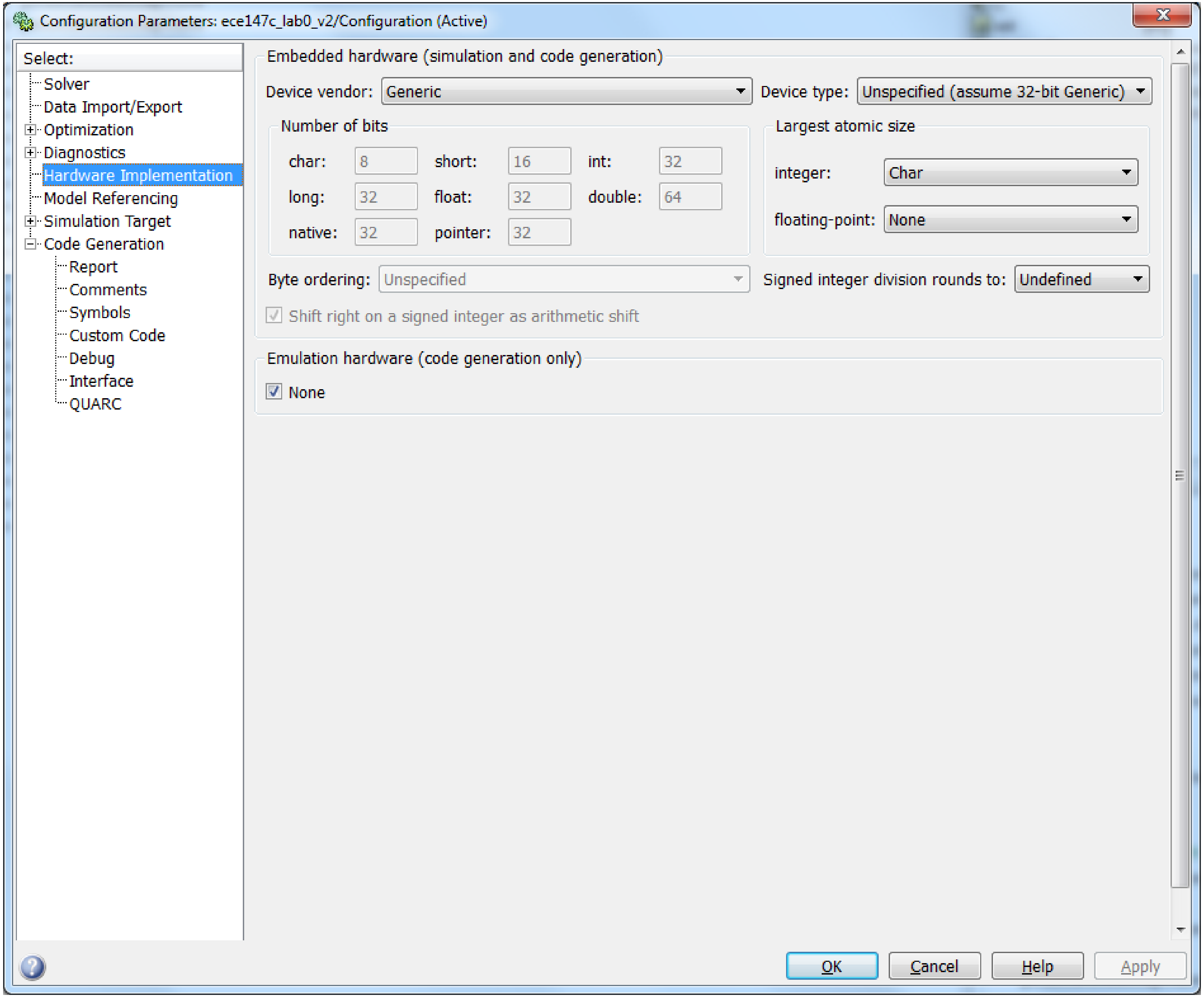
Tasking and sample time options

Periodic sample time constraint: Unconstrained

Tasking mode for periodic sample times: Auto

□ Automatically handle rate transition for data transfer

☑ Higher priority value indicates higher task priority



Configuration Parameters: ece147c_lab0_v2/Configuration (Active)

Embedded hardware (simulation and code generation)

Device vendor: Generic

Device type: Unspecified (assume 32-bit Generic)

Number of bits

char: 8 short: 16 int: 32

long: 32 float: 32 double: 64

native: 32 pointer: 32

Largest atomic size

integer: Char

floating-point: None

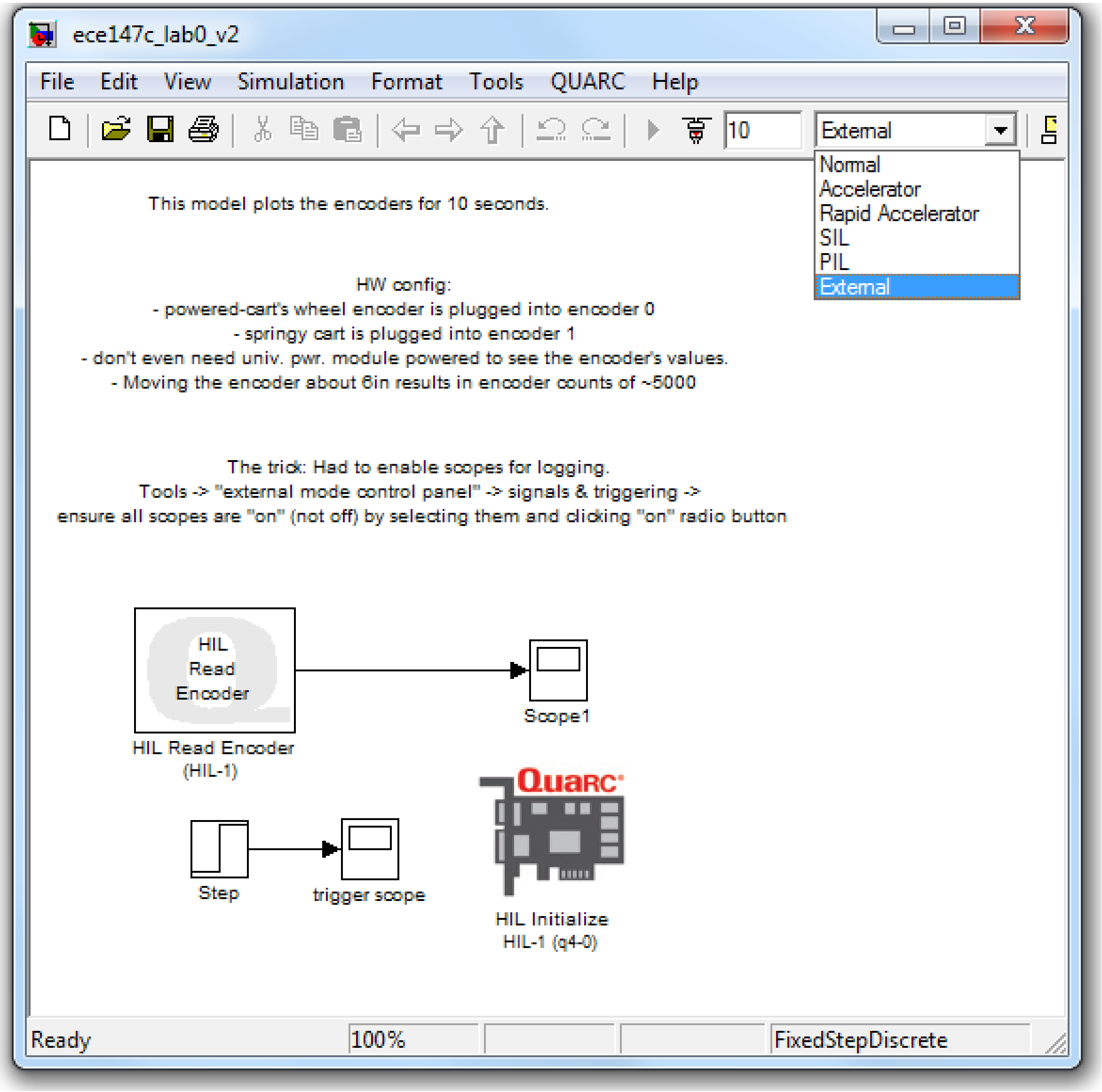
Byte ordering: Unspecified

Signed integer division rounds to: Undefined

☑ Shift right on a signed integer as arithmetic shift

Emulation hardware (code generation only)

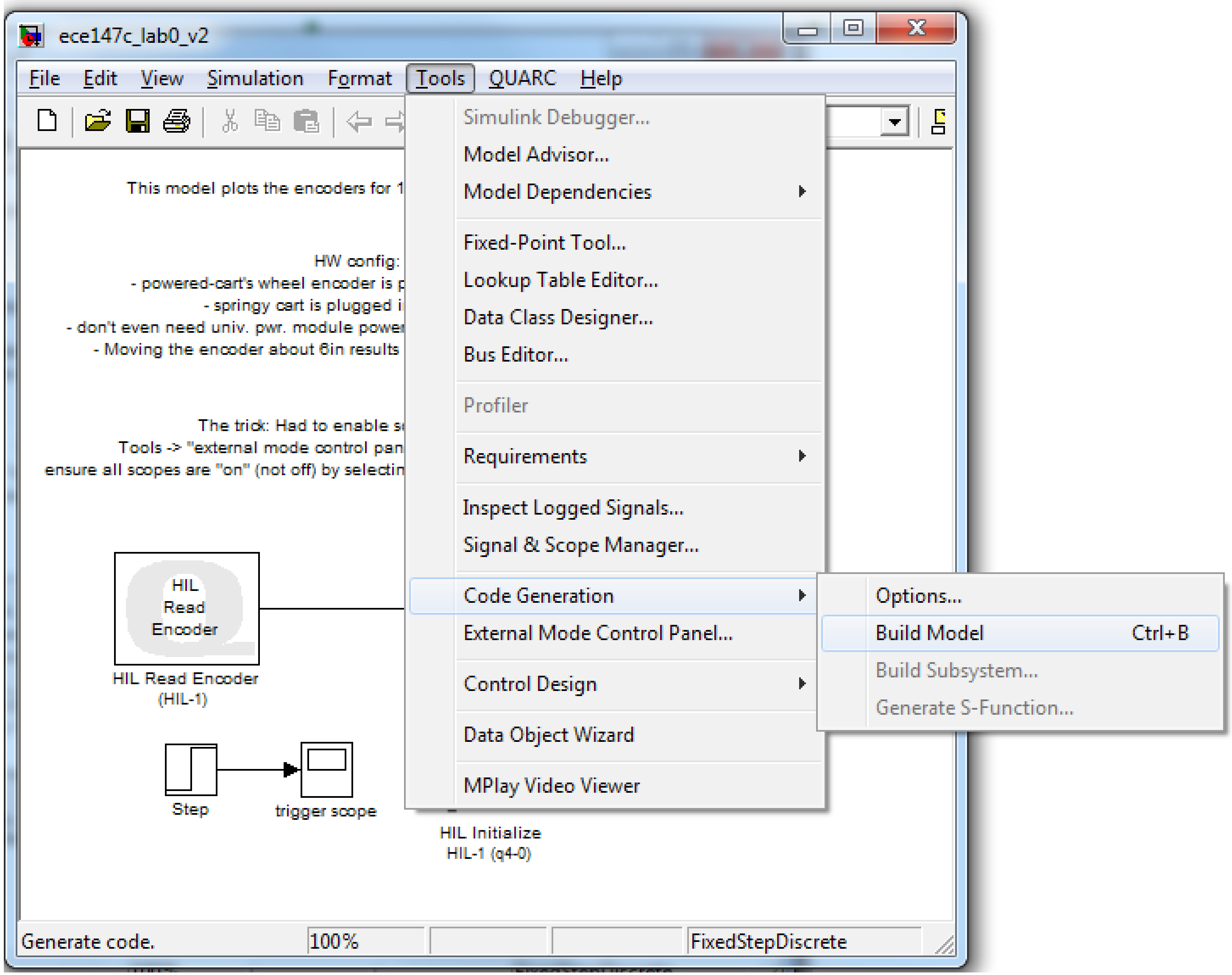
☑ None



This model plots the encoders for 10 seconds.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

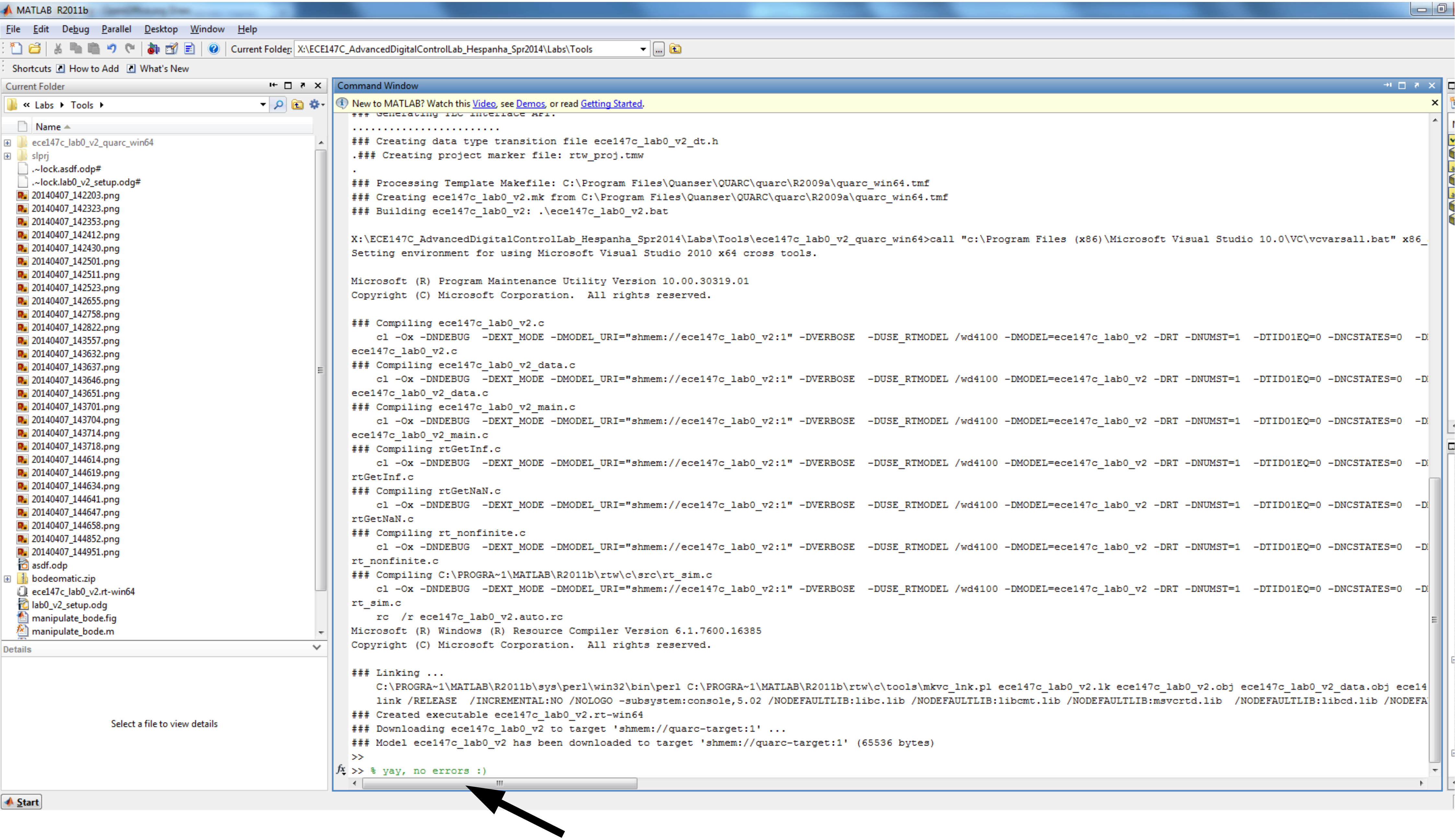


This model plots the encoders for 1

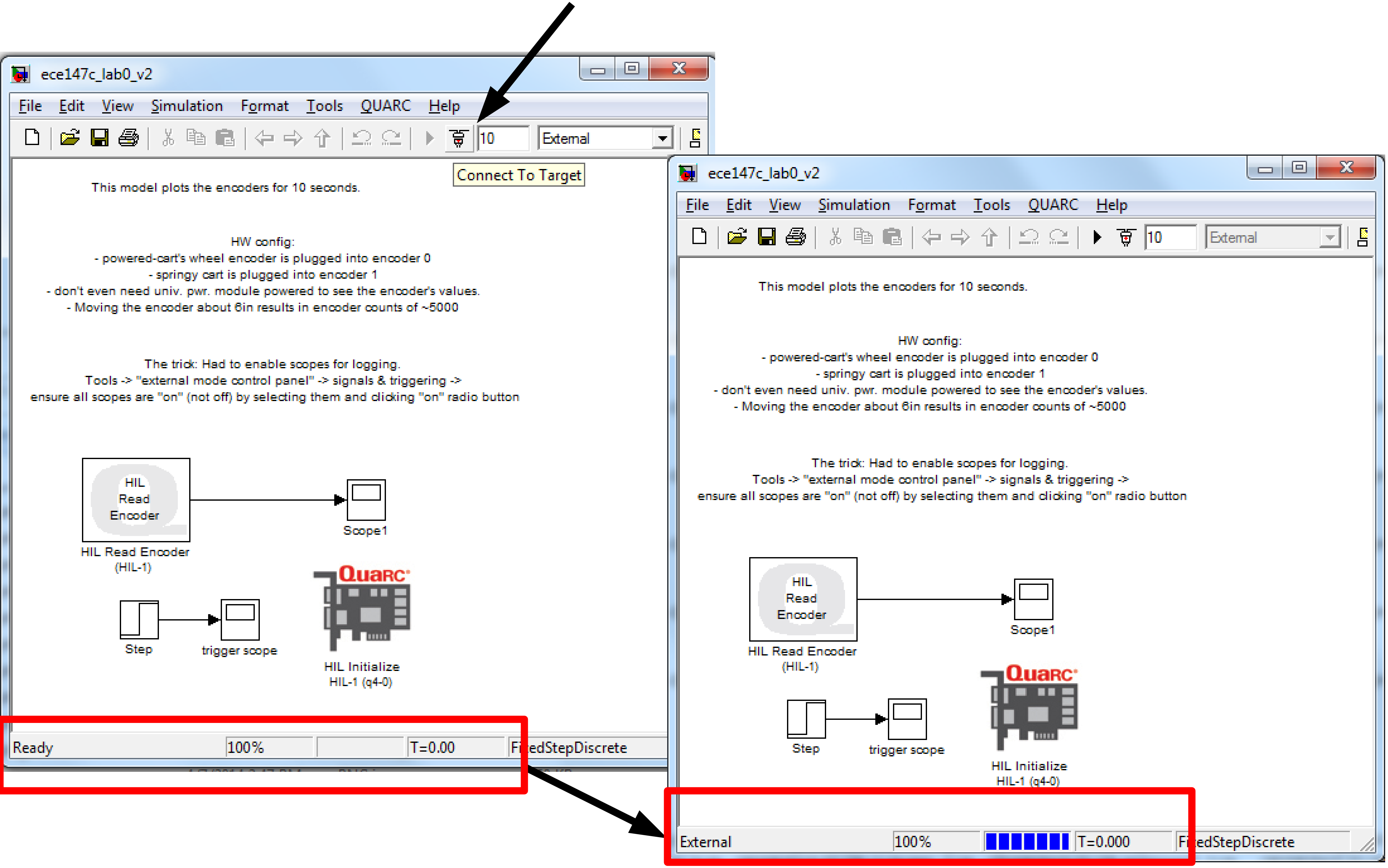
HW config:
- powered-cart's wheel encoder is p
- springy cart is plugged i
- don't even need univ. pwr. module power
- Moving the encoder about 6in results

The trick: Had to enable s
Tools -> "external mode control pan
ensure all scopes are "on" (not off) by selectin

Generate code.



```
### Generating TLC interface API.
.......................
### Creating data type transition file ece147c_lab0_v2_dt.h
.### Creating project marker file: rtw_proj.tmw
.
### Processing Template Makefile: C:\Program Files\Quanser\QUARC\quarc\R2009a\quarc_win64.tmf
### Creating ece147c_lab0_v2.mk from C:\Program Files\Quanser\QUARC\quarc\R2009a\quarc_win64.tmf
### Building ece147c_lab0_v2: .\ece147c_lab0_v2.bat

X:\ECE147C_AdvancedDigitalControlLab_Hespanha_Spr2014\Labs\Tools\ece147c_lab0_v2_quarc_win64>call "c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\vcvarsall.bat" x86_
Setting environment for using Microsoft Visual Studio 2010 x64 cross tools.

Microsoft (R) Program Maintenance Utility Version 10.00.30319.01
Copyright (C) Microsoft Corporation.  All rights reserved.

### Compiling ece147c_lab0_v2.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab0_v2:1" -DVERBOSE  -DUSE_RTMODEL /wd4100 -DMODEL=ece147c_lab0_v2 -DRT -DNUMST=1  -DTID01EQ=0 -DNCSTATES=0  -D
ece147c_lab0_v2.c
### Compiling ece147c_lab0_v2_data.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab0_v2:1" -DVERBOSE  -DUSE_RTMODEL /wd4100 -DMODEL=ece147c_lab0_v2 -DRT -DNUMST=1  -DTID01EQ=0 -DNCSTATES=0  -D
ece147c_lab0_v2_data.c
### Compiling ece147c_lab0_v2_main.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab0_v2:1" -DVERBOSE  -DUSE_RTMODEL /wd4100 -DMODEL=ece147c_lab0_v2 -DRT -DNUMST=1  -DTID01EQ=0 -DNCSTATES=0  -D
ece147c_lab0_v2_main.c
### Compiling rtGetInf.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab0_v2:1" -DVERBOSE  -DUSE_RTMODEL /wd4100 -DMODEL=ece147c_lab0_v2 -DRT -DNUMST=1  -DTID01EQ=0 -DNCSTATES=0  -D
rtGetInf.c
### Compiling rtGetNaN.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab0_v2:1" -DVERBOSE  -DUSE_RTMODEL /wd4100 -DMODEL=ece147c_lab0_v2 -DRT -DNUMST=1  -DTID01EQ=0 -DNCSTATES=0  -D
rtGetNaN.c
### Compiling rt_nonfinite.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab0_v2:1" -DVERBOSE  -DUSE_RTMODEL /wd4100 -DMODEL=ece147c_lab0_v2 -DRT -DNUMST=1  -DTID01EQ=0 -DNCSTATES=0  -D
rt_nonfinite.c
### Compiling C:\PROGRA~1\MATLAB\R2011b\rtw\c\src\rt_sim.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab0_v2:1" -DVERBOSE  -DUSE_RTMODEL /wd4100 -DMODEL=ece147c_lab0_v2 -DRT -DNUMST=1  -DTID01EQ=0 -DNCSTATES=0  -D
rt_sim.c
    rc  /r ece147c_lab0_v2.auto.rc
Microsoft (R) Windows (R) Resource Compiler Version 6.1.7600.16385
Copyright (C) Microsoft Corporation.  All rights reserved.

### Linking ...
    C:\PROGRA~1\MATLAB\R2011b\sys\perl\win32\bin\perl C:\PROGRA~1\MATLAB\R2011b\rtw\c\tools\mkvc_lnk.pl ece147c_lab0_v2.lk ece147c_lab0_v2.obj ece147c_lab0_v2_data.obj ece14
    link /RELEASE  /INCREMENTAL:NO /NOLOGO -subsystem:console,5.02 /NODEFAULTLIB:libc.lib /NODEFAULTLIB:libcmt.lib /NODEFAULTLIB:msvcrtd.lib  /NODEFAULTLIB:libcd.lib /NODEFA
### Created executable ece147c_lab0_v2.rt-win64
### Downloading ece147c_lab0_v2 to target 'shmem://quarc-target:1' ...
### Model ece147c_lab0_v2 has been downloaded to target 'shmem://quarc-target:1' (65536 bytes)
>>
>> % yay, no errors :)
```



This model plots the encoders for 10 seconds.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

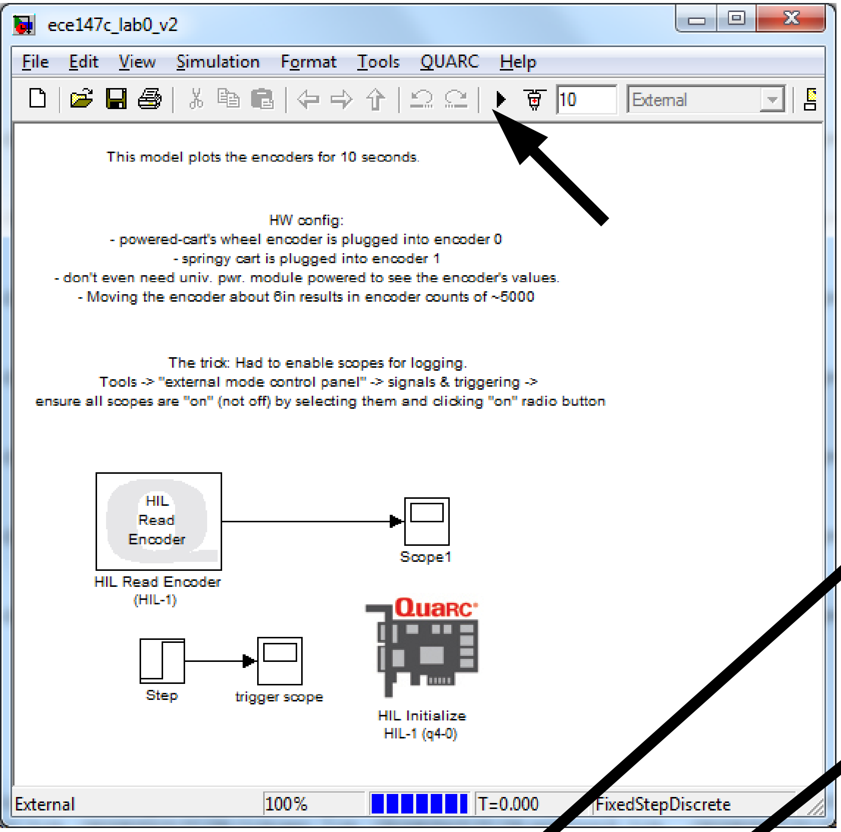
```
Ready                100%          T=0.00     Fi edStepDiscrete
```

This model plots the encoders for 10 seconds.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

```
External             100%  ███████ T=0.000    Fi edStepDiscrete
```



This model plots the encoders for 10 seconds.

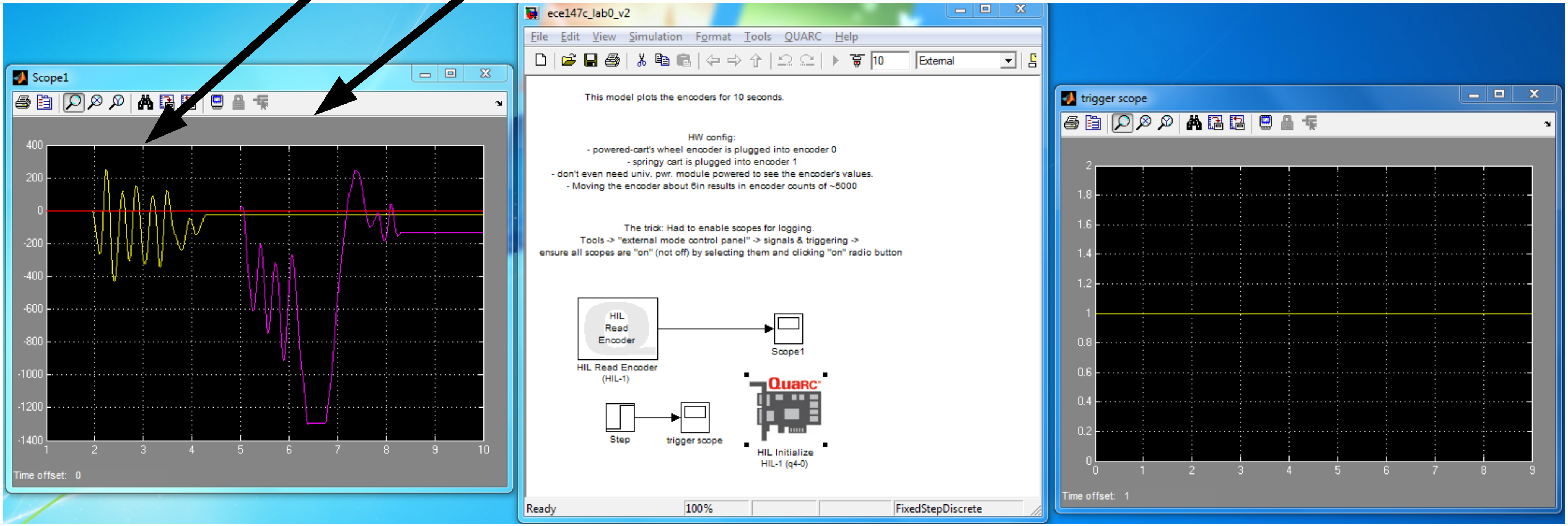
HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

Wiggle motor-cart's wheel encoder

Wiggle springy-cart's wheel encoder

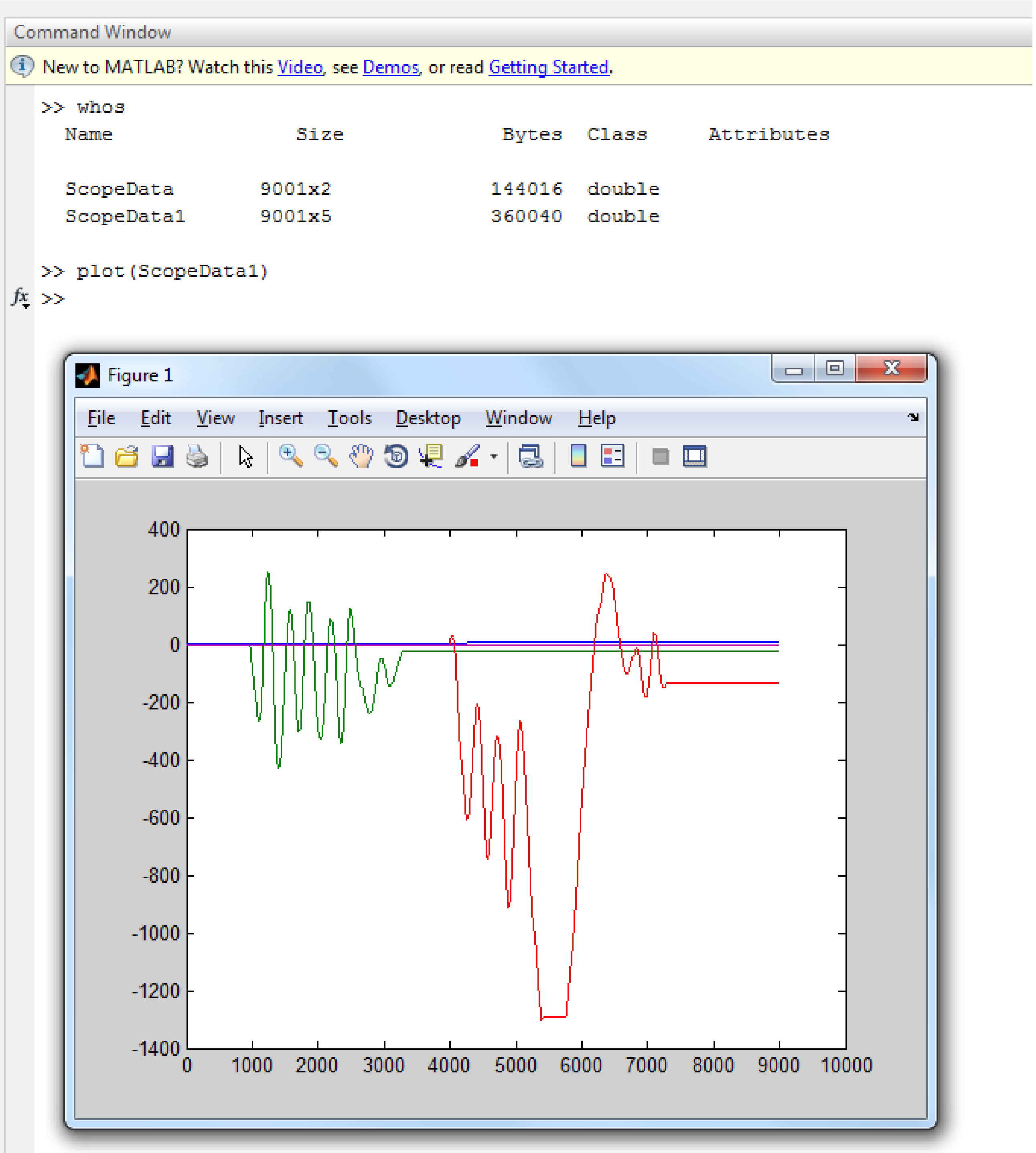
Note: doesn't show 3 encoders??



This model plots the encoders for 10 seconds.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

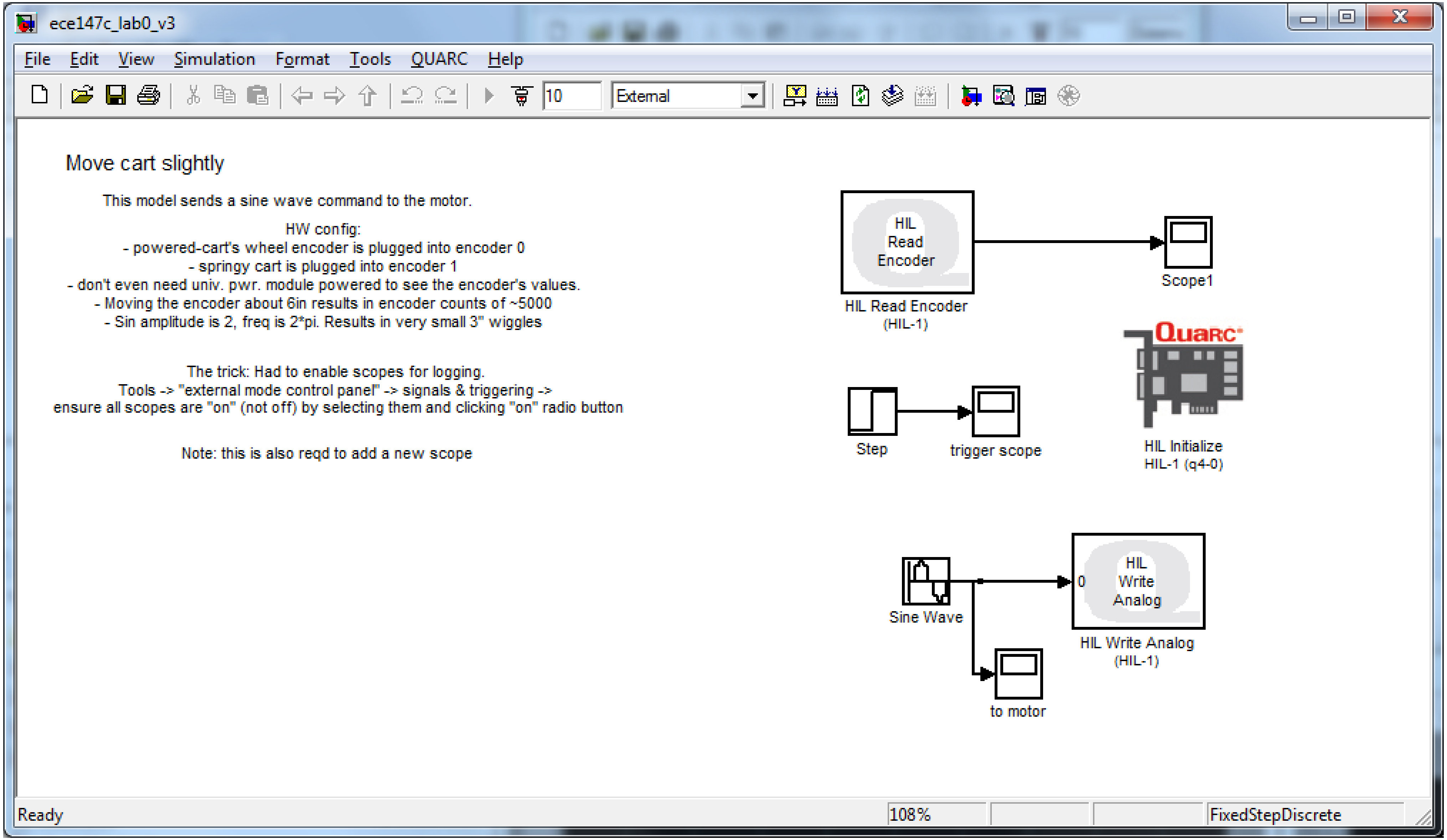


Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> whos
  Name                Size              Bytes  Class     Attributes

  ScopeData        9001x2              144016  double              
  ScopeData1       9001x5              360040  double              

>> plot(ScopeData1)
>> 
```

Next: move motor



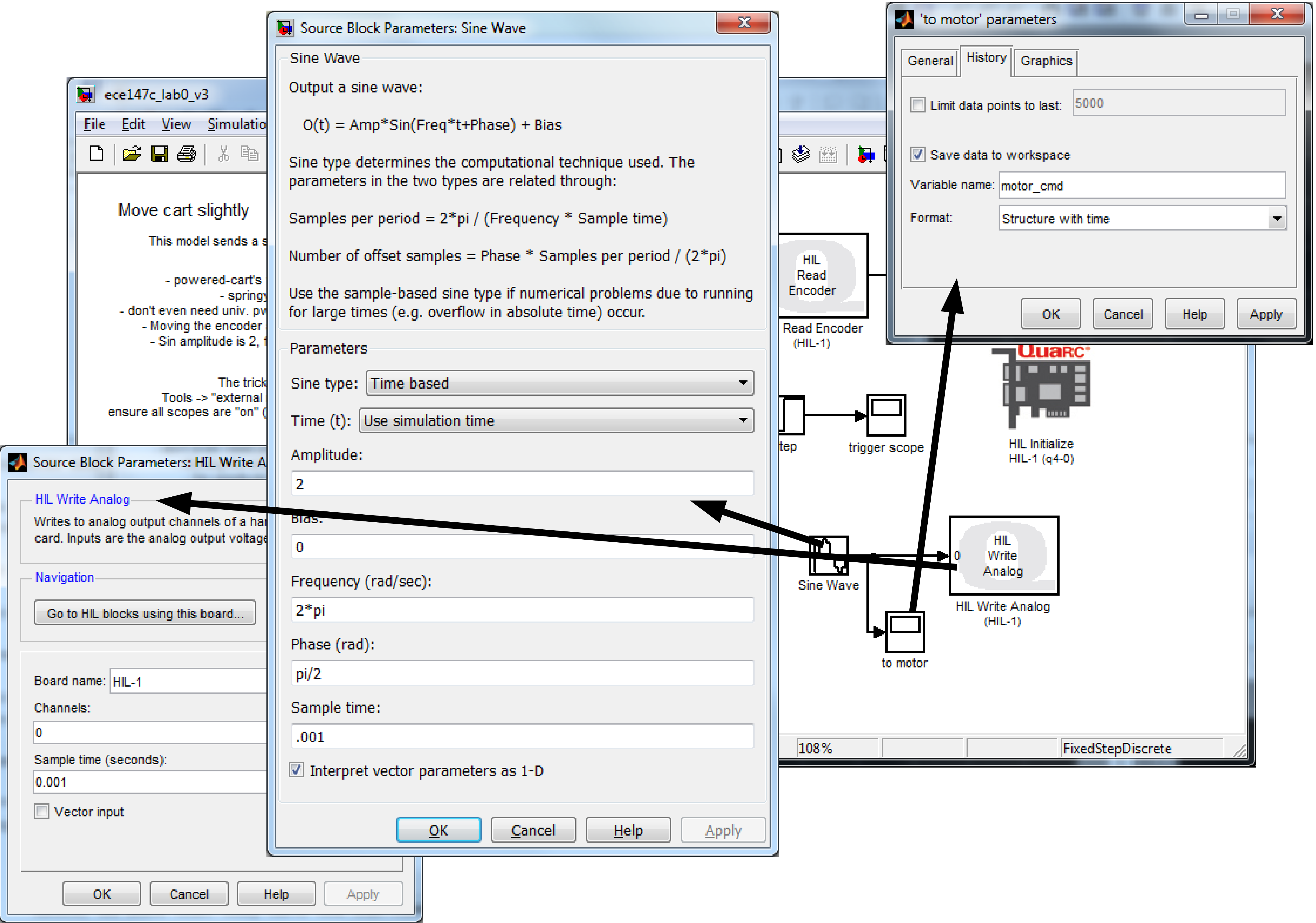
Move cart slightly

This model sends a sine wave command to the motor.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000
- Sin amplitude is 2, freq is 2*pi. Results in very small 3" wiggles

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

Note: this is also reqd to add a new scope



ece147c_lab0_v3

Move cart slightly

This model sends a s

- powered-cart's
- spring
- don't even need univ. pw
- Moving the encoder
- Sin amplitude is 2,

The trick
Tools -> "external
ensure all scopes are "on" (

Source Block Parameters: Sine Wave

Sine Wave

Output a sine wave:

O(t) = Amp*Sin(Freq*t+Phase) + Bias

Sine type determines the computational technique used. The parameters in the two types are related through:

Samples per period = 2*pi / (Frequency * Sample time)

Number of offset samples = Phase * Samples per period / (2*pi)

Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.

Parameters

Sine type: Time based

Time (t): Use simulation time

Amplitude: 2

Bias: 0

Frequency (rad/sec): 2*pi

Phase (rad): pi/2

Sample time: .001

☑ Interpret vector parameters as 1-D

'to motor' parameters

□ Limit data points to last: 5000

☑ Save data to workspace

Variable name: motor_cmd

Format: Structure with time

Source Block Parameters: HIL Write A

HIL Write Analog

Writes to analog output channels of a har card. Inputs are the analog output voltage

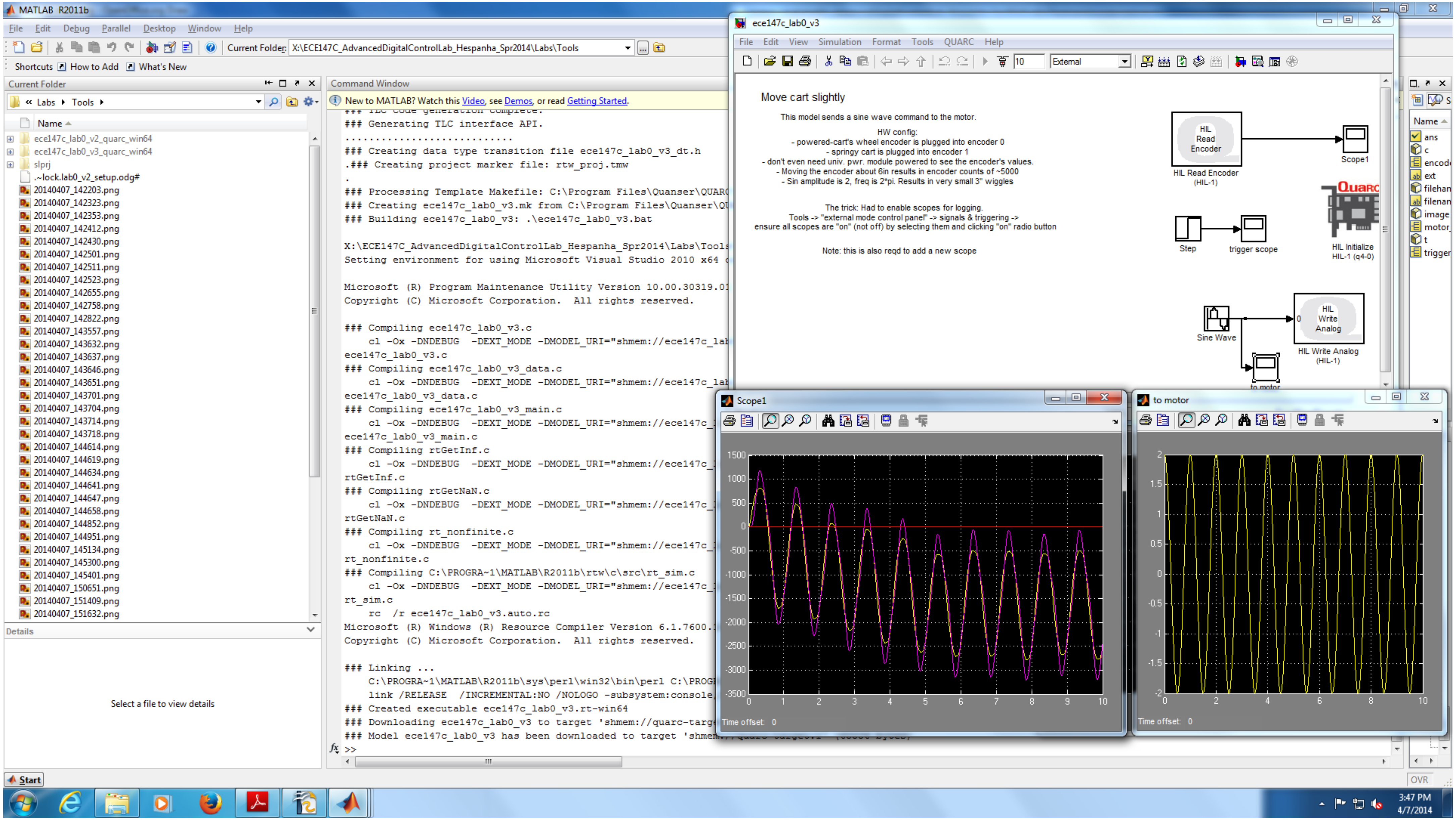
Navigation

Board name: HIL-1

Channels: 0

Sample time (seconds): 0.001

□ Vector input



```
ece147c_lab0_v2_quarc_win64
ece147c_lab0_v3_quarc_win64
slprj
.~lock.lab0_v2_setup.odg#
20140407_142203.png
20140407_142323.png
20140407_142353.png
20140407_142412.png
20140407_142430.png
20140407_142501.png
20140407_142511.png
20140407_142523.png
20140407_142655.png
20140407_142758.png
20140407_142822.png
20140407_143557.png
20140407_143632.png
20140407_143637.png
20140407_143646.png
20140407_143651.png
20140407_143701.png
20140407_143704.png
20140407_143714.png
20140407_143718.png
20140407_144614.png
20140407_144619.png
20140407_144634.png
20140407_144641.png
20140407_144647.png
20140407_144658.png
20140407_144852.png
20140407_144951.png
20140407_145134.png
20140407_145300.png
20140407_145401.png
20140407_150651.png
20140407_151409.png
20140407_151632.png
```

```
### TLC code generation complete.
### Generating TLC interface API.
.............................
### Creating data type transition file ece147c_lab0_v3_dt.h
.### Creating project marker file: rtw_proj.tmw
.
### Processing Template Makefile: C:\Program Files\Quanser\QUARC
### Creating ece147c_lab0_v3.mk from C:\Program Files\Quanser\QU
### Building ece147c_lab0_v3: .\ece147c_lab0_v3.bat

X:\ECE147C_AdvancedDigitalControlLab_Hespanha_Spr2014\Labs\Tools
Setting environment for using Microsoft Visual Studio 2010 x64 c

Microsoft (R) Program Maintenance Utility Version 10.00.30319.01
Copyright (C) Microsoft Corporation.  All rights reserved.

### Compiling ece147c_lab0_v3.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab
ece147c_lab0_v3.c
### Compiling ece147c_lab0_v3_data.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_lab
ece147c_lab0_v3_data.c
### Compiling ece147c_lab0_v3_main.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_
ece147c_lab0_v3_main.c
### Compiling rtGetInf.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_
rtGetInf.c
### Compiling rtGetNaN.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_
rtGetNaN.c
### Compiling rt_nonfinite.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_
rt_nonfinite.c
### Compiling C:\PROGRA~1\MATLAB\R2011b\rtw\c\src\rt_sim.c
    cl -Ox -DNDEBUG  -DEXT_MODE -DMODEL_URI="shmem://ece147c_
rt_sim.c
    rc  /r ece147c_lab0_v3.auto.rc
Microsoft (R) Windows (R) Resource Compiler Version 6.1.7600.
Copyright (C) Microsoft Corporation.  All rights reserved.

### Linking ...
    C:\PROGRA~1\MATLAB\R2011b\sys\perl\win32\bin\perl C:\PROG
    link /RELEASE  /INCREMENTAL:NO /NOLOGO -subsystem:console
### Created executable ece147c_lab0_v3.rt-win64
### Downloading ece147c_lab0_v3 to target 'shmem://quarc-targ
### Model ece147c_lab0_v3 has been downloaded to target 'shmem
>>
```

Move cart slightly

This model sends a sine wave command to the motor.

HW config:
- powered-cart's wheel encoder is plugged into encoder 0
- springy cart is plugged into encoder 1
- don't even need univ. pwr. module powered to see the encoder's values.
- Moving the encoder about 6in results in encoder counts of ~5000
- Sin amplitude is 2, freq is 2*pi. Results in very small 3" wiggles

The trick: Had to enable scopes for logging.
Tools -> "external mode control panel" -> signals & triggering ->
ensure all scopes are "on" (not off) by selecting them and clicking "on" radio button

Note: this is also reqd to add a new scope

Troubleshooting:
- is Universal power module on?
- Is analog output cable connecting analog output 0 to UPM's "From D/A"?
- Is cable from UPM's "To Load" to motor-cart's motor plug?
- Is simulink "HIL Write Analog" block set to same channel as the board's analog output?