

UCSB CONTROLS LAB TA MANUAL

BY MAX CRISAFULLI

SEPTEMBER 2024



THIS DOCUMENT LIVES AT
<https://github.com/MaxCrisafulli/UCSB-ControlsLab-Manual-2024>

TA MANUAL FOR THE UCSB ECE CONTROLS LAB

This document is to act as a guide for TAs in setting up the Quanser inverted pendulum cart hardware in the UCSB ECE Controls Lab (Harold Frank Hall 3120A). This guide is an update of the existing manual, located at:

<https://github.com/justinpearson/UCSB-Quanser-Inverted-Pendulum-Lab-Manual>

This updated guide will live (and hopefully be maintained) at

<https://github.com/MaxCrisafulli/UCSB-ControlsLab-Manual-2024>

HARDWARE IN THE LAB

The ECE147 labs make use of the Quanser ‘Linear Servo Base Unit with Inverted Pendulum’. This manual will detail the setup of just the cart system, with motor voltage as input and linear position as output. There are 3 main components of the hardware setup, with each of which being explained below.

Q4 DAQ BOARD

The SIMULINK model that interfaces with the hardware interacts with the world via the Q4 HIL control card and DAQ (Data Acquisition Board). The control card is inside the lab PCs connected via PCIe and interfaces with the DAQ board via the two gray ribbon cables. Ensure that these are connected properly and that the red Status LED is lit, indicating that the board is powered.

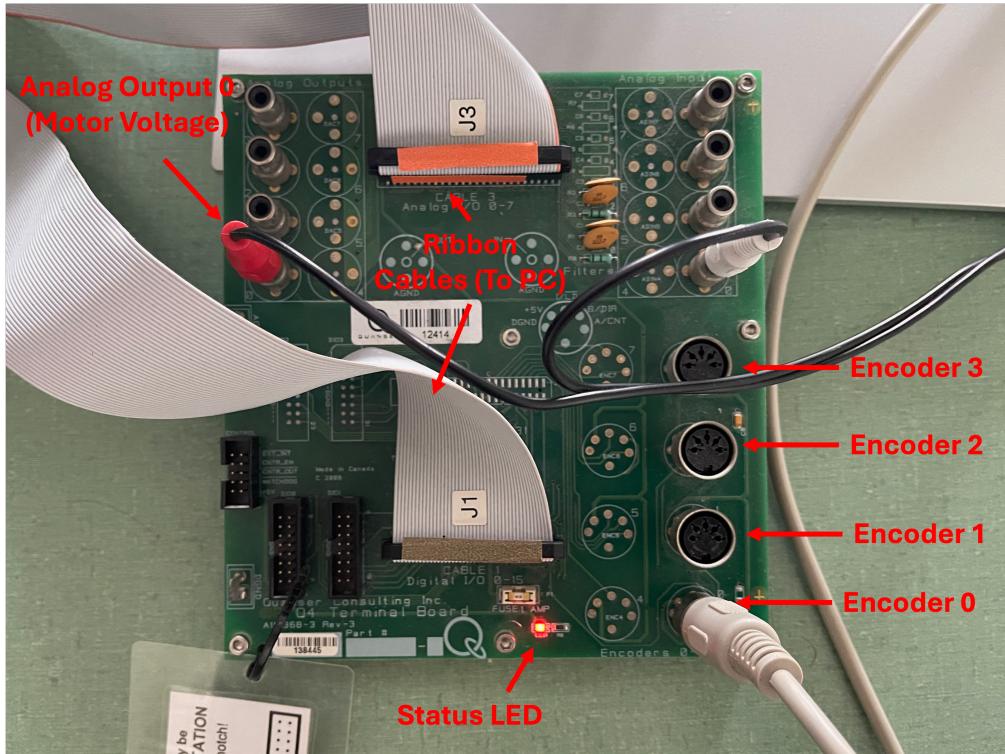


Figure 1: Quanser Q4 Terminal DAQ Board

The DAQ board uses the four encoder ports to read signals from the hardware. Generally, Encoder 0 is used for the linear position encoder. In the top left corner of the board are the Analog Outputs, Analog Output 0 is used to drive the amplifier box which ultimately drives the cart motor.

VOLTPAQ-X1 AMPLIFIER

The Quanser VoltPAQ-X1 Amplifier is powered from a standard wall socket, and turned on via a switch at the back of the box. Make sure that the green status LED in the center of the amplifier diagram is lit, indicating that the amplifier is powered on. Also check that the Amplifier Gain switch is set to 1x.

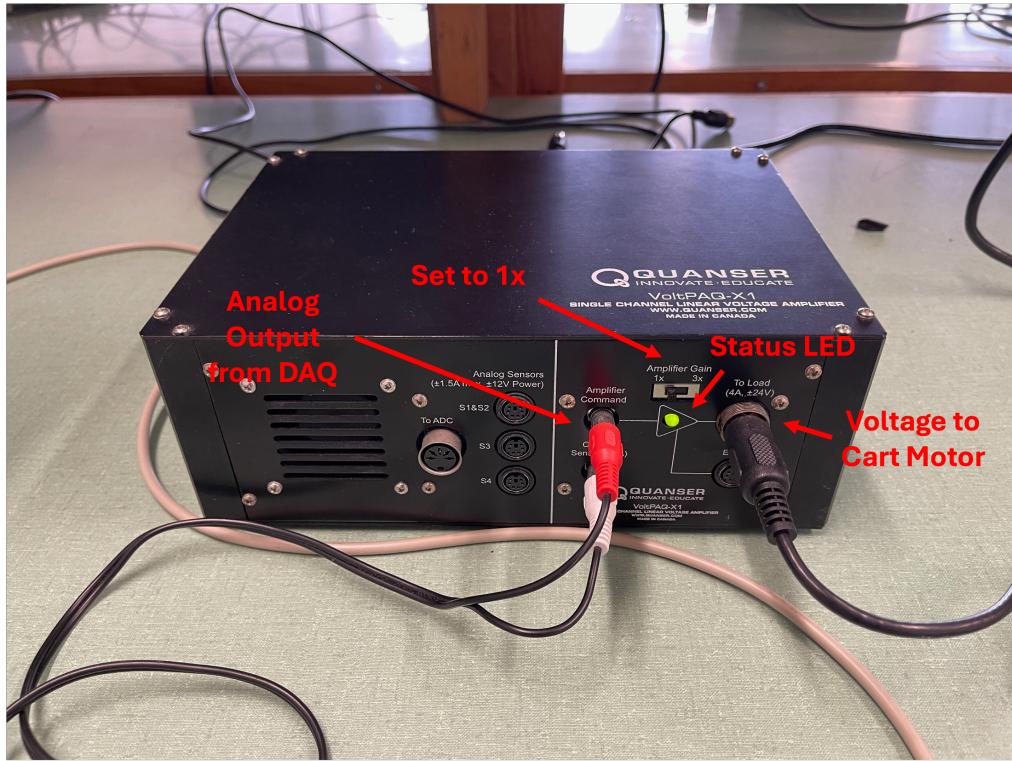


Figure 2: Quanser VoltPAQ-X1 Linear Voltage Amplifier

The red AV cable coming from Analog Output 0 on the DAQ board should be connected to the Amplifier Command (input) port. The white AV cable should be connected to Current Sense on the amplifier and Analog Input 0 on the DAQ board, although this is not necessary.

The output of the amplifier (To Load) is what drives the motor onboard the cart. Check that all the cable connections are correct and that the cable going to the motor has enough slack to allow the cart to move along the track.

INVERTED PENDULUM CART SYSTEM

The cart system has 2 encoder outputs and 1 analog voltage input. The positions of these ports may vary between the carts but in general they are as depicted in Figure 3. You should check what each of the ports connect to before plugging anything in.

The cart controls its position along the track with the smaller motor drive wheel, which is driven by the motor voltage coming from the amplifier. The larger position encoder wheel measures the relative position along the track from the time that the program was initialized in ‘ticks’. The conversion ratio from ticks to inches is approximately 5000 ticks per 6 inches.

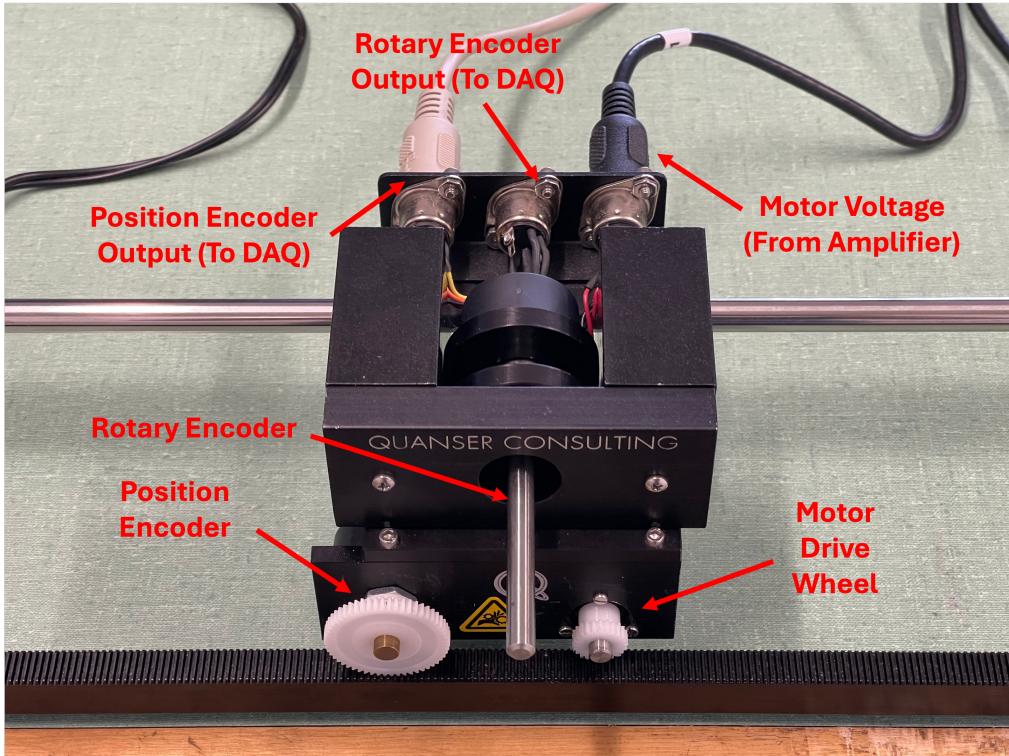


Figure 3: Quanser Linear Servo Base Unit (The ‘Cart’)

INSTRUCTIONS

The following slides will detail setting up a minimum working SIMULINK model that can pass a voltage to the motor and read position from the cart encoder.

Ensure that you are using MATLAB R2023a and that the QuaRC package (Quanser Realtime Control) is installed. Run the command `mex -setup` and `mex -setup C++` in the command line to make sure that MATLAB can interface with a compiler, which should be Microsoft Visual C++ Redistributable 2022.

The SIMULINK model `hardware_test.mdl` in the Github repo should be configured in such a way that it can be run (assuming all the hardware is setup correctly). It can also be used as a template for other models using the same Hardware.

Configuring the model to work with DAQ

The figure illustrates the configuration process for a hardware-in-the-loop (HIL) test setup using QUARC and Simulink.

Top Left: Source Block Parameters: HIL Initialize dialog box. It shows the configuration for a Quanser Q4-PCI board. The "Board name" is set to "HIL-1" and the "Board type" is set to "q4". A red box highlights these settings, and a red arrow points from the "Defaults" button at the bottom right to the "HIL Configuration" window in the center.

Top Right: SIMULINK interface showing the "hardware_test" model. The "QUARC" tab is selected. The "HIL Configuration" block is visible in the workspace. A red arrow points from the "HIL Configuration" block to the "Motor Cart Plant" block in the model diagram.

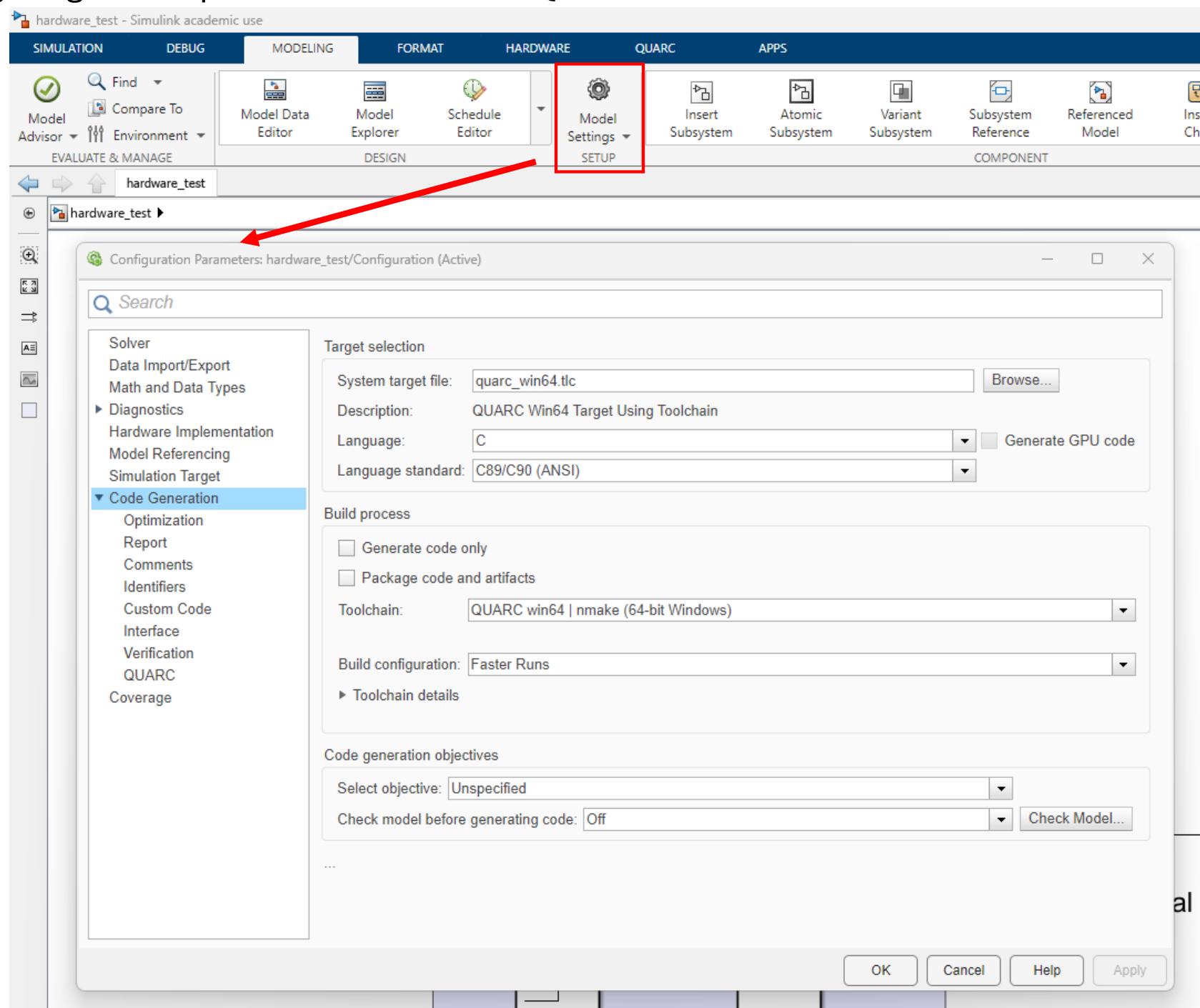
Bottom Left: QUARC Targets/Data Acquisition/Generic/Configuration library browser. The "Configuration" block is highlighted in the tree view. A red arrow points from the "HIL Configuration" block in the center to the "Configuration" block in the library browser.

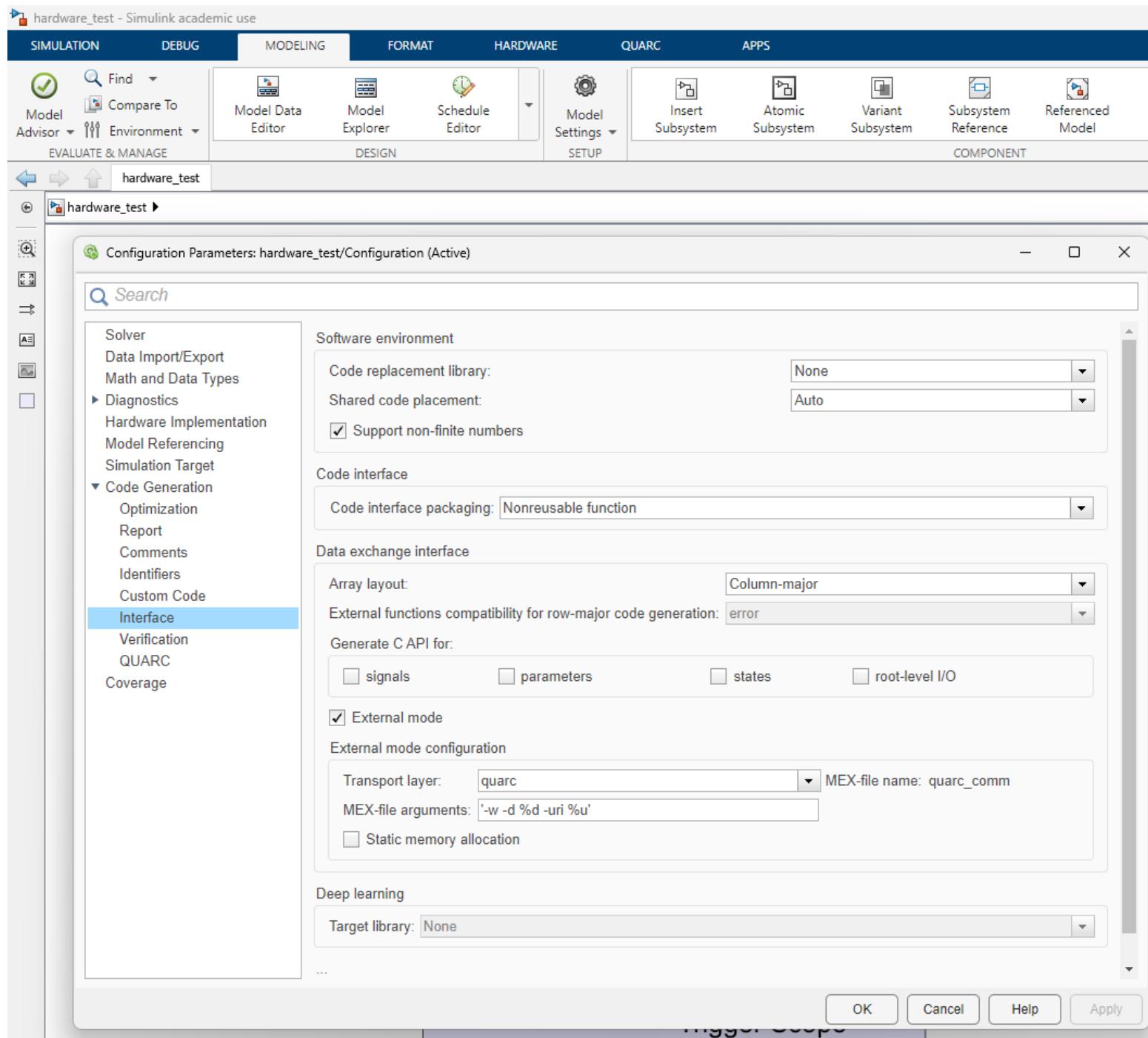
Bottom Center: The "hardware_test.mdl" model diagram. It shows a "Test Signal" block connected to a "Motor Voltage" block. The "Motor Voltage" block has two outputs: one goes to a "Position" scope, and the other goes to a "Motor Cart Plant" block. The "Motor Cart Plant" block contains a "ScopeTemplate" block and a "Trigger Scope" block.

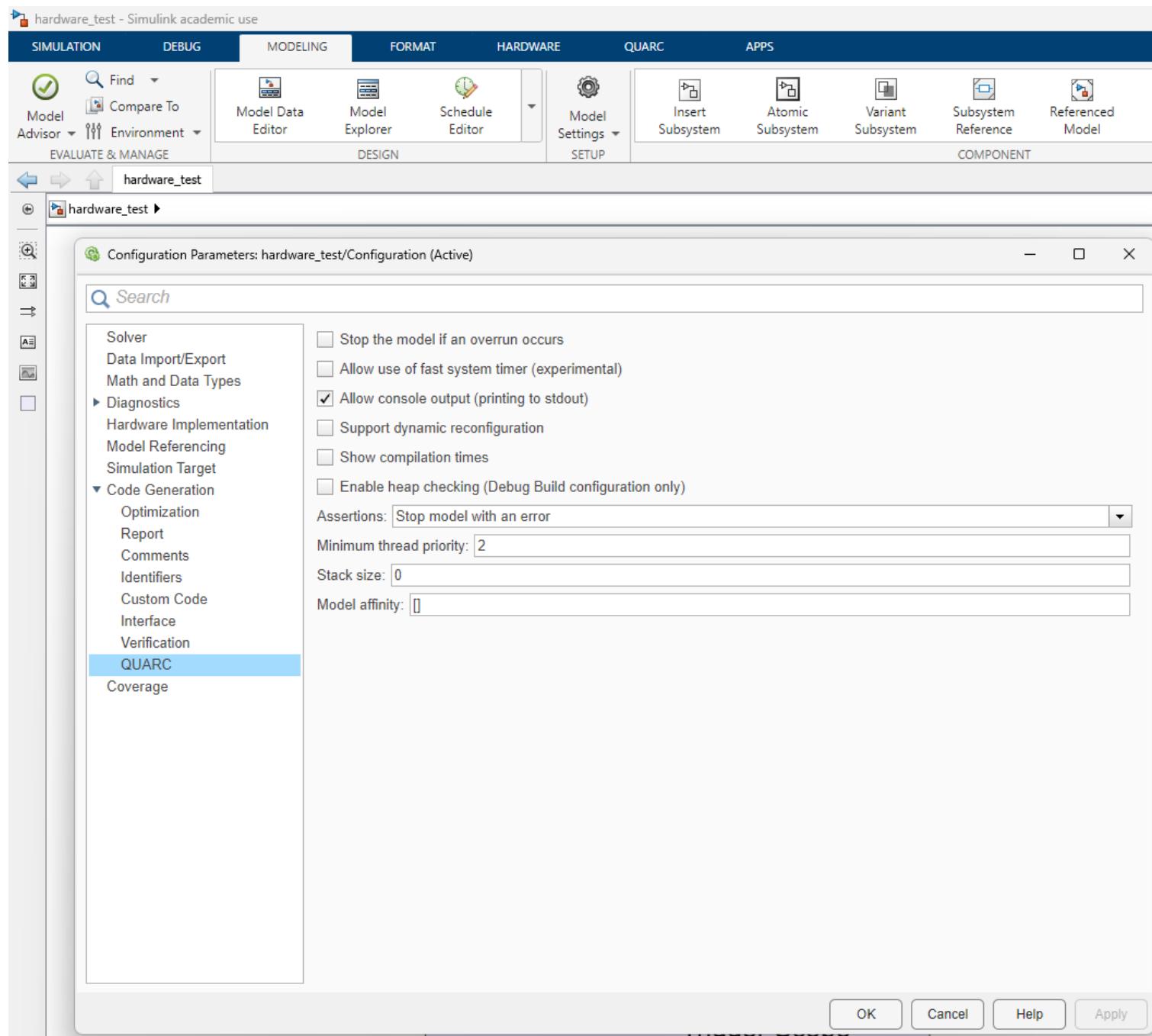
Text:

Set the HIL Initialize block to have board name HIL-1 and Board type q4. Click defaults to set up

Configuring model parameters for HIL & Quarc







Configuration Parameters: hardware_test/Configuration (Active)

Search

Solver

- Data Import/Export
- Math and Data Types
- Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target
- Code Generation
 - Optimization
 - Report
 - Comments
 - Identifiers
 - Custom Code
 - Interface
 - Verification
 - QUARC
 - Coverage

Simulation time

Start time: 0.0 Stop time: 10

Solver selection

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

Solver details

Fixed-step size (fundamental sample time): 0.001

Tasking and sample time options

Periodic sample time constraint: Unconstrained

Treat each discrete rate as a separate task

Allow tasks to execute concurrently on target

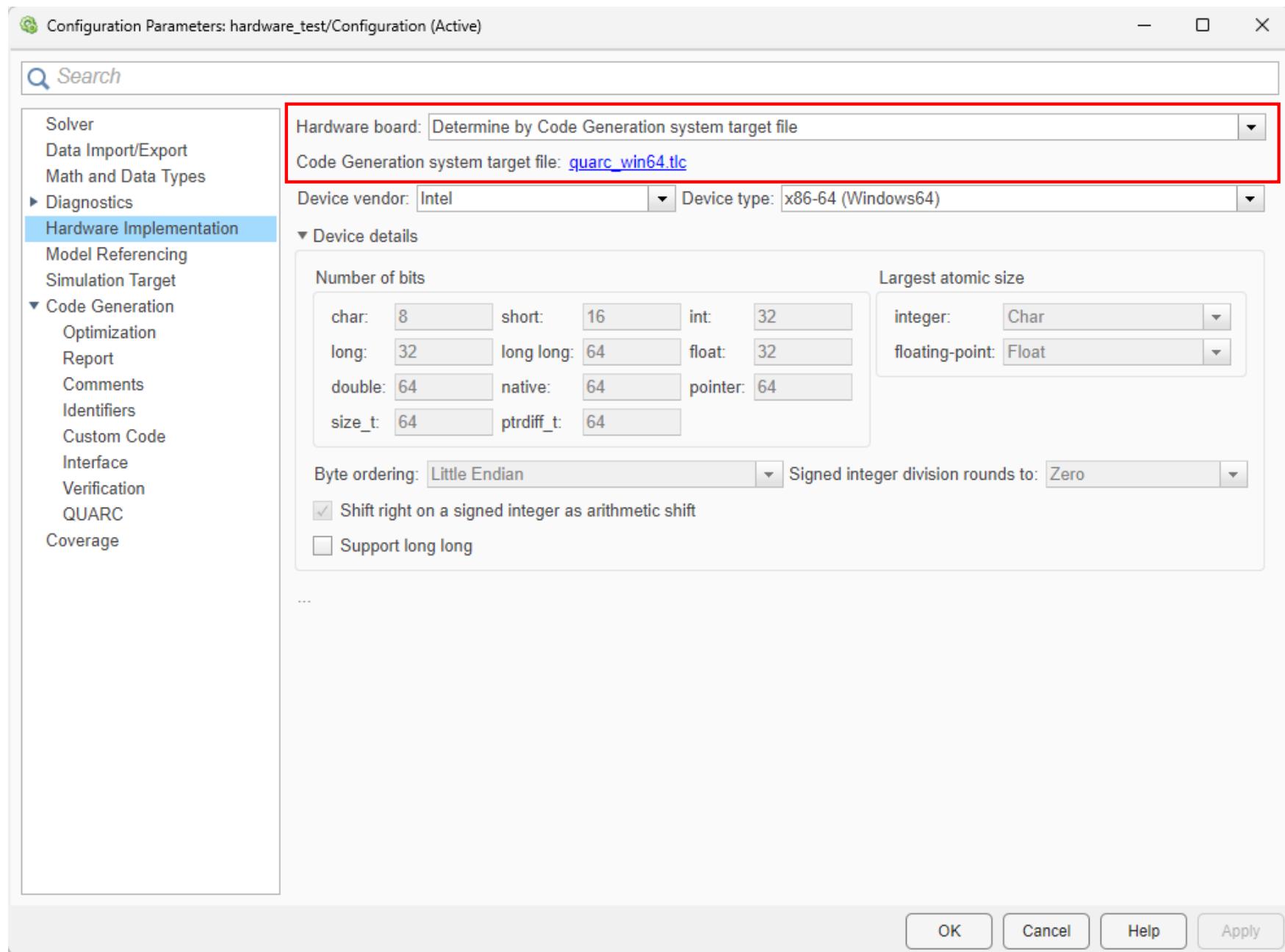
Automatically handle rate transition for data transfer

Allow multiple tasks to access inputs and outputs

Higher priority value indicates higher task priority

Important to have these set correctly
 $T_s = 0.001$, if you change it here change it everywhere...

OK Cancel Help Apply



Configuring scopes for data logging:

The screenshot shows the MATLAB/Simulink environment. On the left, the 'HIL Configuration' window displays a block diagram. It includes a QUARC hardware icon, a 'HIL-1 (q4-0)' block, a step input signal, a 'Trigger Scope' block, and a 'ScopeTemplate' block. A red arrow points from the 'ScopeTemplate' block towards the 'View' menu in the top navigation bar. The 'View' menu is open, showing options like 'Layout ...', 'Configuration Properties ...', 'Style ...', 'Legend', 'Bring All Scope Windows Forward' (with a keyboard shortcut 'Ctrl+F'), 'Toolbar' (unchecked), 'Status Bar' (unchecked), and 'Highlight Simulink Block' (with a keyboard shortcut 'Ctrl+L')). Another red arrow points from the 'Configuration Properties ...' option in the 'View' menu down to the 'Configuration Properties: ScopeTemplate' dialog box. This dialog box has tabs for 'Main', 'Time', 'Display', and 'Logging'. The 'Logging' tab is selected, showing settings: 'Decimation:' checked with value '1', 'Log data to workspace' checked, 'Variable name:' set to 'ScopeTemplate', and 'Save format:' set to 'Array'. A third red arrow points from the 'ScopeTemplate' block in the main window down to the text 'ScopeTemplate(:,1) = `time`' and 'ScopeTemplate(:,2) = `data`' located at the bottom left. To the right of this text is a large explanatory note: '`Variable name` is what will appear in the workspace after the run is complete'. At the very bottom of the dialog box are buttons for '?', 'OK', 'Cancel', and 'Apply'.

ScopeTemplate(:,1) = `time`
ScopeTemplate(:,2) = `data`

Saved as an array in the workspace

Configuration Properties ...

Main Time Display Logging

Limit data points to last: 5000

Decimation: 1

Log data to workspace

Variable name: ScopeTemplate

Save format: Array

OK Cancel Apply

ScopeTemplate

HIL Configuration

File Tools View Simulation Help

Layout ...

Configuration Properties ...

Style ...

Legend

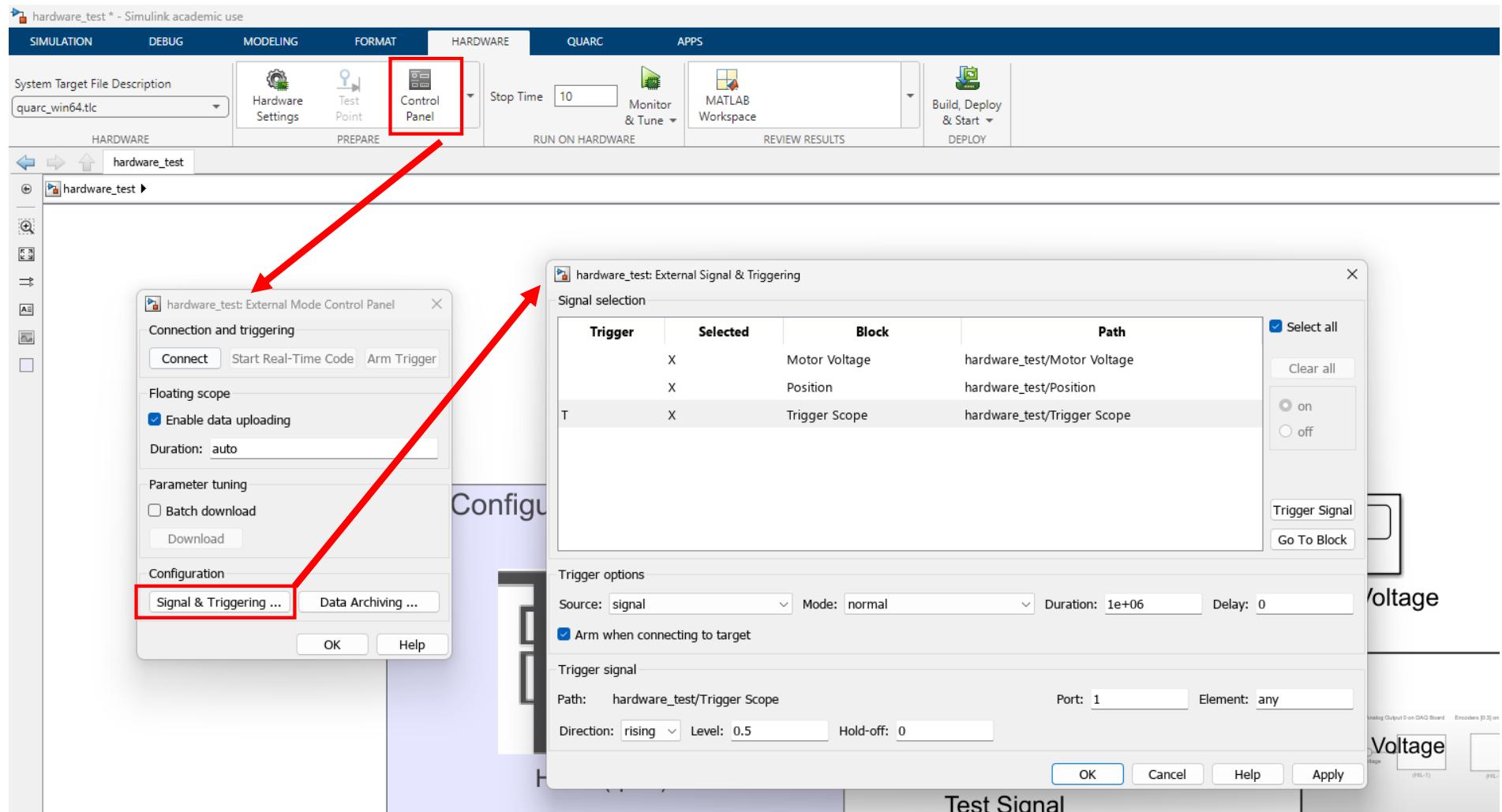
Bring All Scope Windows Forward Ctrl+F

Toolbar

Status Bar

Highlight Simulink Block Ctrl+L

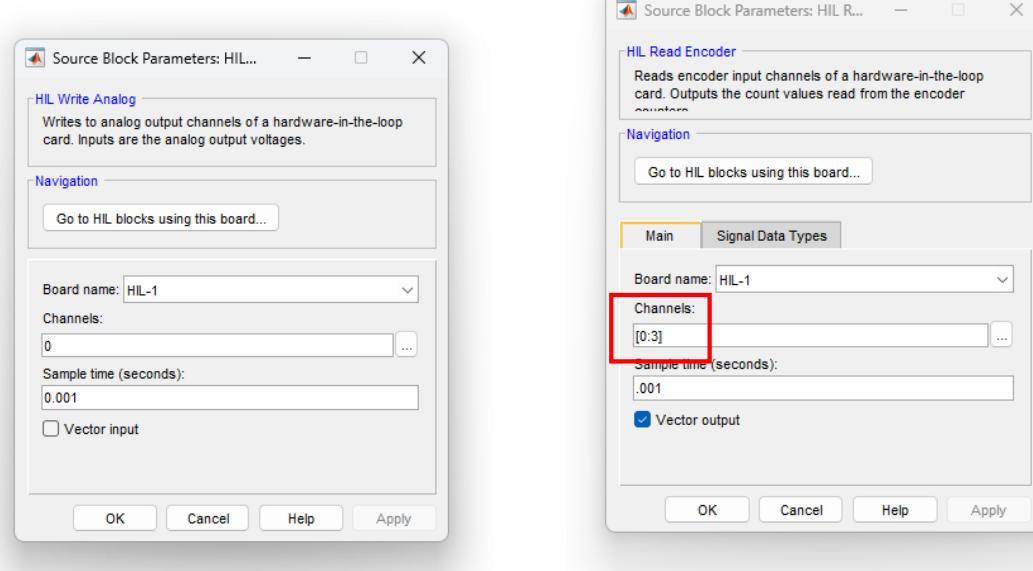
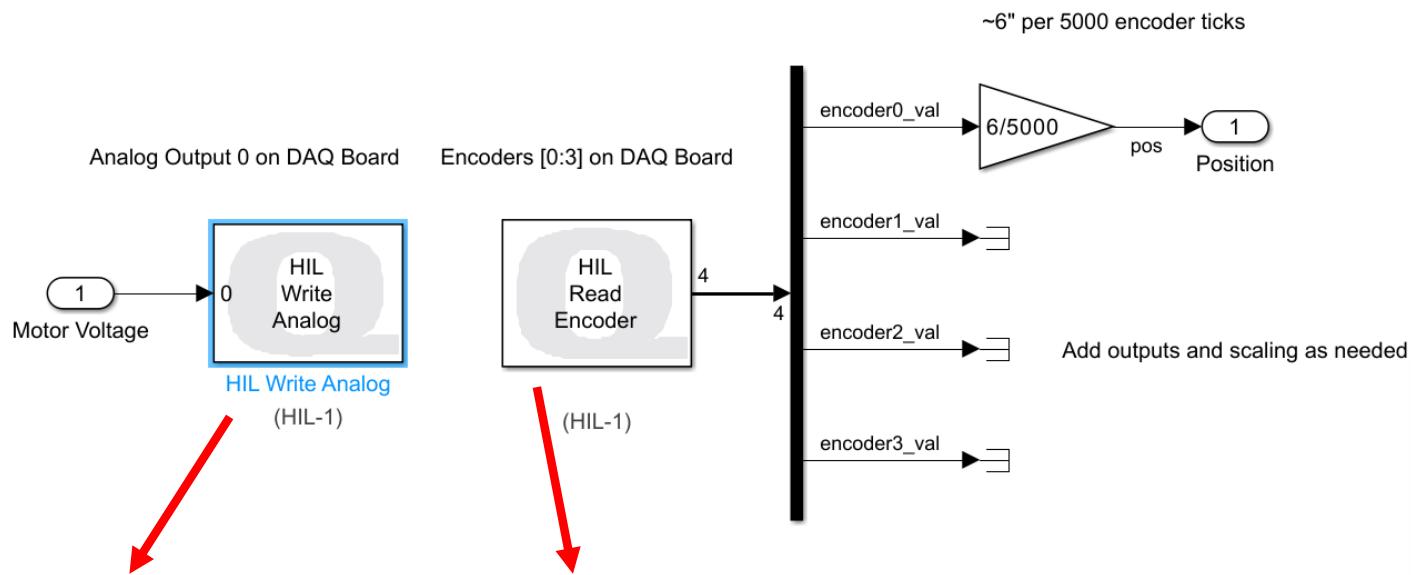
Configuring triggering and model data logging:



X under Selected means the signal will be logged

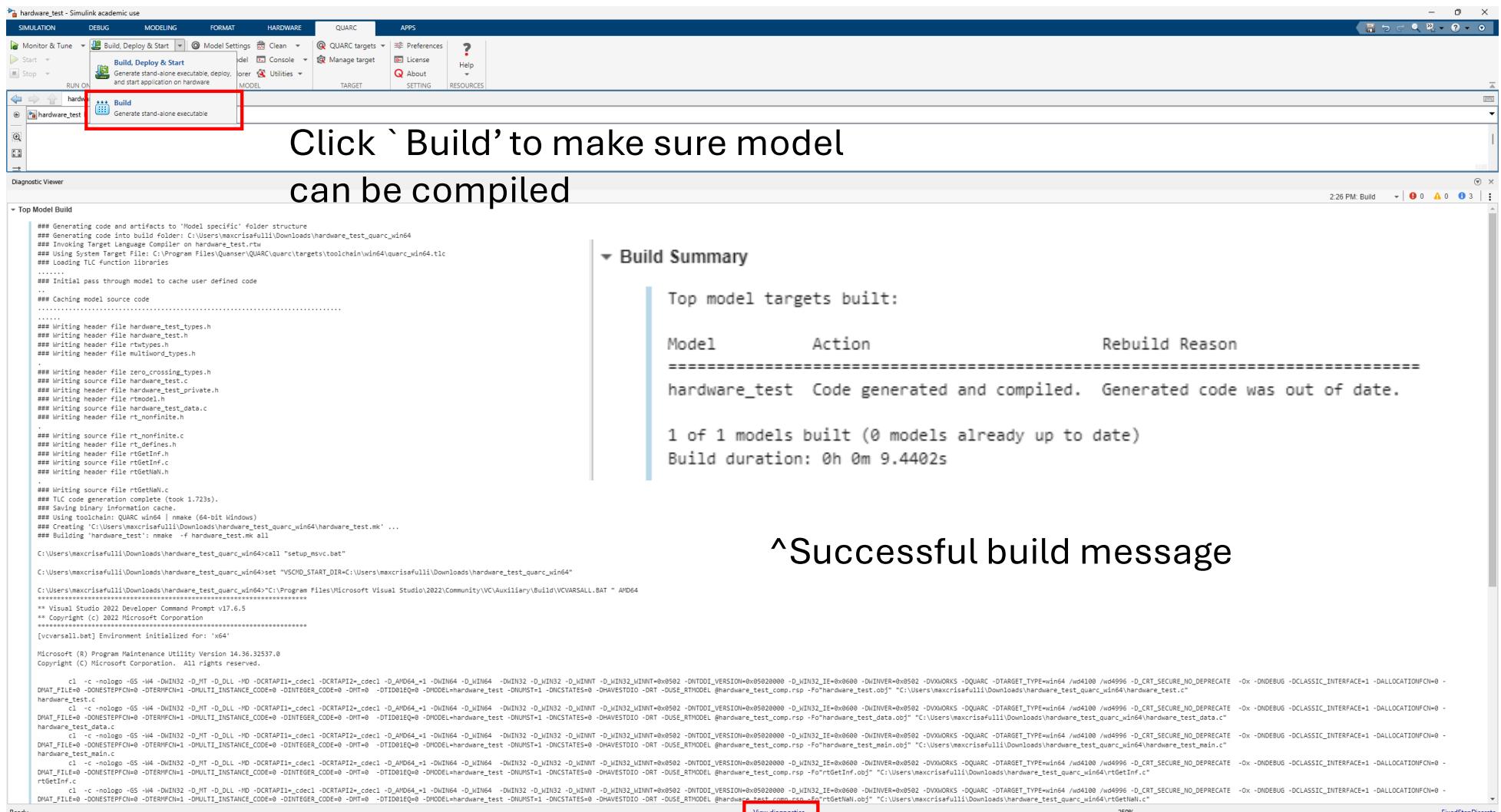
T under trigger means that will be used as trigger signal (don't change)

Duration is maximum # of samples logged, $Duration = t_{Stop}/T_s$



Demo model is reading all 4 encoders as vector and demuxxing (shown above)

Can set to just read a single encode



Monitor & Tune builds & runs the code. Expected outputs shown

