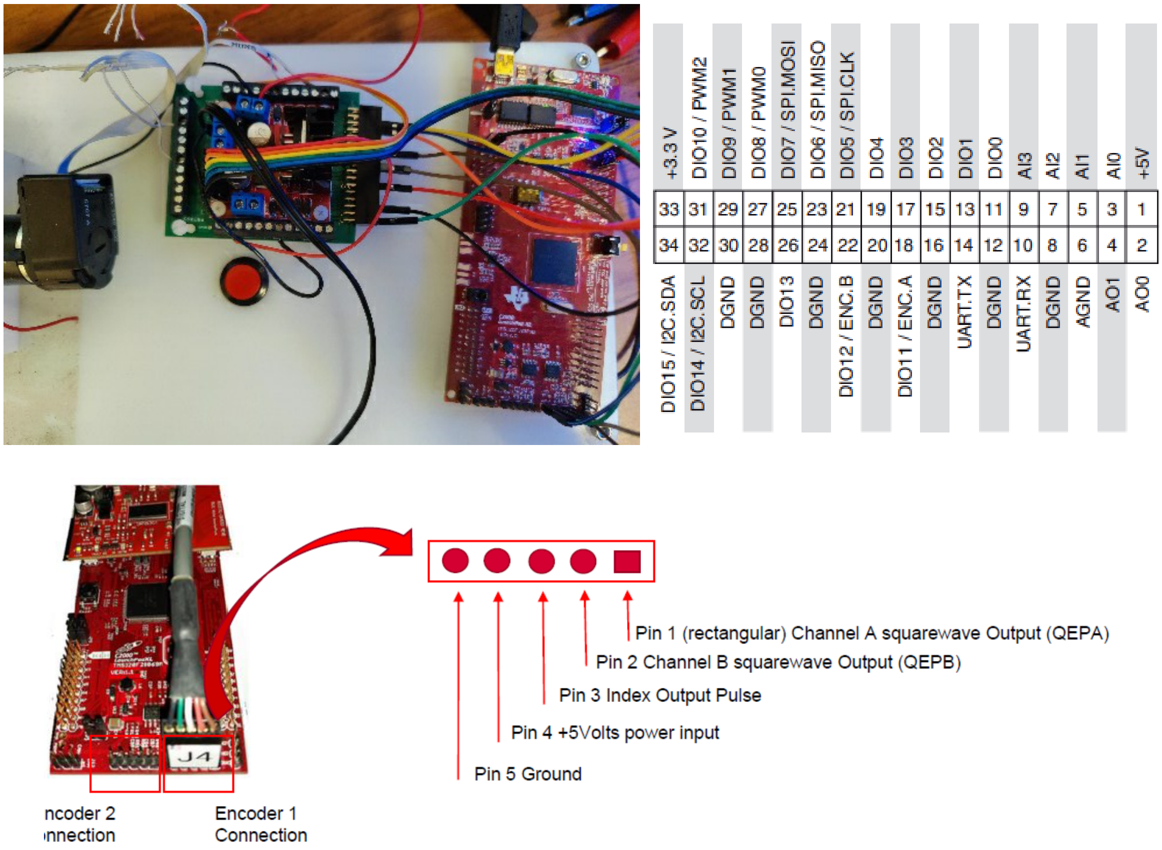
Setting up TI C2000 with Simulink Interface to the Motor-Pendulum Kit

Original: Han Wang 10/2022

Update: Tsu-Chin Tsao, 11/2022

This note shows you how to set up the hardware interface and Matlab software so that you can perform digital control of the Motor-Pendulum by Simulink.

Hardware Connection: You will need to use jumper wires to connect the PWM output of the C2000 to the motor drive circuit and the motor encoder inputs to C2000 (Figure 1). You may switch pins for PWM A and B, or Enc A and B as you want, which depends on the rotating direction definition. Since the Motor was connected to myRIO, Table 1 shows the mapping for converting the myRIO to C2000 connection.



+3.3 V	33	34	DIO15 / I2C.SDA
DIO10 / PWM2	31	32	DIO14 / I2C.SCL
DIO9 / PWM1	29	30	DGND
DIO8 / PWM0	27	28	DGND
DIO7 / SPI.MOSI	25	26	DIO13
DIO6 / SPI.MISO	23	24	DGND
DIO5 / SPI.CLK	21	22	DIO12 / ENC.B
DIO4	19	20	DGND
DIO3	17	18	DIO11 / ENC.A
DIO2	15	16	DGND
DIO1	13	14	UART.TX
DIO0	11	12	DGND
AI3	9	10	UART.RX
AI2	7	8	DGND
AI1	5	6	AGND
AI0	3	4	AO1
+5V	1	2	AO0

Pin 1 (rectangular) Channel A squarewave Output (QEPA)
Pin 2 Channel B squarewave Output (QEPB)
Pin 3 Index Output Pulse
Pin 4 +5Volts power input
Pin 5 Ground
J4
ncoder 2
ınnection
Encoder 1
Connection

Figure 1: hardware setup and myRIO pinout

Table 1: TI Launchpad and NI MyRIO pin mapping

Signal	myRIO breakout board pin	TI LaunchPad pin
PWM A	Pin 27	J4: pin 40 or 39
PWM B	Pin 29	J4: pin 39 or 40
Enable	Pin 11	J4: pin 38
Ground	Pin 6	J3: pin GND

Enc A	Pin 18	QEP_A: pin 1 or 2
Enc B	Pin 22	QEP_A: pin 2 or 1
Enc 5V	Pin 1	QEP_A: pin 4
Enc Ground	Pin 30	QEP_A: pin 5

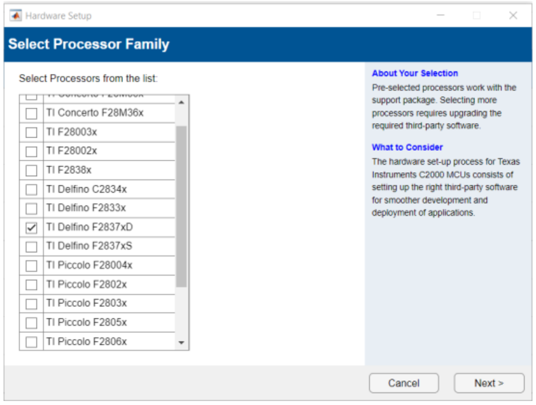
Software Environment:

MATLAB + Simulink R2022b, you are supposed to download and install by University of California Los Angeles - MATLAB Access for Everyone - MATLAB & Simulink (mathworks.com).

To generate C++ code for C2000, several Coder / Compilers Add-ons are needed.

In MATLAB Add-on, please install *MATLAB Coder*, *Simulink Coder*, *Embedded Coder*, *Embedded Coder Support Package for Texas Instruments C2000 Processors*, and *MATLAB Support for MinGW-w64 C/C++ Compiler.*

While installing *Embedded Coder Support Package for TI C2000*, to use C2000 Launchpad, you can just select "TI Delfino F2837xD", if you are planning to use more C2000 devices, select as needed.



Select Processor Family

Select Processors from the list:

- ☐ TI Concerto F28M36x
- ☐ TI F28003x
- ☐ TI F28002x
- ☐ TI F2838x
- ☐ TI Delfino C2834x
- ☐ TI Delfino F2833x
- ☑ TI Delfino F2837xD
- ☐ TI Delfino F2837xS
- ☐ TI Piccolo F28004x
- ☐ TI Piccolo F2802x
- ☐ TI Piccolo F2803x
- ☐ TI Piccolo F2805x
- ☐ TI Piccolo F2806x

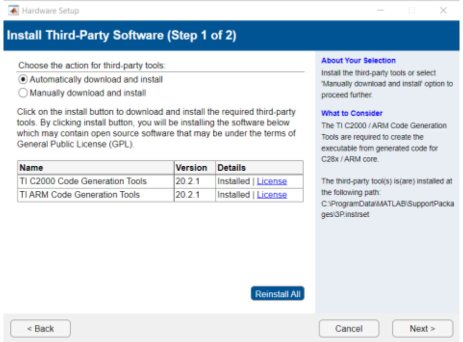
About Your Selection

Pre-selected processors work with the support package. Selecting more processors requires upgrading the required third-party software.

What to Consider

The hardware set-up process for Texas Instruments C2000 MCUs consists of setting up the right third-party software for smoother development and deployment of applications.

Then, follow the instruction and download all the 3rd party software. Remember to check version requirement carefully while downloading and installing those packages from TI's website manually. You may need to register a TI account to do that.



Install Third-Party Software (Step 1 of 2)

Choose the action for third-party tools:

● Automatically download and install

○ Manually download and install

Click on the install button to download and install the required third-party tools. By clicking install button, you will be installing the software below which may contain open source software that may be under the terms of General Public License (GPL).

Name	Version	Details
TI C2000 Code Generation Tools	20.2.1	Installed | License
TI ARM Code Generation Tools	20.2.1	Installed | License

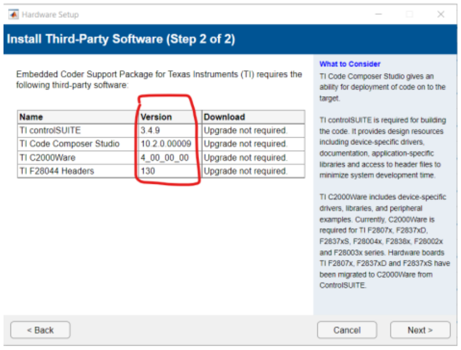
About Your Selection

Install the third-party tools or select 'Manually download and install' option to proceed further.

What to Consider

The TI C2000 / ARM Code Generation Tools are required to create the executable from generated code for C28x / ARM core.

The third-party tool(s) is(are) installed at the following path: C:\ProgramData\MATLAB\SupportPackages\3P.instrset



Install Third-Party Software (Step 2 of 2)

Embedded Coder Support Package for Texas Instruments (TI) requires the following third-party software:

Name	Version	Download
TI controlSUITE	3.4.9	Upgrade not required.
TI Code Composer Studio	10.2.0.00009	Upgrade not required.
TI C2000Ware	4_00_00_00	Upgrade not required.
TI F28044 Headers	130	Upgrade not required.

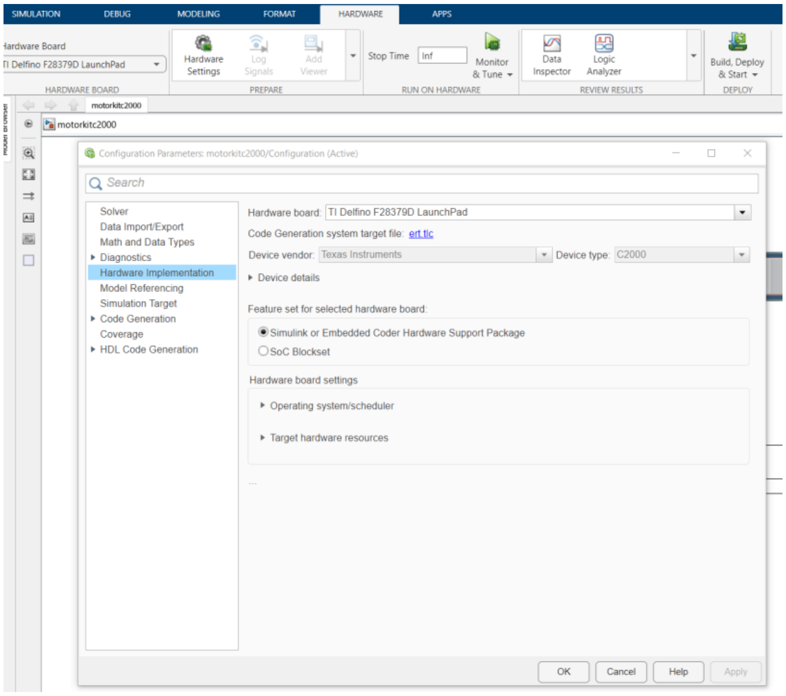
What to Consider

TI Code Composer Studio gives an ability for deployment of code on to the target.

TI controlSUITE is required for building the code. It provides design resources including device-specific drivers, documentation, application-specific libraries and access to header files to minimize system development time.

TI C2000Ware includes device-specific drivers, libraries, and peripheral examples. Currently, C2000Ware is required for TI F2807x, F2837xD, F2837xS, F28004x, F2838x, F28002x and F28003x series. Hardware boards TI F2807x, F2837xD and F2837xS have been migrated to C2000Ware from ControlSUITE.

In Simulink, please choose hardware and feature set as below before *Monitor&Tune*.



Hardware board: TI Delfino F28379D LaunchPad

Code Generation system target file: ert.tlc

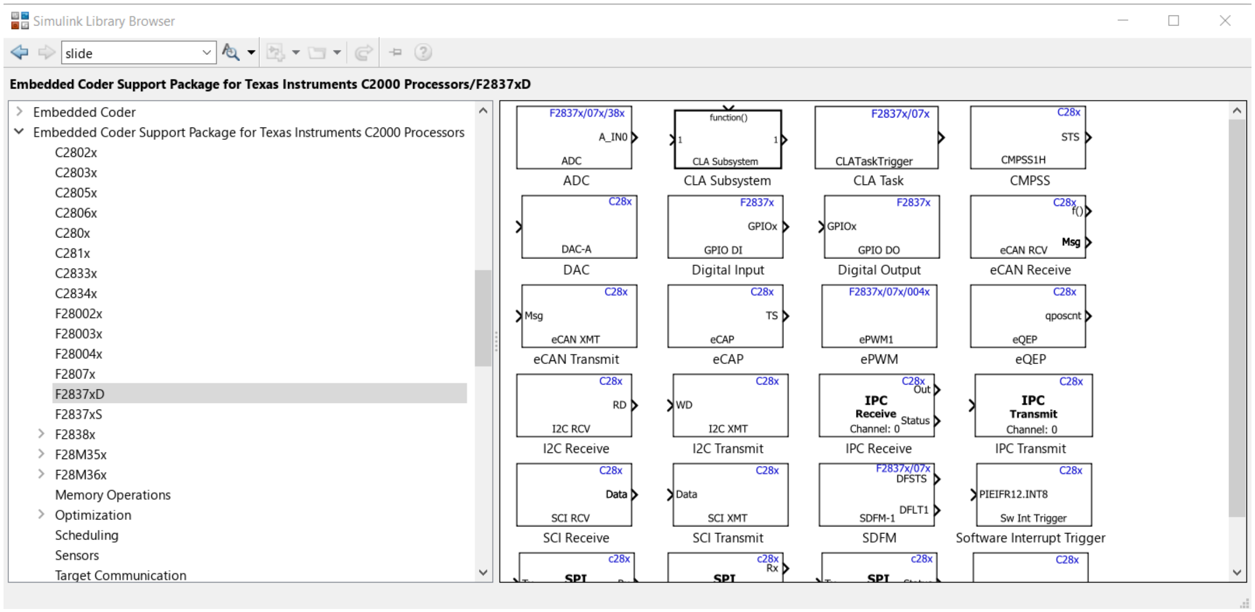
Device vendor: Texas Instruments Device type: C2000

Feature set for selected hardware board:

● Simulink or Embedded Coder Hardware Support Package

○ SoC Blockset

Then you can use all blocks in library to program. You can use eQEP to get Encoder feedback and ePWM to send out PWM duty cycle command.



Embedded Coder Support Package for Texas Instruments C2000 Processors/F2837xD

If you have any questions about programming for TI c2000 launchpad, please check references below. These manuals explain all the features you can use on TMS320F28379D processor.

LAUNCHXL-F28379D Overview User's Guide (Rev. C) (ti.com)

TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual (Rev. I) (ti.com)

Implementation Guide

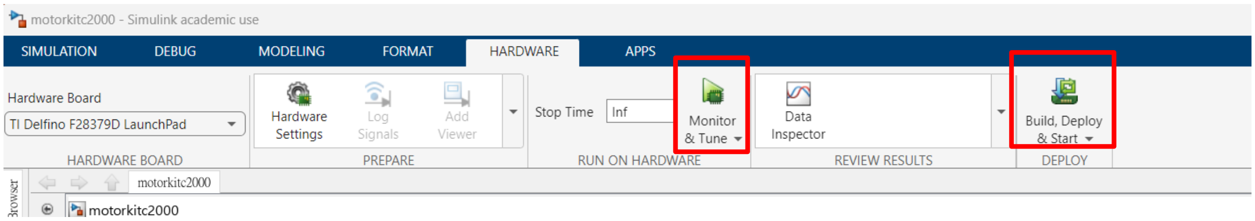
Will Shih, Jimmy Yang

When using C2000 with Simulink, please note that there are two "run" buttons on the top of the “Hardware” tab – “Monitor & Tune” and “Build, Deploy and Start”.

When using “Monitor & Tune”, the C2000 board is running the program and connected to the Simulink software at the same time. Interactive blocks in Simulink such as the scope, or button allows you to monitor and interact with the hardware after clicking Monitor and Tune. You can also change the parameters in your Simulink blocks in real-time while the program is running. However, you should NOT disconnect the board from your computer while running “Monitor & Tune”.

When using “Build, Deploy and Start”, Simulink compiles the program and uploads it to the C2000 board. Once the upload process is done, the board will be running independently without real time communication with Simulink. You cannot monitor the data in the board unless you use other communication protocols such as UART (via serial COM port).

In our Motor-Pendulum Kit case, use “Monitor & Tune” to see the pendulum position output from the encoder and tune the PID gains on the fly.

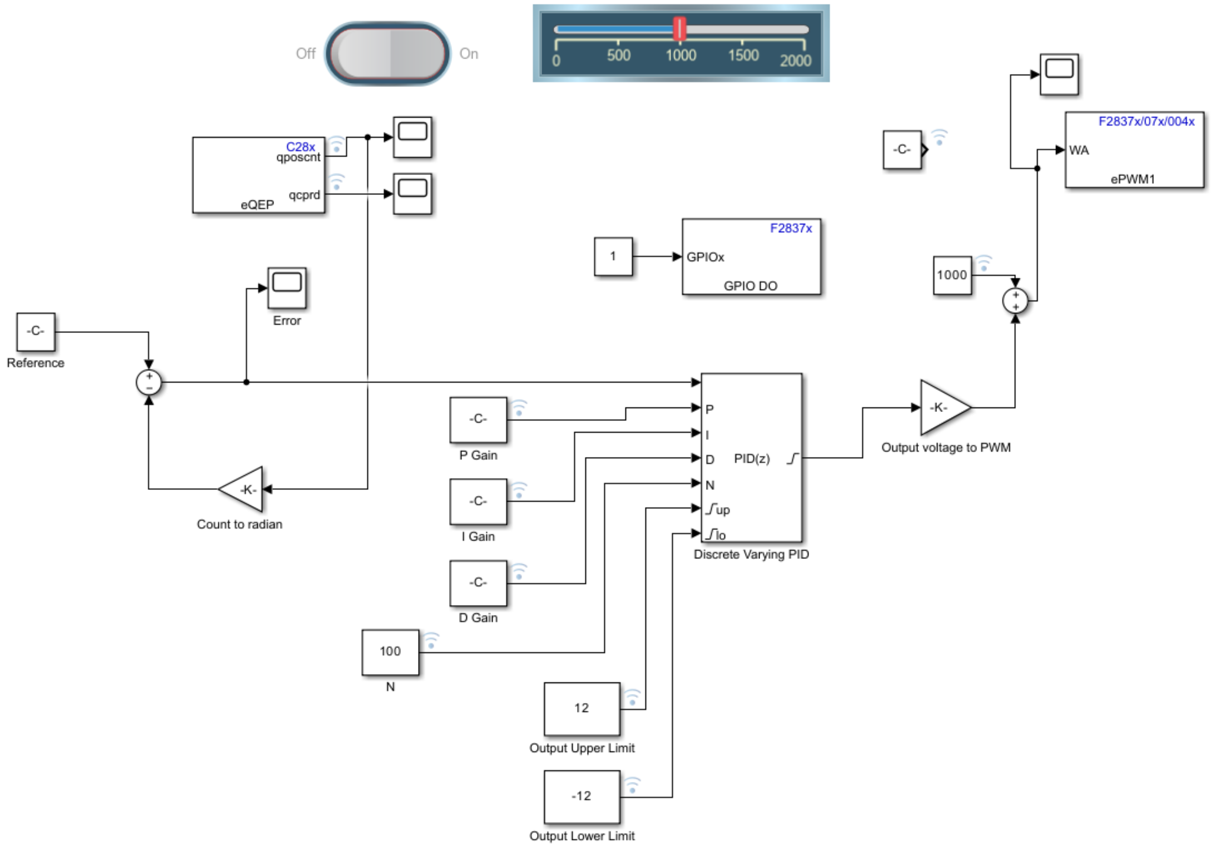


motorkitc2000 - Simulink academic use

Hardware Board: TI Delfino F28379D LaunchPad

Stop Time: Inf

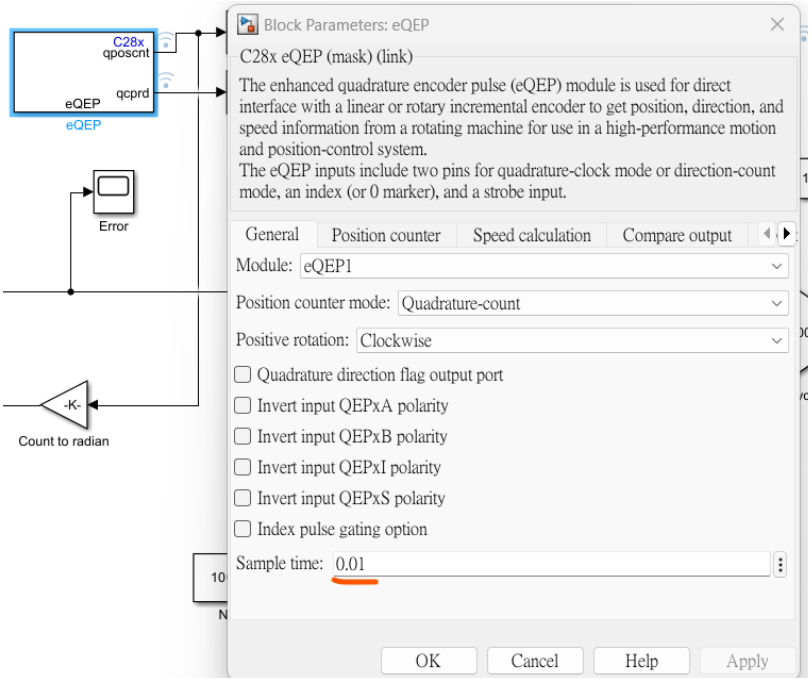
Below is a controller block diagram for reference. You can design and implement your own controller as well.



The PID controller block has the transfer function shown below. You can double click the PID controller block to see more a detailed configuration and fine tune it if necessary. Compare the parameters of this transfer function with yours to figure out the P gain, I gain and D gain. Please note that N is equal to the inverse of the sampling of rate, which you can determine or adjust in the eQEP block described in the next paragraph.

$$P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}}$$

The eQEP block obtains signal from the encoder mounted behind the motor. When running the program, you can change the sampling time in the block by double clicking the block and changing the Sample Time as underlined in the figure below.



Block Parameters: eQEP

C28x eQEP (mask) (link)

The enhanced quadrature encoder pulse (eQEP) module is used for direct interface with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine for use in a high-performance motion and position-control system.
The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input.

General | Position counter | Speed calculation | Compare output

Module: eQEP1

Position counter mode: Quadrature-count

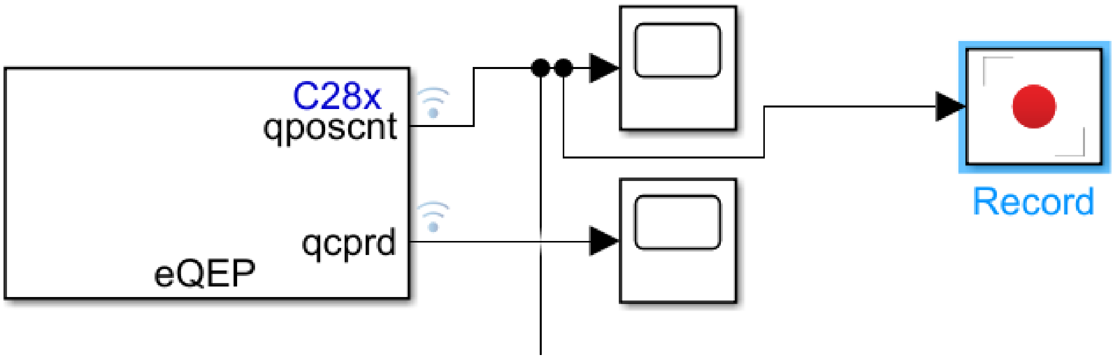
Positive rotation: Clockwise

- □ Quadrature direction flag output port
- □ Invert input QEPxA polarity
- □ Invert input QEPxB polarity
- □ Invert input QEPxI polarity
- □ Invert input QEPxS polarity
- □ Index pulse gating option

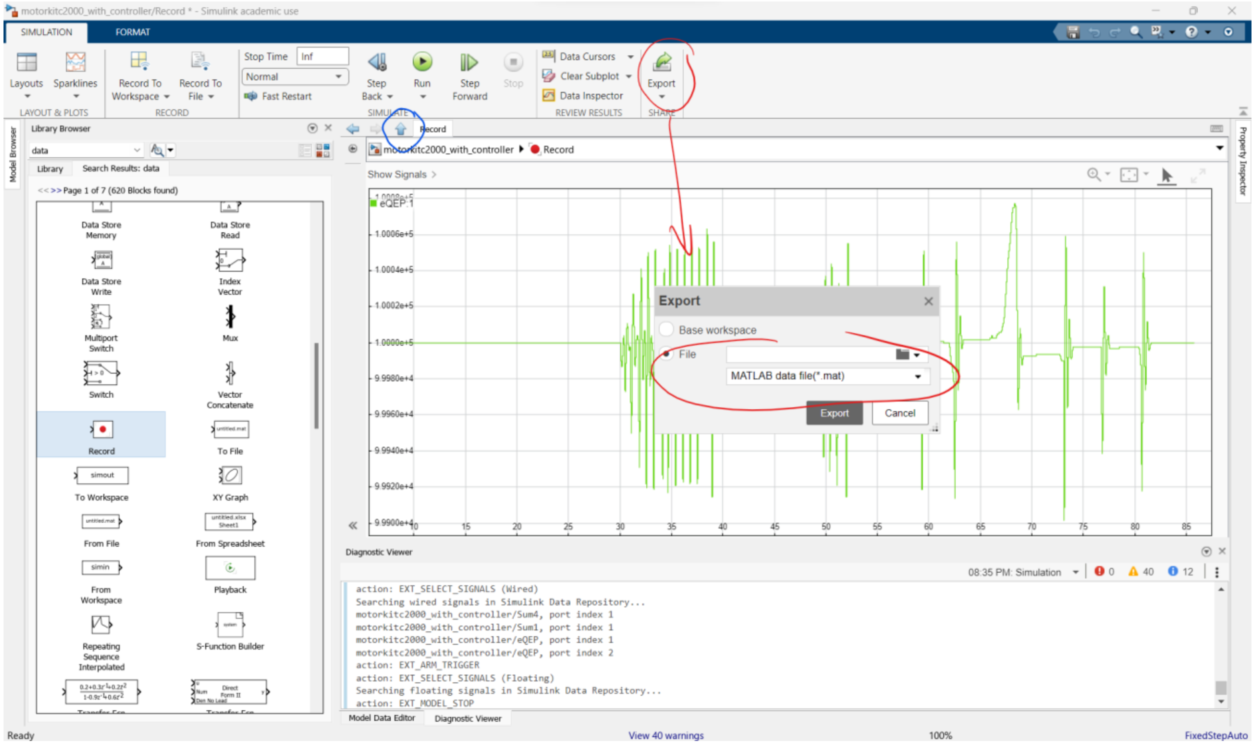
Sample time: 0.01

After hitting the Monitor and Tune button, please make sure to keep the pendulum upright while the controller is being uploaded onto the C2000, or else the reference setpoint will be incorrect.

To save the output data, you can connect the “Record” block to the desired signal to be recorded. For example, for position data, the connection is as follows:



After clicking “Monitor and Tune”, double click on the Record block (note: to exit to the block diagram, click on the arrow pointing up, circled in blue on the diagram below). When you are finished, click “Stop” on top. The Export button will be available, and you can choose to export to a file.



Export

○ Base workspace

● File

MATLAB data file(*.mat)

Export Cancel

```
action: EXT_SELECT_SIGNALS (Wired)
Searching wired signals in Simulink Data Repository...
motorkitc2000_with_controller/Sum4, port index 1
motorkitc2000_with_controller/Sum1, port index 1
motorkitc2000_with_controller/eQEP, port index 1
motorkitc2000_with_controller/eQEP, port index 2
action: EXT_ARM_TRIGGER
action: EXT_SELECT_SIGNALS (Floating)
Searching floating signals in Simulink Data Repository...
action: EXT_MODEL_STOP
```

The following is a set of sample code to read the and plot the data contents. Note that the data from the recording was saved as “test_recording.mat”.

```
x = load('test_recording.mat')

plot(x.data{1}.Values.Time, x.data{1}.Values.Data)
```