Interfacing BiSS-C Absolute Encoder from Simulink® - (F2837x)

This document will briefly summarize how to interface a BiSS-C Absolute Encoder using the Texas Instruments (TI) Position Manager board and TI F28379D launchpad board using the *Embedded Coder® Hardware Support Package for TI C2000™ Processors:*

https://www.mathworks.com/matlabcentral/fileexchange/43096-embedded-coder-support-package-for-texas-instruments-c2000-processors

Overview of demo model

This model is based on the reference design note as published by TI for interfacing the BiSS-C absolute encoder using the Position Manager technology. The model file 'F2837x_CPU1_bissc.slx' would run on CPU1 to read the position information and transmit the same to CPU2 via IPC. The model 'F2837x_CPU2_bissc.slx' is configured to run in external mode on CPU2 and would display the position information as received from CPU1. Applications generally capture position on one CPU core and using the captured position in algorithms in another CPU core. Hence the approach.

Refer to the demo video on this example here:

https://www.youtube.com/watch?v=FxRkGY6WPRM&t=23s

Required hardware:

- TI F28379D launchpad
- TI Position Manager Booster Pack
- Absolute encoder with BiSS-C interface

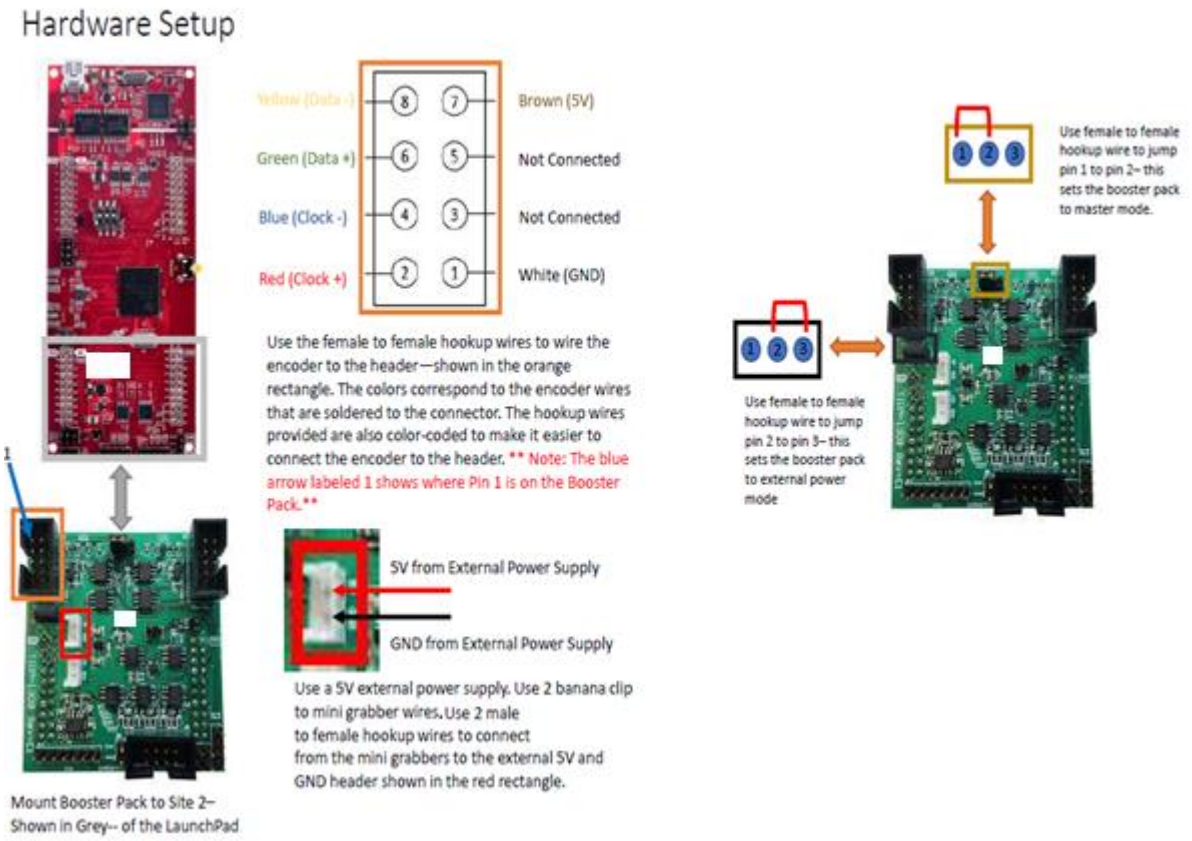
BiSS is an open-source digital interface for sensors and actuators. BiSS-C is a pure serial, digital interface based on the RS-485 standard. The interface transmits position values or additional physical quantities from the encoder (BiSS-C Slave) to the MCU (BiSS-C Master). Please refer the TI reference design document for more details:
https://www.ti.com/lit/tidue73

TI provides C2000 Position Manager BiSS-C (PM_bissc) library that supports implementing the BiSS-C interface in subsequent electronics. The BiSS-C Library features an integrated-MCU solution for the BiSS-C interface that meets the digital-interface protocol requirements. Right now, the implementation of the demo model is using the routines as provided in this library to initialize the BiSS interface and read the position information.

Please refer the C2000 Position Manager BISS-C Library user guide for more details on the library.
https://www.ti.com/lit/ug/sprui37/sprui37.pdf

Implementation scheme

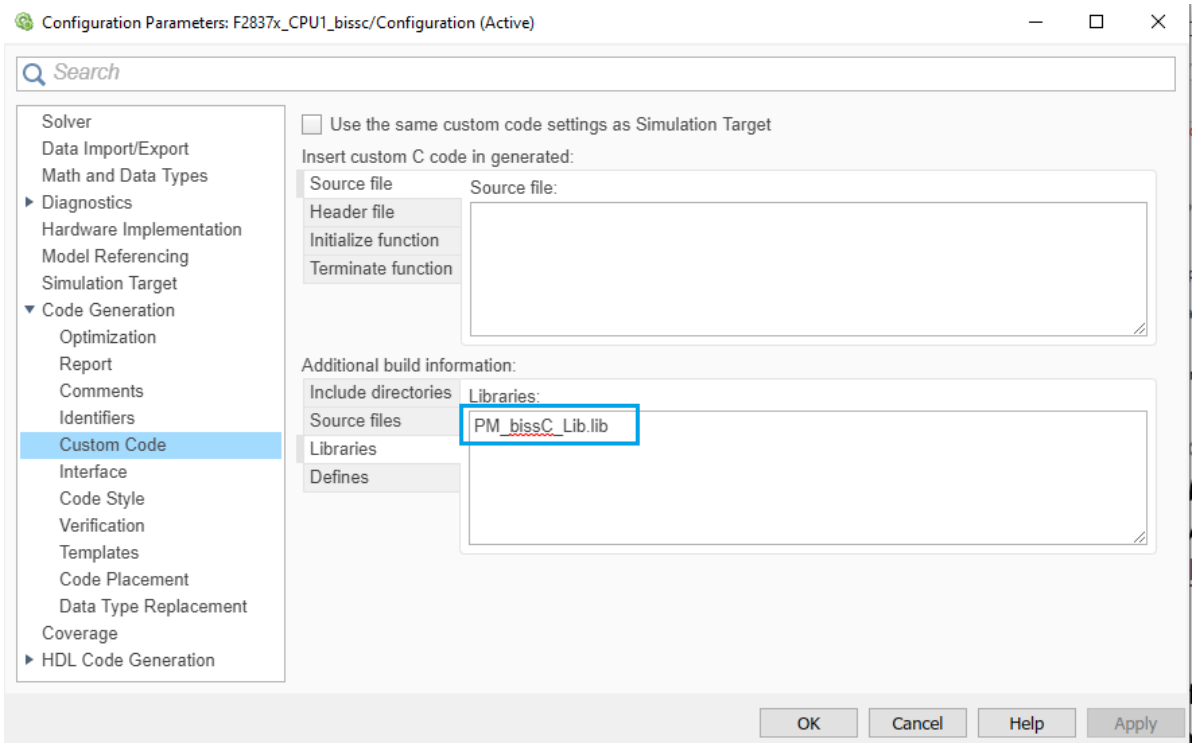
1. Hardware setup

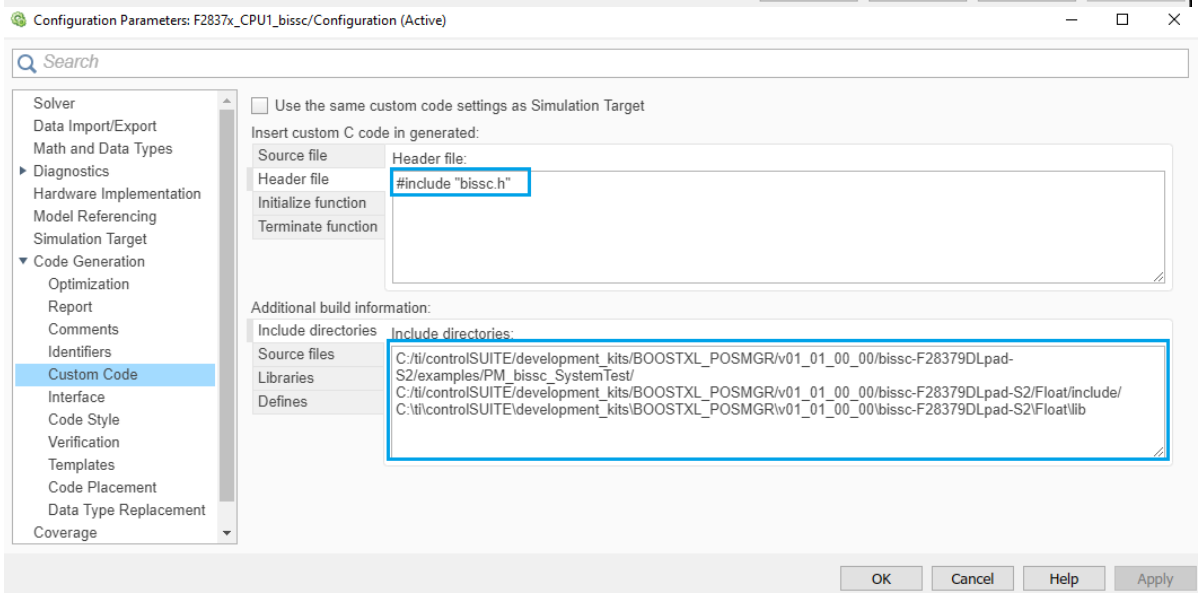


Note: we have used the LinACE™ absolute linear shaft encoder for this demo. The hardware setup is true for this device. Refer below link for datasheet.
https://resources.renishaw.com/download.aspx?lang=en&data=82135&btn=1

2. Using PM_bissc library from TI

To begin with, the header and source files (BiSS-C Library 'PM_bissC_Lib.lib' as provided by TI) is used. To include the header and source files for use within the Simulink, use custom code under Code Generation option as shown below.

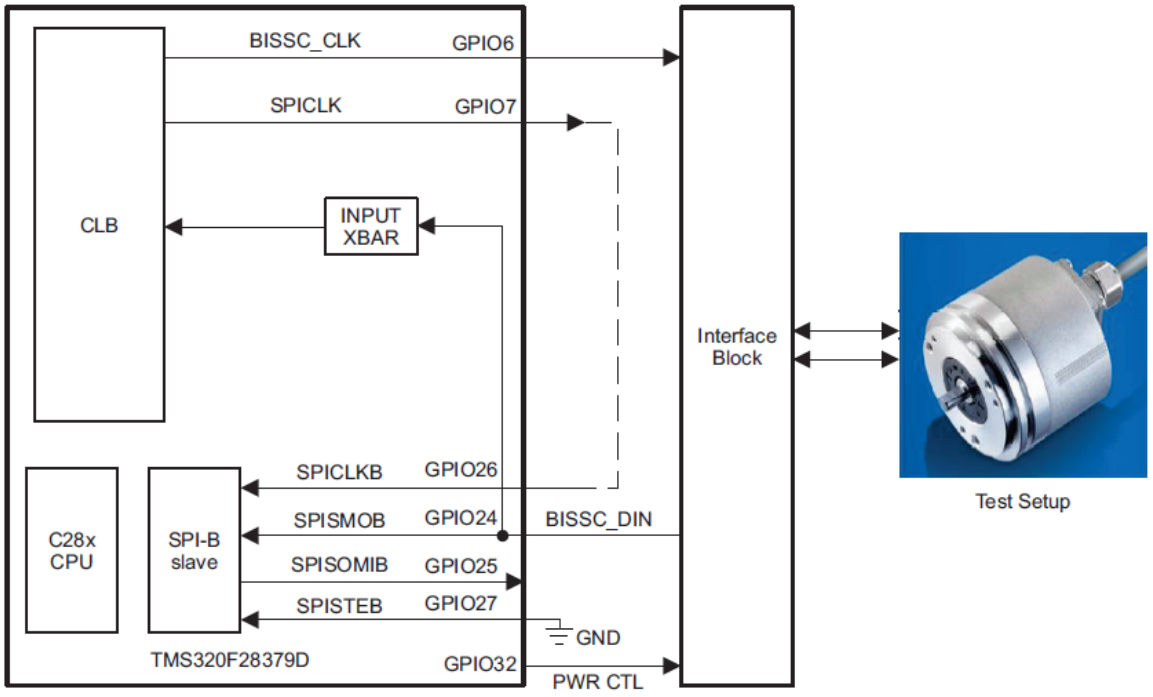




modify/update the path to ControlSUITE if found different in your setup.

3. **Assigning peripherals to CPU1 for BiSS-C operation**

The below figure is showing the key peripherals as used by the BiSS-C Library from the F2837x device to implement the BiSS-C interface on a C2000 Position Manager.

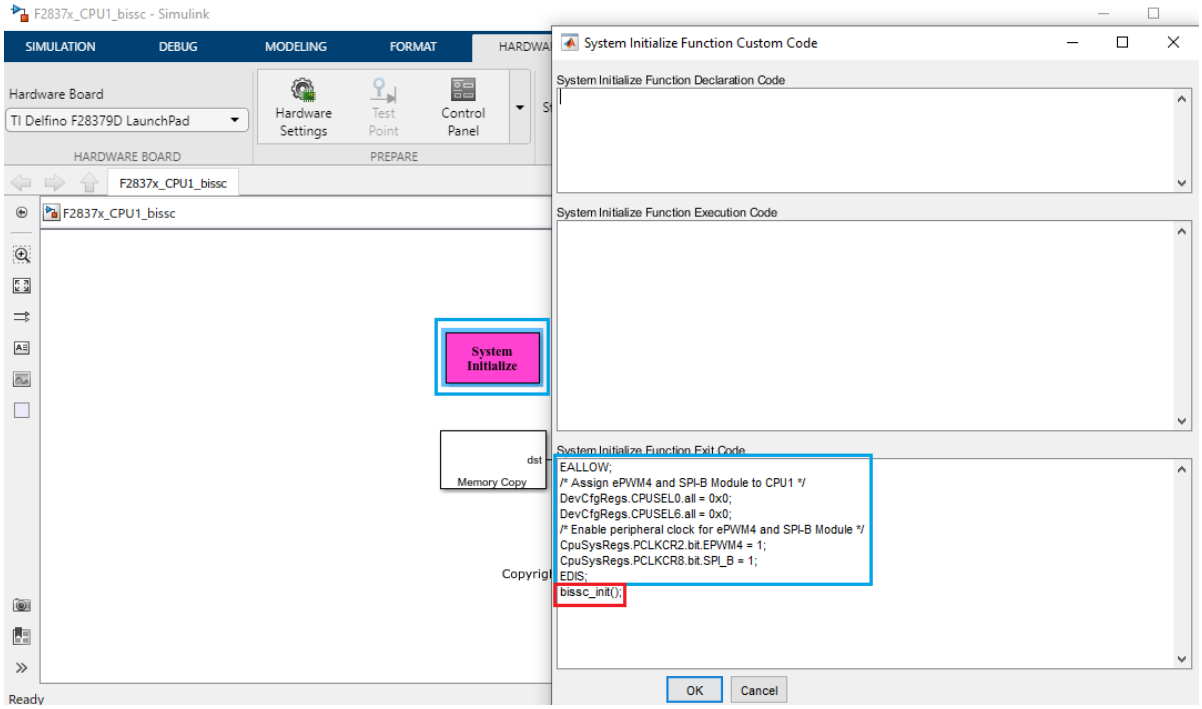


RESOURCE NAME	TYPE	PURPOSE	USE RESTRICTIONS
DEDICATED RESOURCES			
GPIO6	I/O	BiSS-C clock from master to encoder	I/O dedicated for BiSS-C
GPIO7	I/O	SPI clock generated by MCU	I/O dedicated for BiSS-C
EPWM4	I/O	Internal for clock generation	EPWM4 dedicated for BiSS-C
GPIO32	I/O	BiSS-C power control on LaunchPad	Dedicated I/O for encoder power enable
CONFIGURABLE RESOURCES			
SPI	Module and I/Os	One SPI instance to emulate the BiSS-C interface (SPIB on LaunchPad)	Any instance of SPI can be chosen — module and corresponding I/Os are dedicated for BiSS-C
SHARED RESOURCES			
CPU and memory	Module	Check CPU and memory use for various functions	Application to ensure enough CPU cycles and memory are allocated
Input XBAR	Module, I/O	Connected to SPISIMOB of the corresponding SPI instance dedicated for BiSS-C	INPUTXBAR1 is used for BiSS-C implementation, remaining inputs are available for application use

In Simulink, for F2837x device, all peripherals are by default assigned to CPU2. Based on which peripheral blocks are used in CPU1 model, only those

peripherals are re-assigned to CPU1. Rest of the peripherals remain with CPU2. But in the current implementation we are not using any peripheral blocks, but directly using the routines from BiSS-C Library to initialize and run BiSS-C interface on CPU1. Since Simulink would never know about the library using peripherals like SPI and ePWM on CPU1(note we never use any blocks here) hence we must explicitly assign these peripherals to CPU1 to enable the library to work properly on CPU1.

This is shown in below figure.

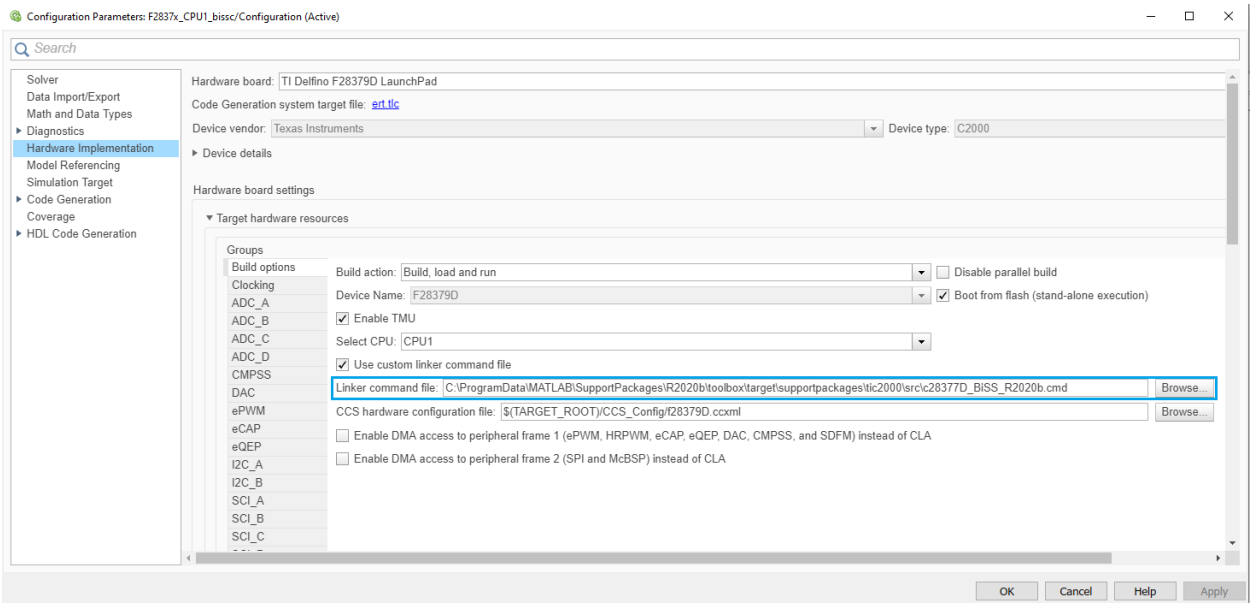


Note BiSS-C initiation routine is called from here.

4. Setup CLB peripheral definitions

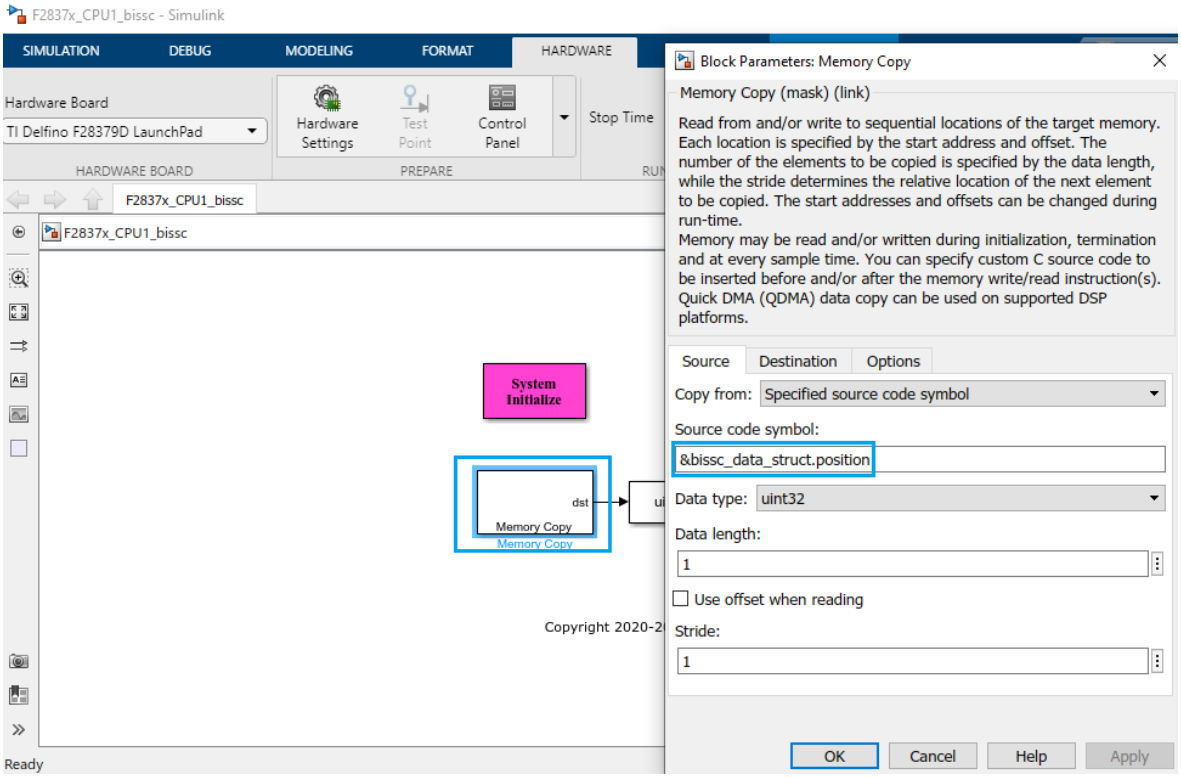
The BiSS-C Library makes use of the CLB peripheral. In R2020b, the CLB register definitions are not yet available. Hence, we need to add the same to get the BiSS-C interface to work.

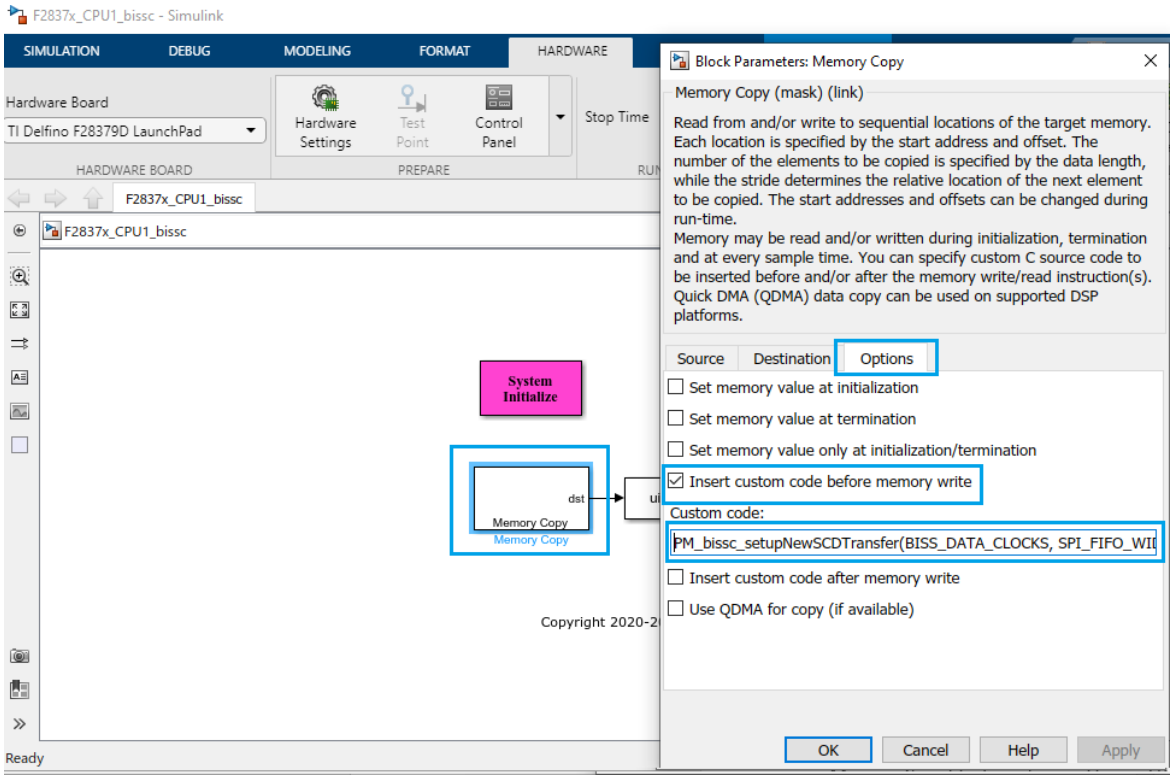
Place this c28377D_BiSS_R2020b.cmd provided with model file at path as shown below. Run command 'matlabshared.supportpkg.getSupportPackageRoot' to get support package install path. Point to the path as shown below in both 'F2837x_CPU1_bissc.slx' and 'F2837x_CPU2_bissc.slx' model files.



5. **<u>Measure position in CPU1</u>**

   Memory Copy block can be used to read the position information. The block also provides to invoke the BiSS-c runtime routines that would setup the protocol transfer to read the position.





6. Read the measured position in CPU2 model in external mode

- After the build and download of code from 'F2837x_CPU1_bissc.slx' is done, run the 'F2837x_CPU2_bissc.slx' in external mode.
- Vary the position the of the linear shaft encoder and observe the change in the position value on CPU2.