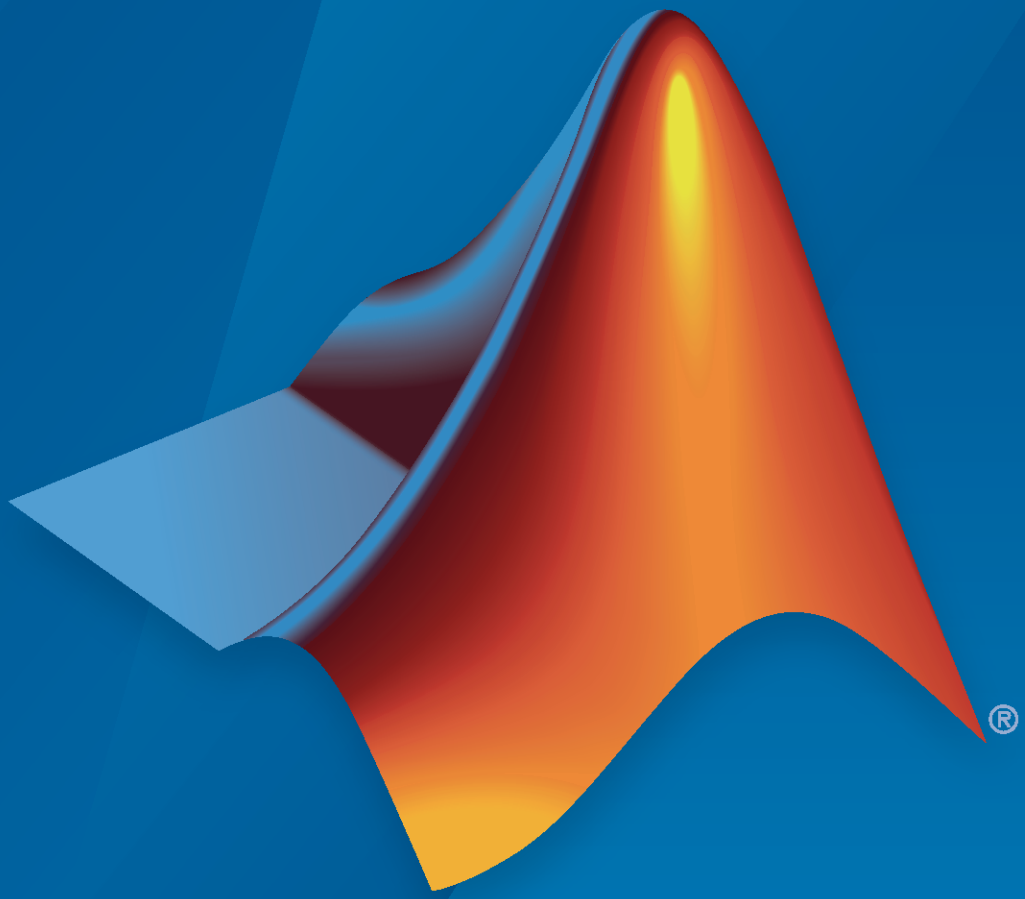
Motor Control Blockset™

Sensor-Based Position Control of PMSM by Running Field-Oriented Control (FOC) on Microchip's dsPICDEM™ MCLV-2 Development Board - ATSAME70Q21



MATLAB®

R2022a

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

<Title tag in book>

Revision History

November 2022	First printing	"Release for R2022a"

Contents

1 About This Example

Introduction **1-2**

Hardware Requirements **1-3**

Software Requirements **1-4**
Required MathWorks Products **1-4**
Required Microchip Products **1-4**

Hardware Setup **1-5**

Contents of Downloaded ZIP Folder **1-8**

2 Sensor-Based Position Control of PMSM by Running Field Oriented Control (FOC) on Microchip Hardware

Simulate Target Model **2-2**

Generate Code and Deploy Model to Target Hardware **2-4**

Verification Test **2-6**

About This Example

- “Introduction” on page 1-2
- “Hardware Requirements” on page 1-3
- “Software Requirements” on page 1-4
- “Hardware Setup” on page 1-5
- “Contents of Downloaded ZIP Folder” on page 1-8

Introduction

This example implements the field-oriented control (FOC) technique to control the position of a three-phase permanent magnet synchronous motor (PMSM). The example uses the quadrature encoder sensor to determine the rotor position.

The motor control algorithm runs on Microchip dsPICDEM™ MCLV-2 Development Board. This board provides a cost-effective solution to develop and evaluate either 3-phase sensor-based or 3-phase sensorless Brushless DC (BLDC) and PMSM control applications. It also supports Microchip's 100-pin motor control Plug-In-Modules (PIMs) with these controllers:

- dsPIC33C, dsPIC33E, and dsPIC33F Digital Signal Controllers (DSCs)
- PIC32MK, ATSAME7x, ATSAME5x, and ATSAMC2x controller families

The board supports usage of internal on-chip op-amps found on certain dsPIC® or PIC32MK devices and the external op-amps provided on the MCLV-2 board. For more details about the hardware, see dsPICDEM™ MCLV-2 Development Board (DM330021-2).

Hardware Requirements

This section lists the hardware requirements to run the example:

1. MCLV-2 Development Board (DM330021-2)
2. ATSAME70Q21 Motor Control Plug-In-Module (MA320203) with EXTERNAL op-amp matrix board
3. ICD 4 In-Circuit Debugger (DV164045) or PICkit 4 In-Circuit Debugger (PG164140)
4. Debugger Adapter Board for MPLAB ICD4 (AC102015) (required only if you use ICD4 as a debugger)
5. 24V Power Supply (AC002013)
6. 24V 3-Phase Brushless DC Permanent Magnet Motor Hurst motor (AC300022)
7. FTDI cable / FT232RL FTDI USB to UART

Software Requirements

This section lists the software products from MathWorks® and Microchip that you need to simulate and run the example models on the dsPICDEM™ MCLV-2 Development Board.

Required MathWorks Products

To simulate the example model:

- MATLAB®
- Simulink®
- Motor Control Blockset
- Stateflow®

To generate code and deploy the example model:

- MATLAB
- Simulink
- Motor Control Blockset
- Embedded Coder®
- Simulink Coder™
- MATLAB Coder

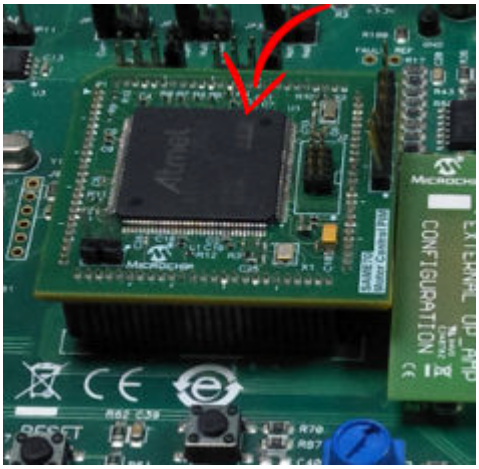
Required Microchip Products

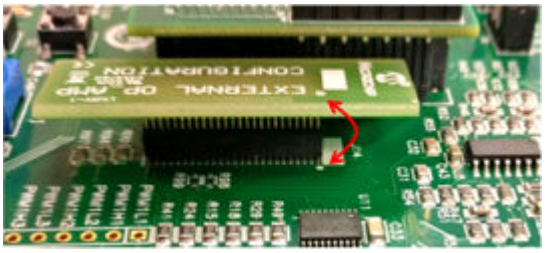
- MPLAB X IDE and IPE (v5.45 or later)
- XC32 compiler (v4.0 or later)
- MPLAB Device Blocks for Simulink Toolbox

Hardware Setup

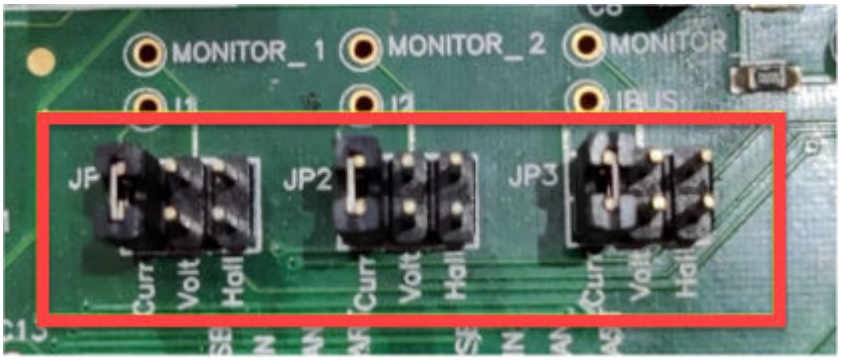
Follow these instructions to setup the hardware:

1. Connect the ATSAME70Q21 Plug-In-Module (PIM) and External Op-Amp matrix board to the MCLV-2 motor control board.



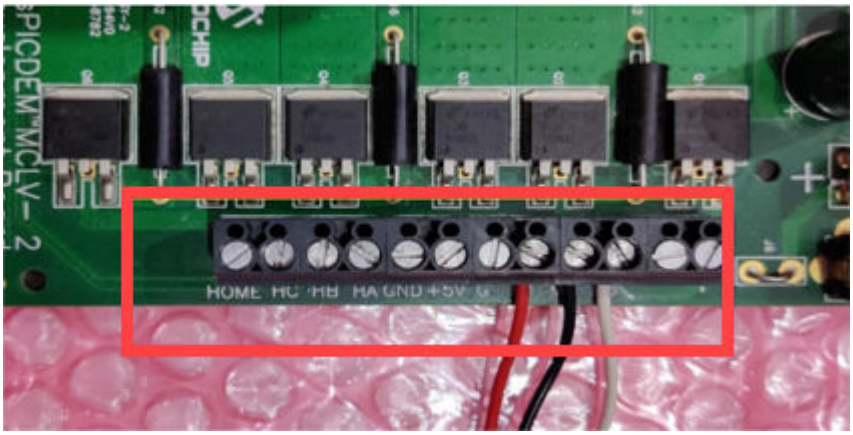


2. Keep the jumper connections of JP1, JP2, and JP3 at Curr position.

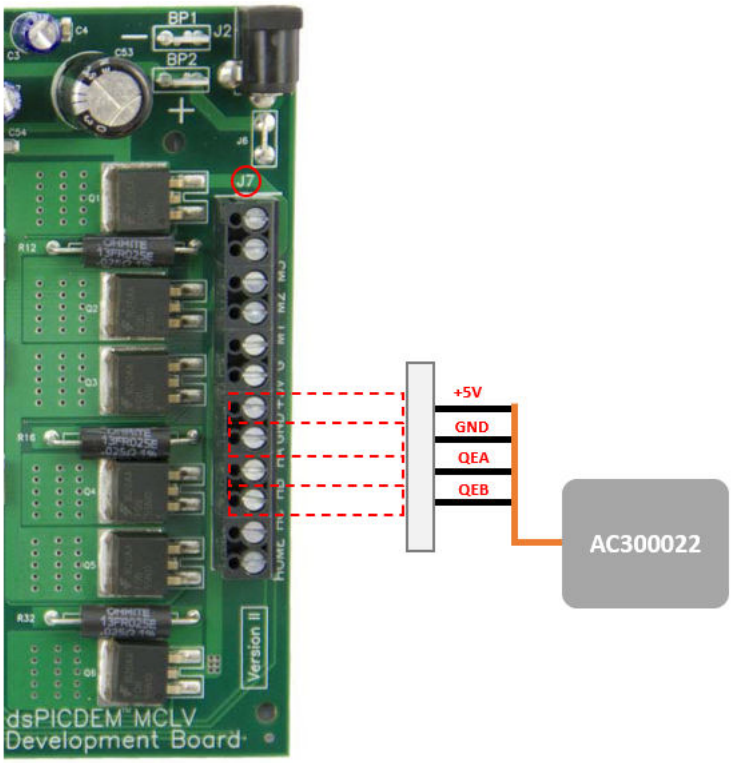


3. The following image describes the Hurst Motor phase connections.

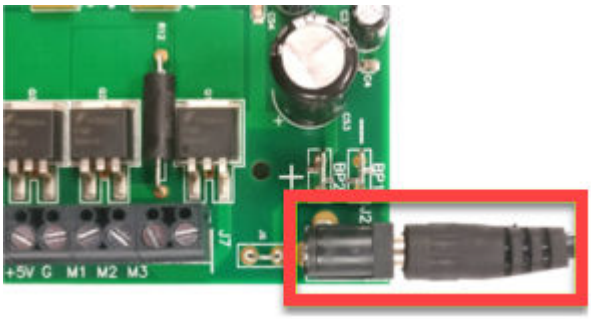
Note The connection sequence of 3-phase wires is irrelevant for sensorless algorithm.



4 Connect the quadrature encoder cables from the AC300022 motor to the J7 connector on the MCLV2 board.



5 Connect the 24V power supply to the MCLV2 board.



6 Connect the ICD 4 and adaptor board to ATSAME70Q21 PIM on the MCLV2 board.

Contents of Downloaded ZIP Folder

The ZIP package that you downloaded from the GITHUB® repository, includes the following files inside the `Position Ctrl QEP` folder:

File Name	Description
`mcb_microchip_pmsm_posCtrl_SAME70.slx`	Target model
`mcb_microchip_pmsm_posCtrl_SAME70_data.m`	Target model initialization script
`mcb_microchip_pmsm_posCtrl_host_SAME70.slx`	Host model
`Quadrature_encoder_pos_ctrl_FOC_32_bit_SAME70.pdf`	Documentation

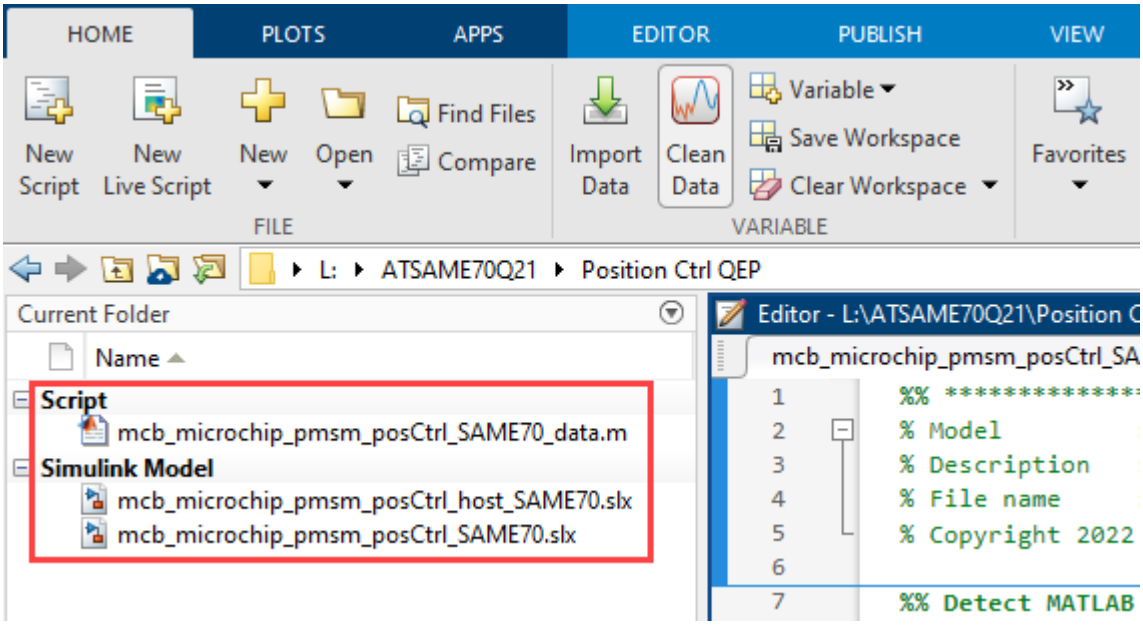
Sensor-Based Position Control of PMSM by Running Field Oriented Control (FOC) on Microchip Hardware

- “Simulate Target Model” on page 2-2
- “Generate Code and Deploy Model to Target Hardware” on page 2-4
- “Verification Test” on page 2-6

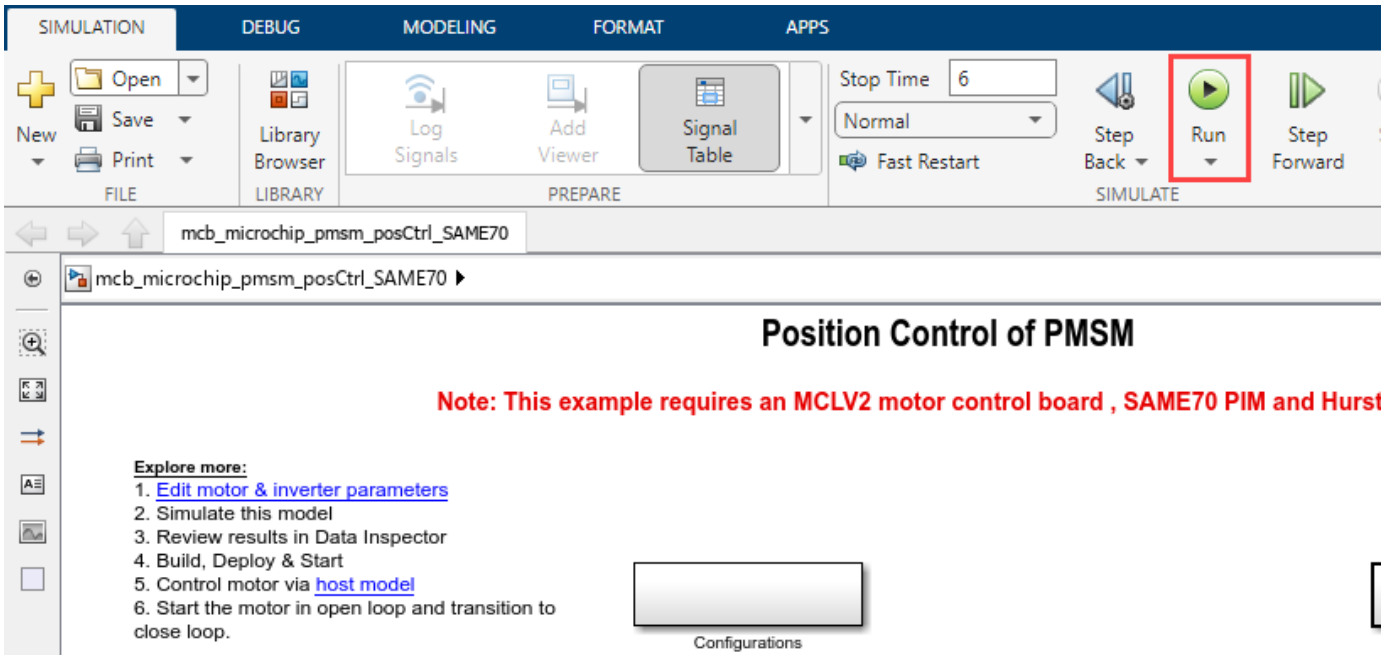
Simulate Target Model

This example supports simulation. Follow these steps to simulate the target model.

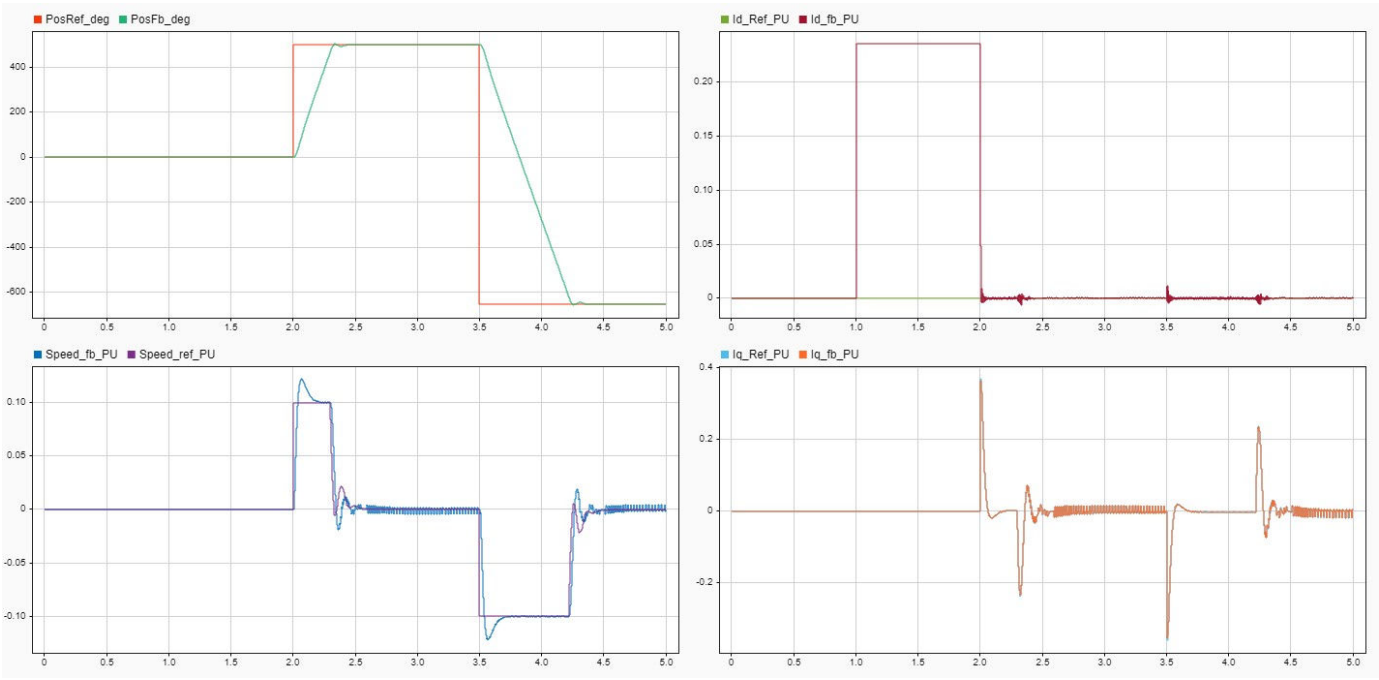
1 Open the target model `mcb_microchip_pmsm_posCtrl_SAME70.slx` included in this workflow.



2 Click **Run** on the **Simulation** tab to simulate the model.



3 Click **Data Inspector** on the **Simulation** tab to view and analyze the simulation results.


PosRef_deg
PosFb_deg
Id_Ref_PU
Id_fb_PU
Speed_fb_PU
Speed_ref_PU
Iq_Ref_PU
Iq_fb_PU

Generate Code and Deploy Model to Target Hardware

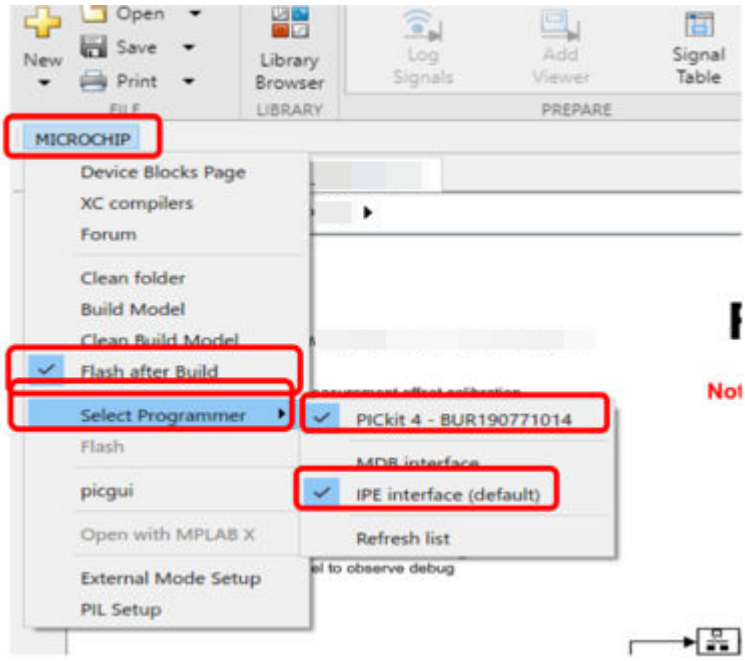
This section explains how to generate code and run the FOC algorithm on the target hardware.

Follow these steps to deploy and run the target model on the dsPICDEM™ MCLV-2 Development Board kit.

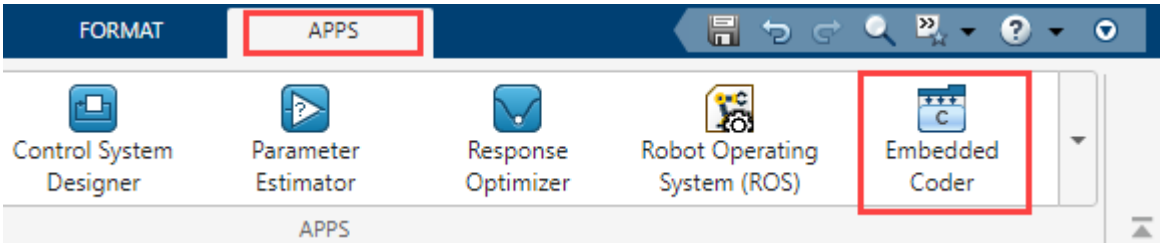
1. Simulate the target model and observe the simulation results.
2. Connect the dsPICDEM™ MCLV-2 Development Board kit to the host computer through a serial port.
3. Click the **Edit motor & inverter parameters** hyperlink available in the target model to open the target model initialization script. You can also use this command to open the initialization script.

   ```
   edit mcb_microchip_pmsm_posCtrl_SAME70_data.m
   ```
4. Verify and edit the motor and inverter parameters that are pre-configured in the model initialization script `mcb_microchip_pmsm_posCtrl_SAME70_data.m`.

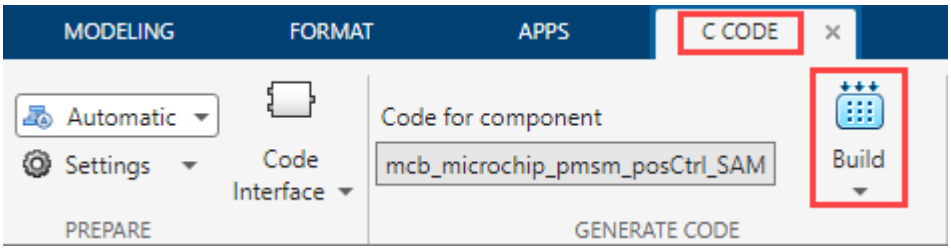
   For instructions to configure the script, see Estimate Control Gains from Motor Parameters.
5. To run the real PMSM motor, navigate to the **Microchip** tab and enable the tabs shown in the following figure. This enables you to select the programmer and then build and flash code to the microcontroller.



6. Click **Apps** tab and then click **Embedded Coder**.



7 The **C Code** tab appears on the menu bar. Click **C Code > Build** to build, generate, and flash the code.

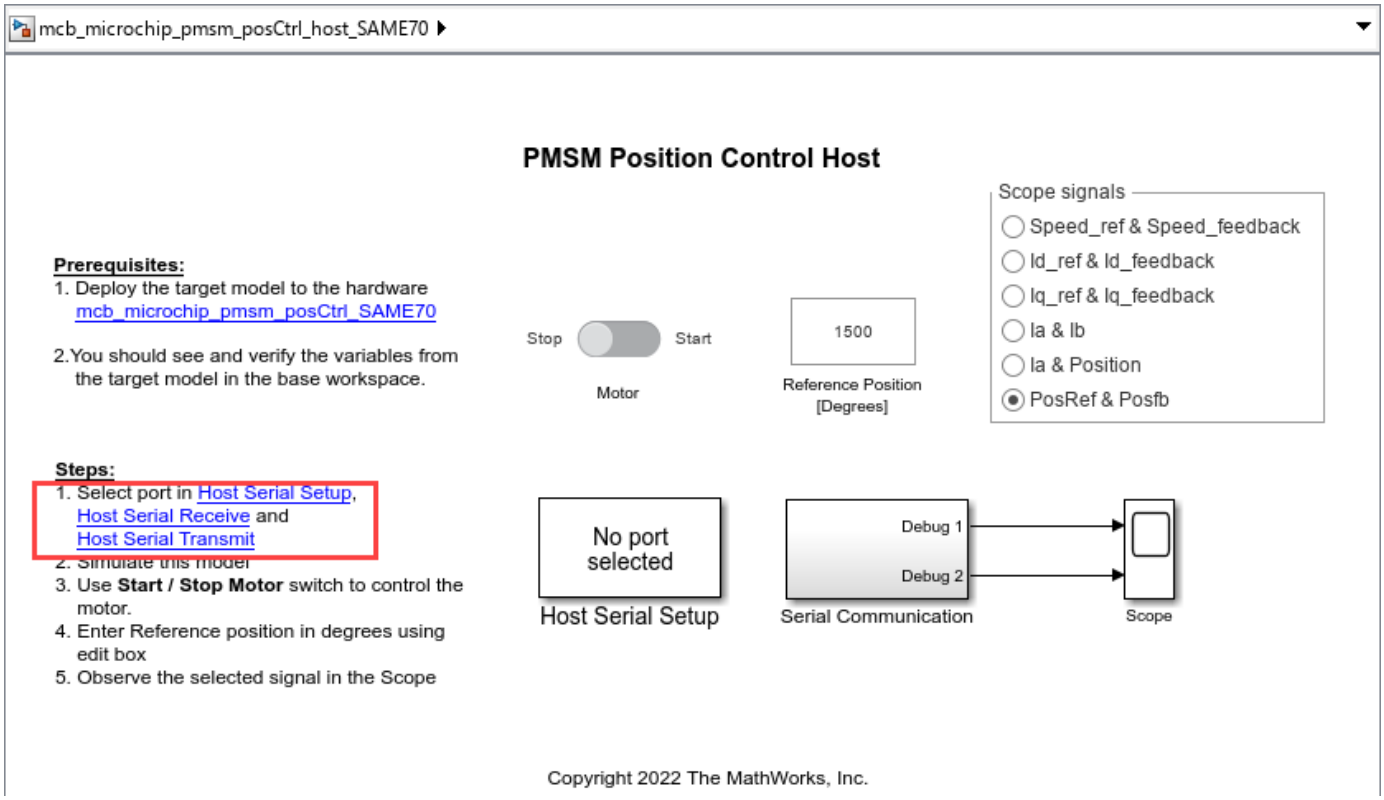


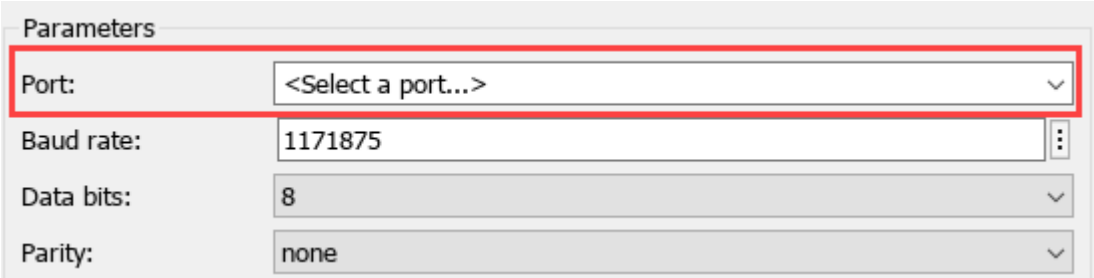
8 The system starts generating the code and opens the **Diagnostics Viewer** window. After few seconds, a notification window opens.

9 After code generation completes, the system flashes the code to dsPIC through the programmer. After flashing process completes the system displays **Operation Succeeded** message on the **Diagnostics Viewer**.

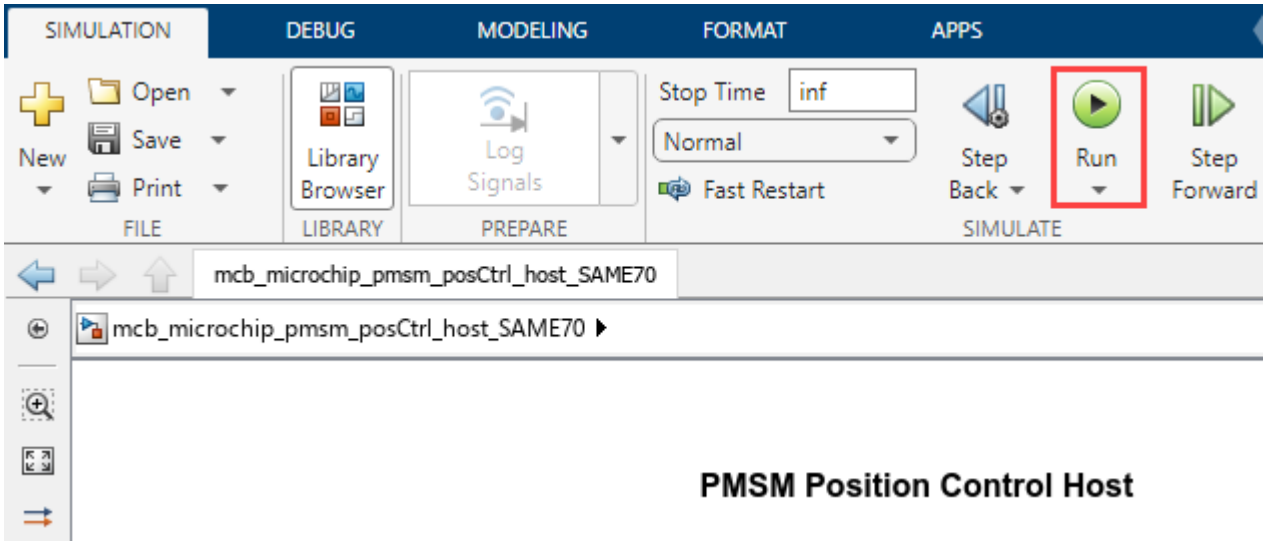
Verification Test

1. Check if the debug LED **D2** is blinking. A blinking LED **D2** confirms that the target device is successfully programmed.
2. Open the host model `mcb_microchip_pmsm_posCtrl_host_SAME70.slx`.
3. In the host model, click the links **Host Serial Setup**, **Host Serial Receive**, and **Host Serial Transmit** links to open the block parameters dialog boxes for the Host Serial Setup, Host Serial Receive, and Host Serial Transmit blocks. In these dialog boxes, select a **Port**.





4. Click **Run** on the **Simulation** tab to run the host model.



5 Update the **Reference Position [Degrees]** value in the host model.

6 Use the **Motor** switch available in the host model to start running the motor.

7 Use the **Scope** available in the host model to view the debug signals received from the target model running the motor.

8 You can select and view a different combination of debug signals using the **Scope signals** section of the host model. Use the **Motor** switch to stop the motor.