



1. INTRODUCTION

This document describes the setup requirements for driving a Switched Reluctance Motor (SRM) using dsPIC33AK128MC106.

For details about the speed control of SRM, refer to Microchip product brief "Switched Reluctance Motor Control Solution".

2. SUGGESTED DEMONSTRATION REQUIREMENTS

2.1 Motor Control Application Firmware Required for the Demonstration

To clone or download this application firmware on GitHub,

- Navigate to the [main page of this repository](#) and
- On the tab <> **Code**, above the list of files in the right-hand corner, click Code, then from the menu, click **Download ZIP** or copy the repository URL to **clone**.

Note:

In this document, hereinafter this firmware package is referred as **firmware**.

2.2 Software Tools Used for Testing the firmware

- MPLAB® X IDE **v6.25**
- Device Family Pack (DFP): **dsPIC33AK-MC_DFP v1.1.109**
- Curiosity/Starter Kits Tool Pack : **PKOB4_TP v1.19.1503**
- MPLAB® XC-DSC Compiler **v3.21**
- MPLAB® X IDE Plugin: **X2C-Scope v1.7.0**

Note:

The software used for testing the firmware prior to release is listed above. It is recommended to use these or later versions of the tool for building the firmware. All previous versions of Device Family Packs (DFP) and Tool Packs can be downloaded from [Microchip Packs Repository](#).

2.3 Hardware Tools Required for the Demonstration

- MCHV-SRM Development Platform
- 350V, 4 phase, 8/6 pole SRM.
- dsPIC33AK128MC106 Motor Control Dual In-line Module (DIM) ([EV68M17A](#))

Note:

- MCHV-SRM Development Platform is currently not available for purchase (contact Microchip sales representative for support).

- The SRM used here is a custom built motor, however one can implement this philosophy for other similar SRMs
- EV68M17A is available at [microchip DIRECT](#)

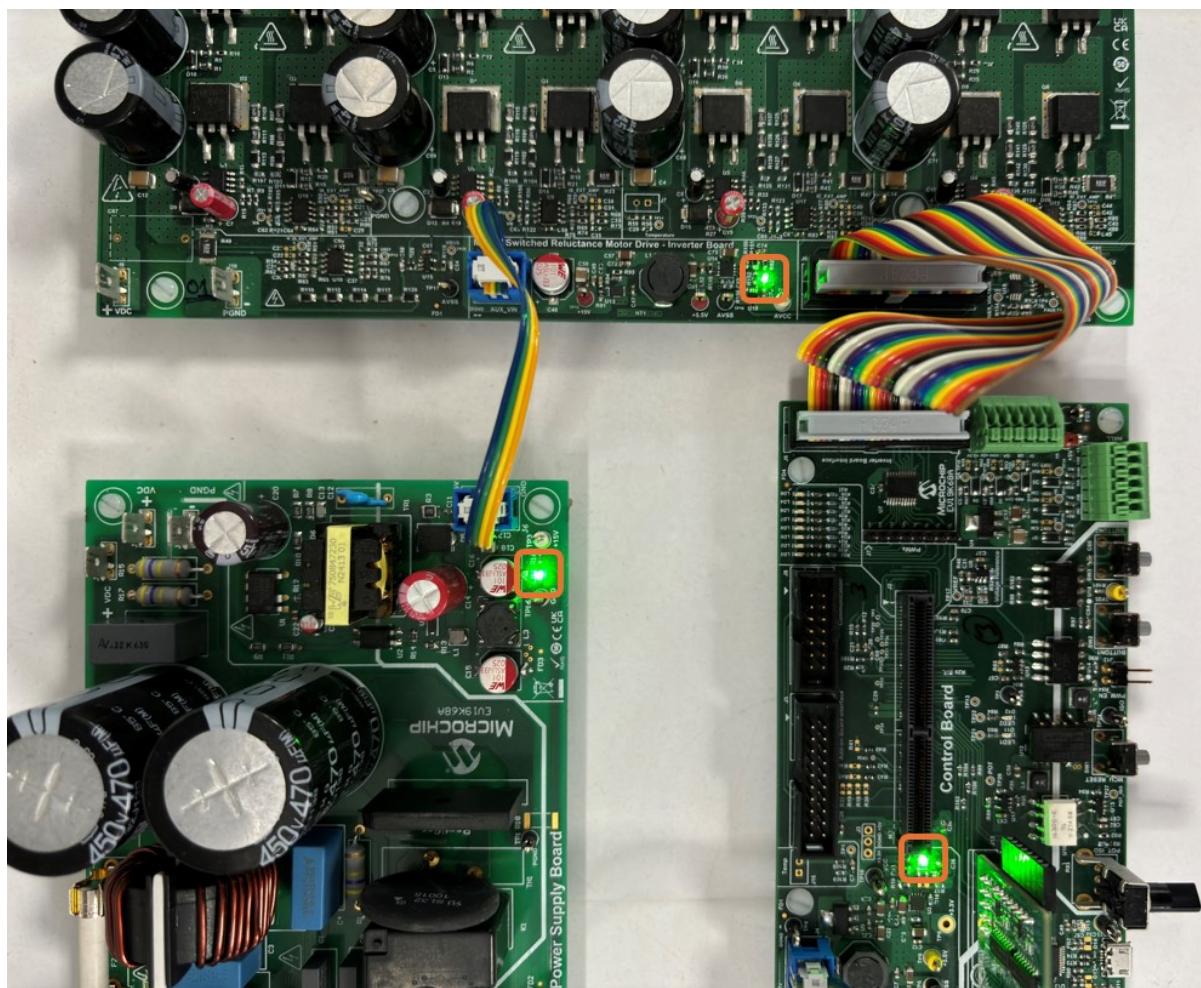
3. HARDWARE SETUP

This section describes the hardware setup required for the demonstration.

1. Motor currents are amplified on the MCHV-SRM Development Platform. The firmware requires DIM to be configured for external amplifier outputs ('**external op-amp configuration**') to measure the motor currents. **Table-1** summarizes the resistors to be populated and removed to convert the DIM from '**internal op-amp configuration**' to '**external op-amp configuration**'.

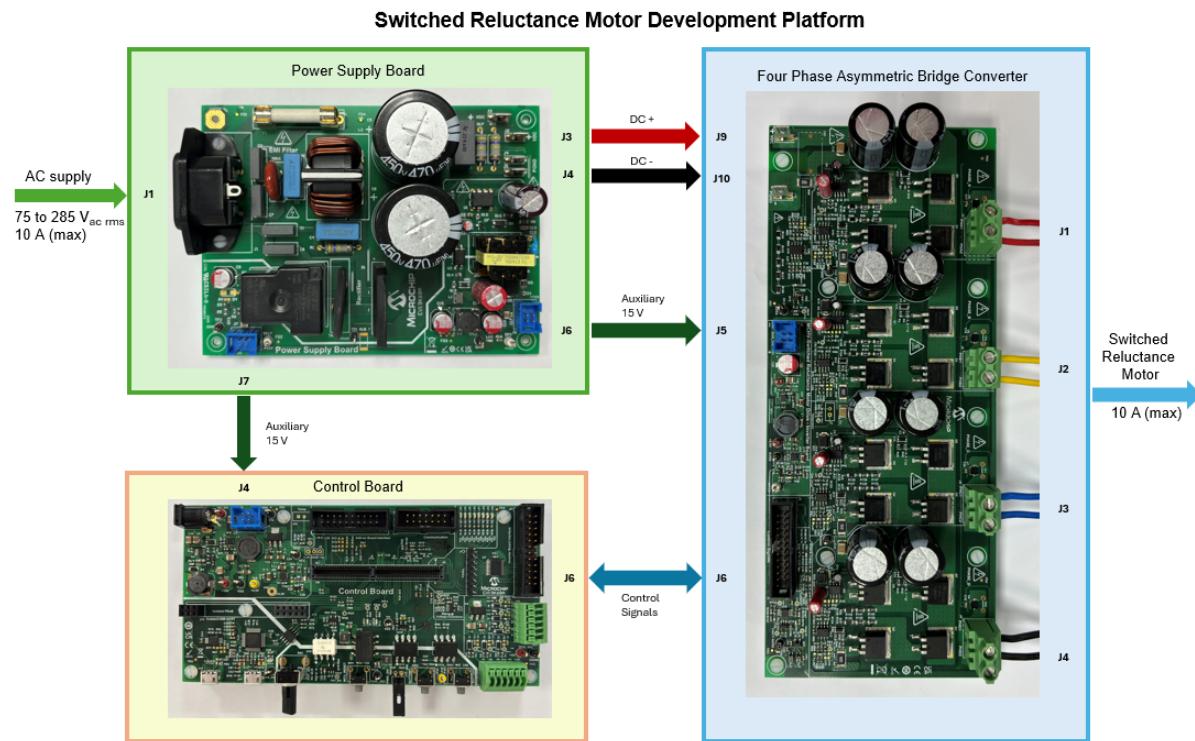
TABLE 1: SELECTION BETWEEN EXTERNAL AND INTERNAL AMPLIFIER OUTPUTS						
Current Signal	Internal Amplifier Output		External Amplifier Output		Firmware setting	
	Jumper Resistor (0R) Settings on the DIM					
	Populate	Remove	Populate	Remove		
Phase Current IA or IA_EXT	R6	R17	R17	R6	<ul style="list-style-type: none"> Configure and enable amplifiers in internal Op-Amp Configuration. Ensure to disable the internal amplifiers in external Op-Amp Configuration. 	
Phase Current IB or IB_EXT	R25	R30	R30	R25		
Bus Current IBUS or IBUS_EXT	R11	R9	R9	R11		

2. Ensure the development board is not powered and it is fully discharged. Verify the LEDs **LD1** (Green) on Power Supply Board, **LD1** (Green) on four phase asymmetric converter Board and **LD4** (Green) on Control Board are not glowing.

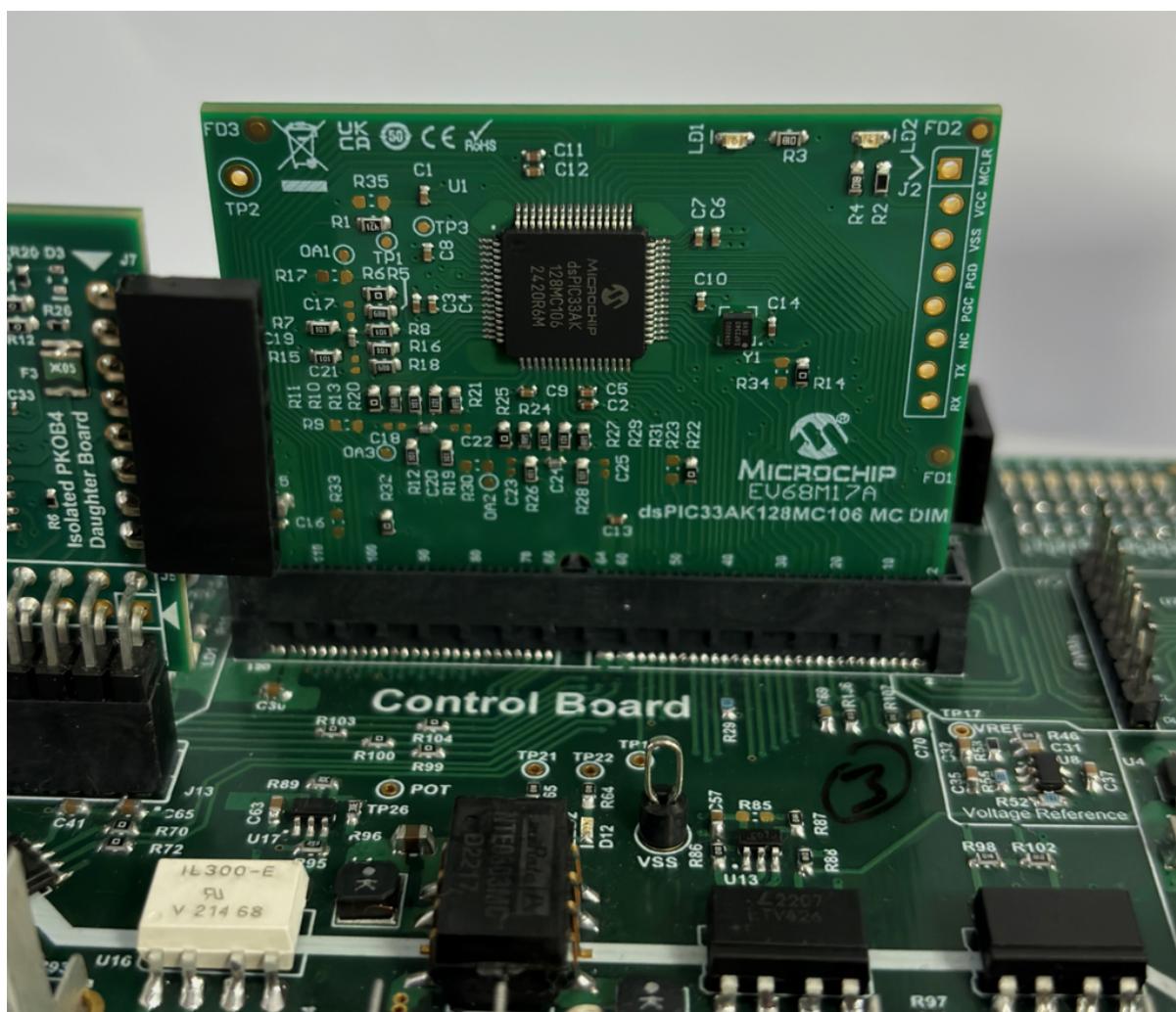


3. Please make the necessary connections as described and illustrated in the figure below:

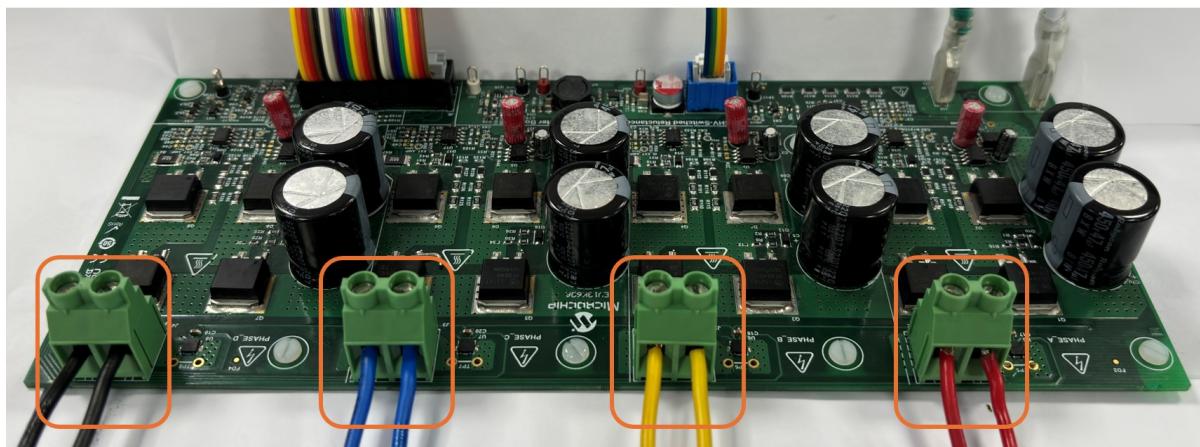
- Connect the AC supply (75 to 285 Vac) to the **J1** connector on the power supply board.
- Connect the DC bus positive terminal from **J3** on the power supply board to **J9** on the four-phase asymmetric converter board.
- Connect the DC bus negative terminal from **J4** on the power supply board to **J10** on the four-phase asymmetric converter board.
- Connect the 15 V auxiliary supply from **J6** on the power supply board to **J5** on the four-phase asymmetric converter board.
- Connect the 15 V auxiliary supply from **J7** on the power supply board to **J4** on the control board.
- Connect the control signals between **J6** on the control board and **J6** on the four-phase asymmetric converter board.
- Connect the motor phase connections to **J1**, **J2**, **J3**, and **J4** on the four-phase asymmetric converter board.



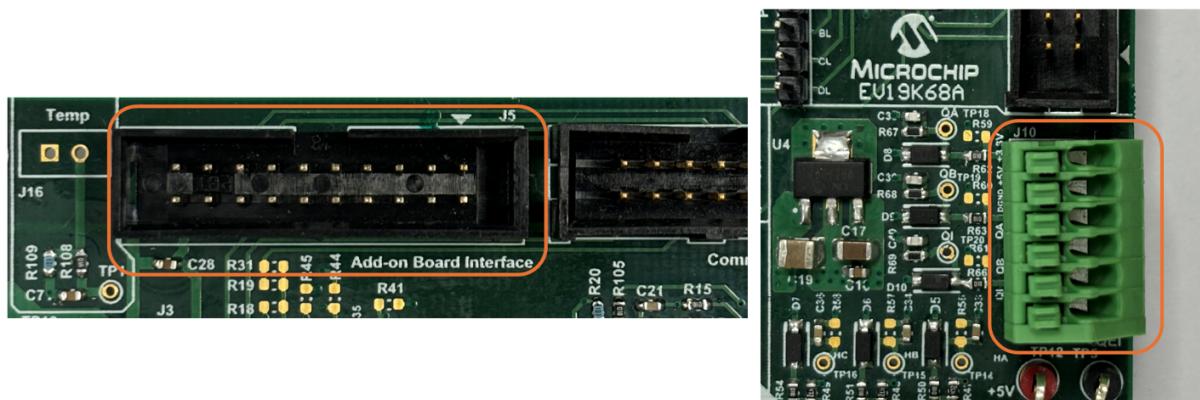
4. Insert the **dsPIC33AK128MC106 Motor Control DIM** into the DIM Interface **connector J2** on the control board. Make sure the DIM is placed correctly and oriented before going ahead.



5. Connect the 4-phase wires from the motor to **A**, **B**, **C**, and **D** of the connector J1, J2, J3 and J4 respectively (**MOTOR**) on the four phase asymmetric converter board.



6. Connect the rotor position sensor interface to the **J5** or **J10** connector control board.

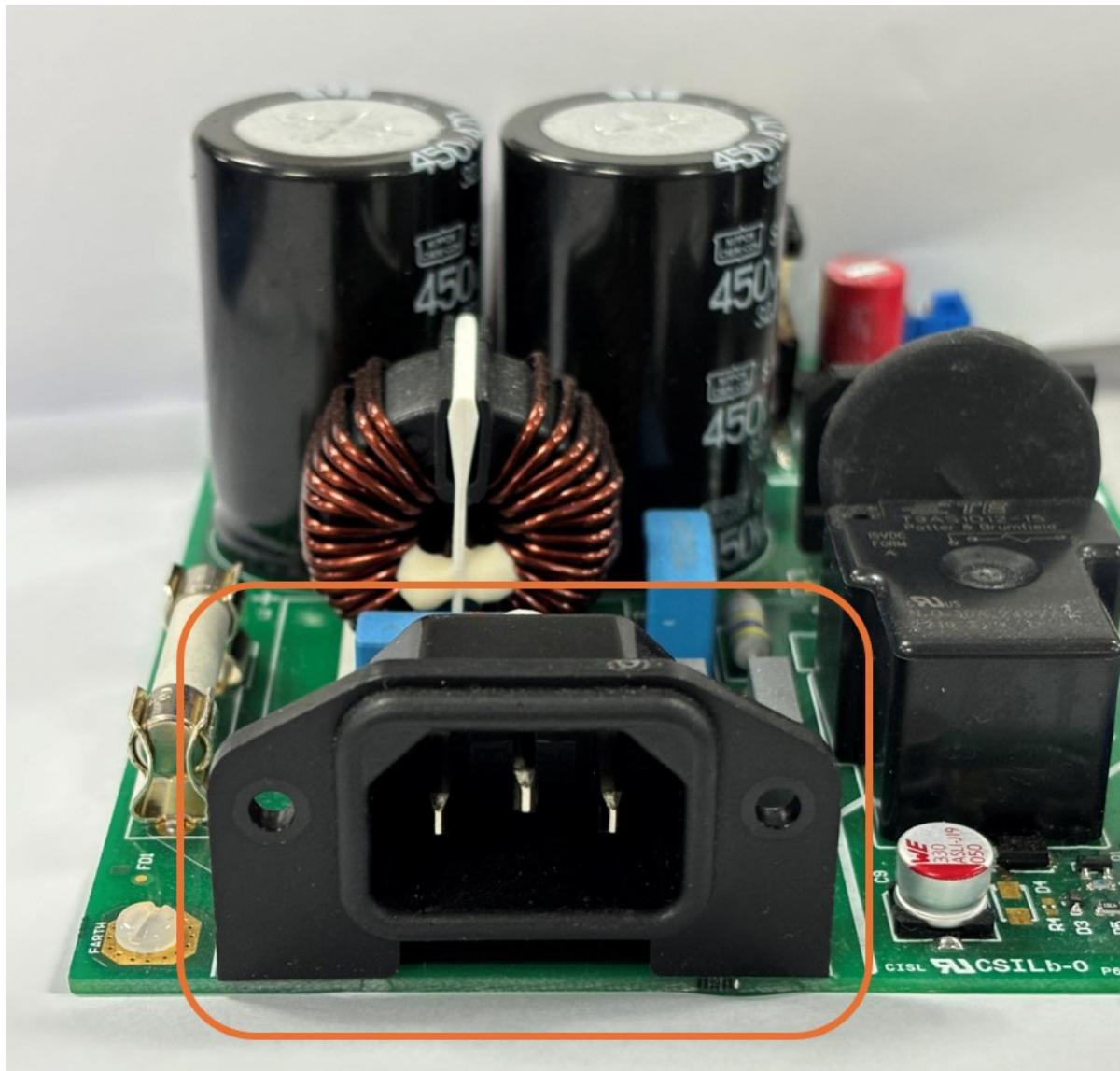


Note:

In this application AM4096 magnetic encoder is used to detect the rotor position. The AM4096 magnetic sensor board is connected to the **J5** connector connector on the control board.. The rotor position is determined using the Synchronous Serial Interface (SSI) between the controller and the sensor.



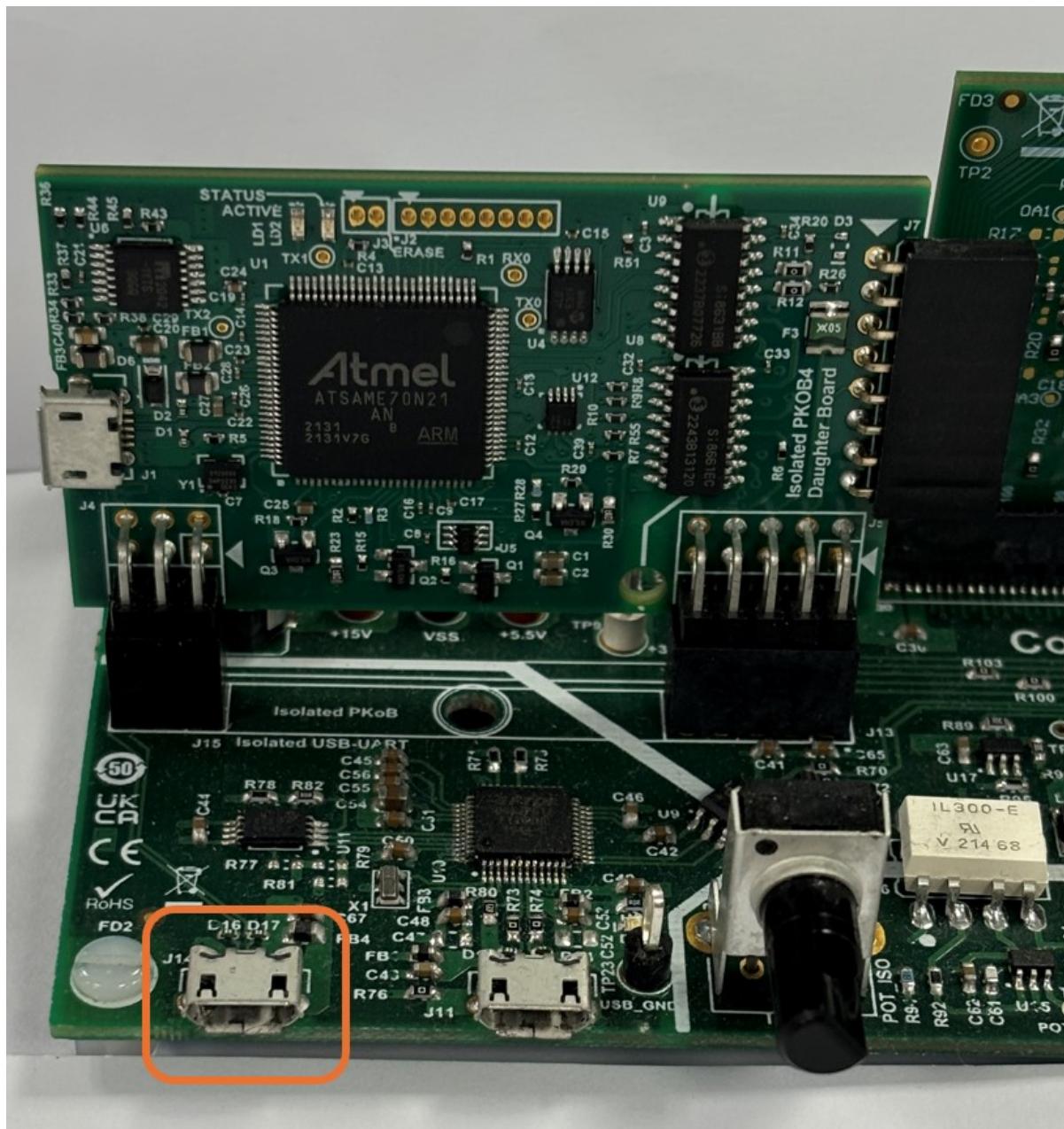
7. Power the development board from a controlled AC source by applying voltage of 220 V_{ac rms} through IEC connector **connector J1** provided on the Power board.



Note:

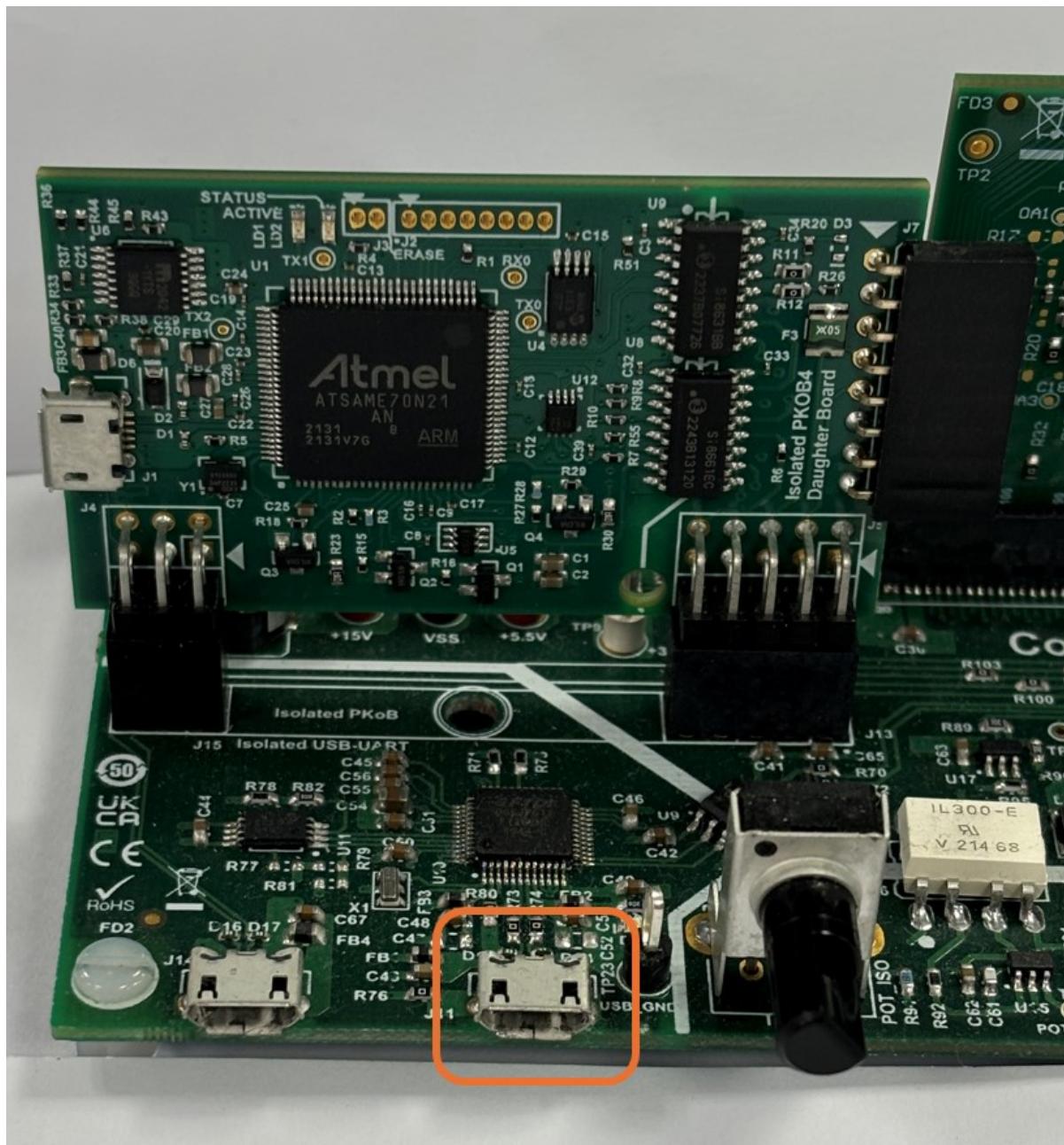
This Board is designed to operate in the 75 to 280 V_{ac rms} voltage range with a maximum input current of 10A_{rms}.

8. The development board has an isolated on-board programming tool called the Isolated PKoB4 Daughter Board. To use the isolated on-board programmer, connect a micro-USB cable between the Host PC and the connector J14 (**PROGRAM**) on the control board.

**Note:**

Use only **shielded micro-USB** cables intended for data transfer.

9. To establish serial communication with the host PC, connect a micro-USB cable between the host PC and the connector **J11 (USB-UART)** on the development board. This interface provides an isolated USB-UART communication.



Note:

Use only **shielded micro-USB** cables intended for data transfer.

4. SOFTWARE SETUP AND RUN

4.1 Setup: MPLAB X IDE and MPLAB XC-DSC Compiler

Install **MPLAB X IDE** and **MPLAB XC-DSC Compiler** versions that support the device **dsPIC33AK128MC106** and **PKoBv4**. The MPLAB X IDE, MPLAB XC-DSC Compiler, and X2C-Scope plug-in used for testing the firmware are mentioned in the [Motor Control Application Firmware Required for the Demonstration](#) section.

To get help on

- MPLAB X IDE installation, refer the [link](#)
- MPLAB XC-DSC Compiler installation steps, refer the [link](#)

If MPLAB IDE v8 or earlier is already installed on your computer, then run the MPLAB driver switcher (Installed when MPLAB® X IDE is installed) to switch from MPLAB IDE v8 drivers to MPLAB X IDE drivers. If you have

Windows 8 or 10, you must run the MPLAB driver switcher in **Administrator Mode**. To run the Device Driver Switcher GUI application as administrator, right-click on the executable (or desktop icon) and select **Run as Administrator**. For more details, refer to the MPLAB X IDE help topic “**Before You Begin: Install the USB Device Drivers (For Hardware Tools): USB Driver Installation for Windows Operating Systems.**”

4.2 Setup: X2C-SCOPE

X2C-Scope is a MPLAB X IDE plugin that allows developers to interact with an application while it runs. X2C-Scope enables you to read, write, and plot global variables (for motor control) in real-time. It communicates with the target using the UART. To use X2C-Scope, the plugin must be installed. To set up and use X2C-Scope, refer to the instructions provided on the [web page](#).

5. BASIC DEMONSTRATION

5.1 Firmware Description

The firmware version needed for the demonstration is mentioned in the section [Motor Control Application Firmware Required for the Demonstration](#) section. This firmware is implemented to work on Microchip’s Digital signal controller (dsPIC® DSC) **dsPIC33AK128MC106**. For more information, see the [dsPIC33AK128MC106 Family datasheet \(DS70005539\)](#).

The Motor Control Demo application uses a push button to start or stop the motor and a potentiometer to vary the speed of the motor. This Motor Control Demo Application configures and uses peripherals like PWM, ADC, UART, SPI etc.

Note:

The project may not build correctly in Windows OS if the Maximum path length of any source file in the project is more than 260 characters. In case the absolute path exceeds or nears the maximum length, do any (or both) of the following:

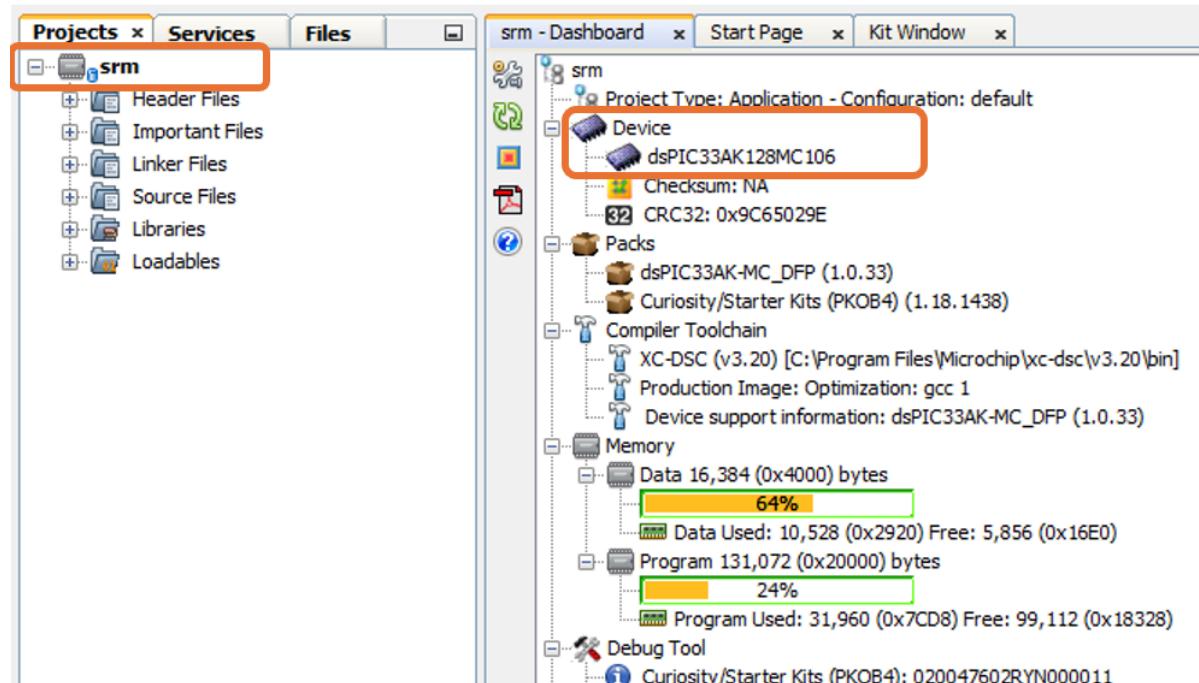
- Shorten the directory name containing the firmware used in this demonstration. If you renamed the directory, consider the new name while reading the instructions provided in the upcoming sections of the document.
- Place firmware in a location such that the total path length of each file included in the projects does not exceed the Maximum Path length specified.

Refer to MPLAB X IDE help topic “**Path, File, and Folder Name Restrictions**” for details.

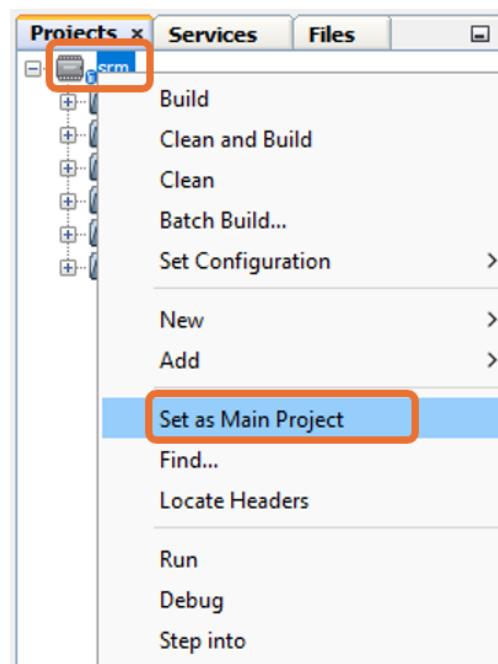
5.2 Basic Demonstration

Follow the below instructions, step by step, to set up and run the motor control demo application:

1. Start **MPLAB X IDE** and open the project **srm.X (File > Open Project)** with device selection **dsPIC33AK128MC106**.



2. Set the project **srm.X** as the main project by right-clicking on the project name and selecting **Set as Main Project** as shown. The project **srm.X** will then appear in **bold**.



3. Open **mc1_user_parms.h** (**srm.X > Header Files**) in the project **srm.X**.

- Ensure that the macros **SPEED_CONTROL** is defined for speed loop functioning and undefined for current loop functioning in the header file **mc1_user_parms.h**.

```
/** Define macros for operational modes */

/* Define SPEED_CONTROL to enable Speed control,
 * undefine SPEED_CONTROL to enable only Current control */
#define SPEED_CONTROL
```

- The macro **ALLEGRO_CT110_CS** is defined for magnetic sensor based current measurement.

```
/* Select sensor used for current measurement
 * Define ALLEGRO_CT110_CS for Allegro CT110 current sensor output
 * undefine ALLEGRO_CT110_CS for Shunt resistor current measurement */
#undef ALLEGRO_CT110_CS
```

Also, the changes to 0R jumper resistors should be done for source of current measurement in the inverter board as shown in the table below

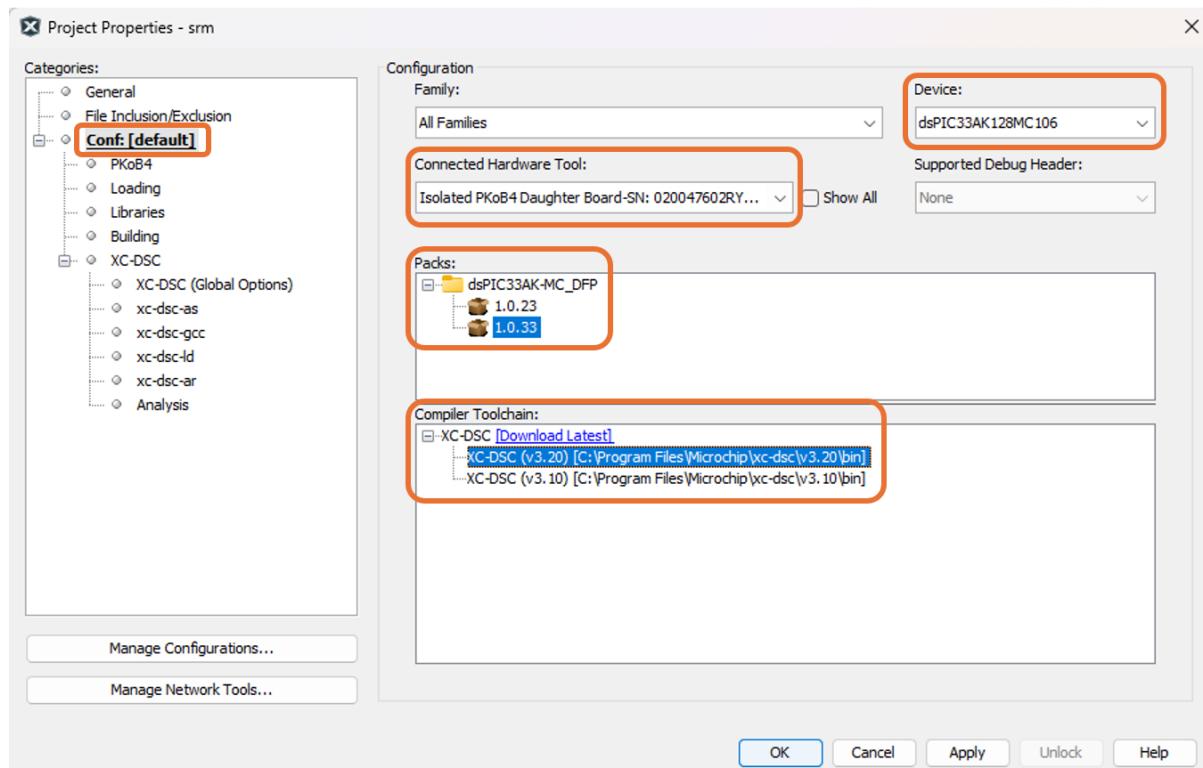
Table 2: SELECTION BETWEEN SHUNT RESISTOR OR CONTACT SENSOR CURRENT MEASUREMENT				
Current Signal	Shunt resistor current measurement		Contact sensor current measurement	
	Jumper resistor(0R) setting on Inverter board			
	Populate	Remove	Populate	Remove
Phase A Current IA_EXT_AMP or IA_EXT_CS	R158	R163	R163	R158
Phase B Current IB_EXT_AMP or IB_EXT_CS	R168	R171	R171	R168
Phase C Current IC_EXT_AMP or IC_EXT_CS	R159	R165	R165	R159
Phase D Current ID_EXT_AMP or ID_EXT_CS	R169	R173	R173	R169

- Right-click on the project **srm.X** and select **Properties** to open its **Project Properties** Dialog. Click the **Conf:[default]** category to reveal the general project configuration information. The development tools used for testing the firmware are listed in section [2.2 Software Tools Used for Testing the firmware](#).

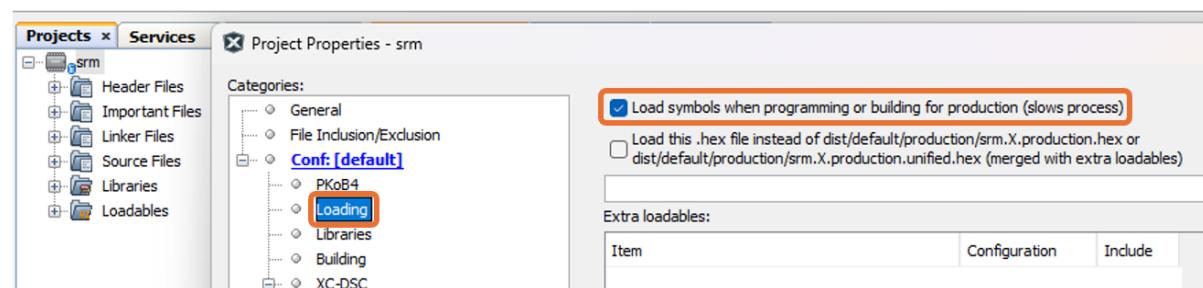
In the **Conf:[default]** category window:

- Ensure the selected **Device** is **dsPIC33AK128MC106**.
- Select the **Connected Hardware Tool** to be used for programming and debugging.
- Select the specific Device Family Pack (DFP) from the available list of **Packs**. In this case, **dsPIC33AK-MC_DFP 1.1.109** is selected.
- Select the specific **Compiler Toolchain** from the available list of **XC-DSC** compilers. In this case, **XC-DSC(v3.21)** is selected.
- After selecting Hardware Tool and Compiler Toolchain, Device Pack, click the button **Apply**

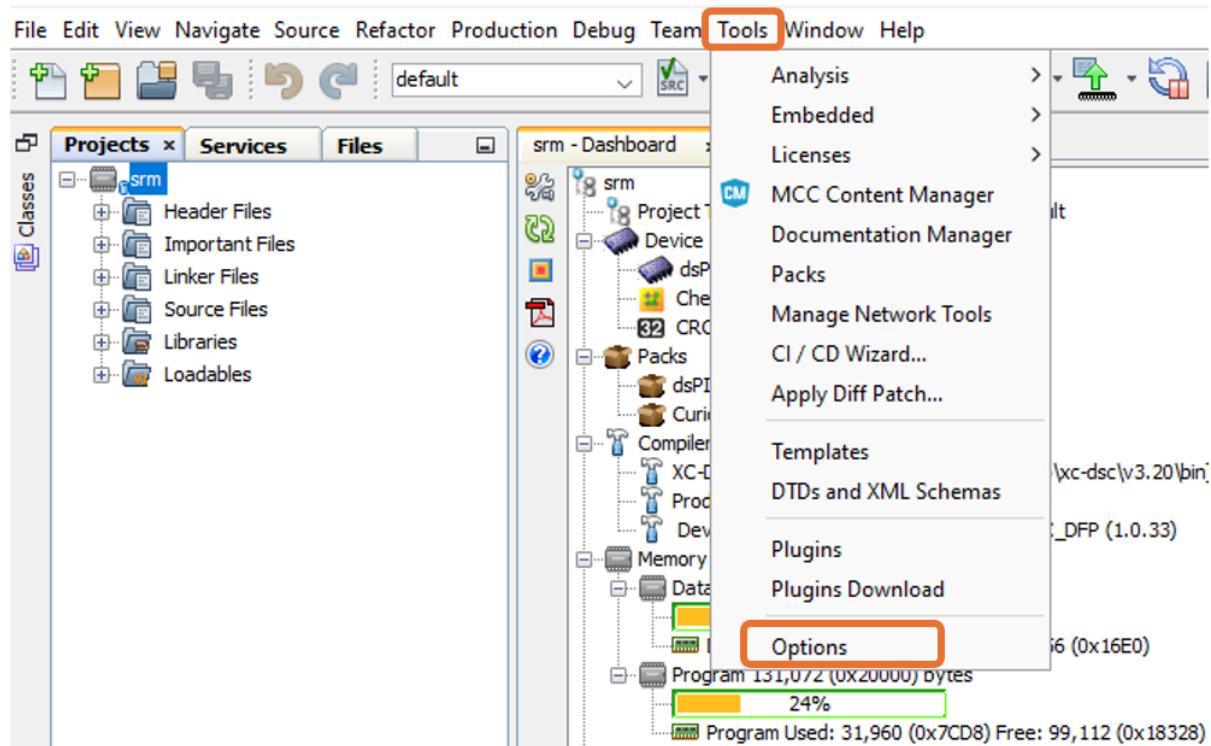
Please ensure that the selected MPLAB® XC-DSC Compiler and Device Pack support the device configured in the firmware



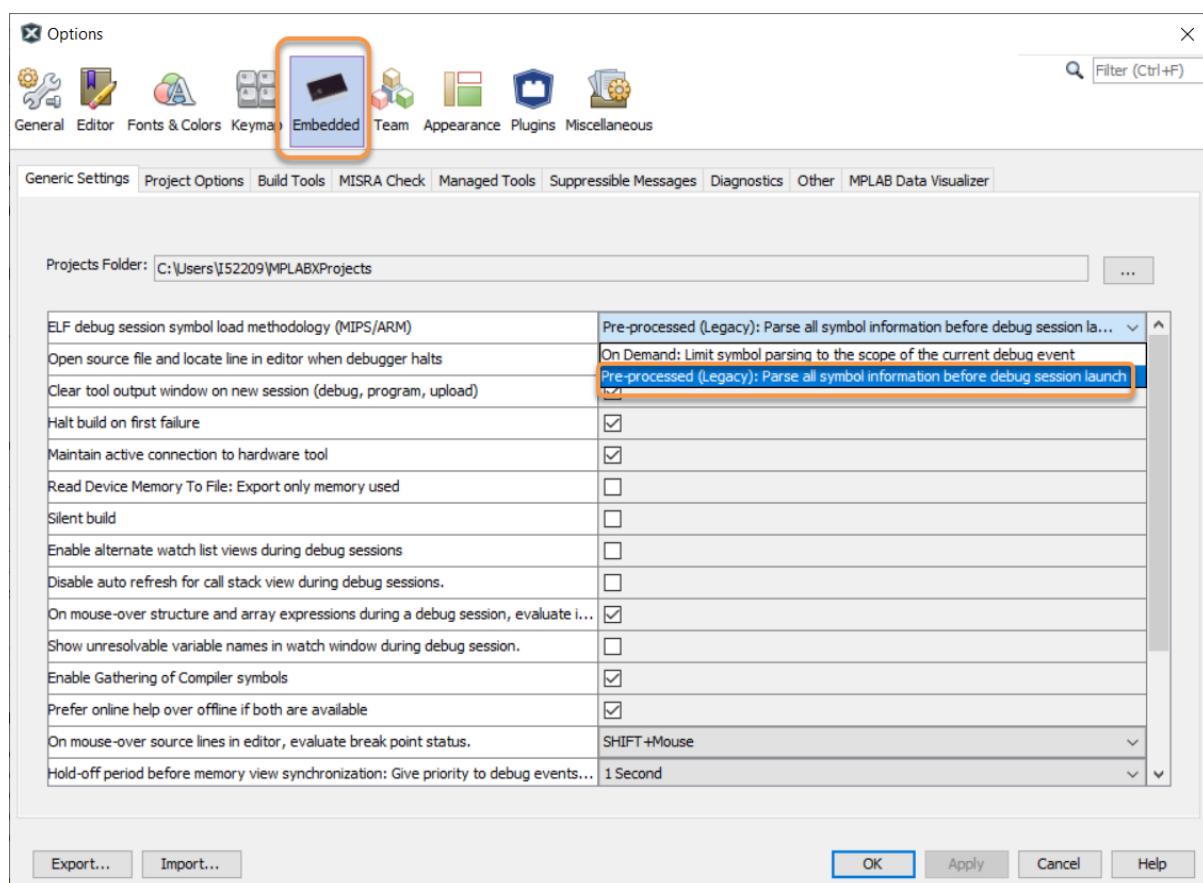
5. Ensure that the checkbox **Load symbols when programming or building for production (slows process)** is checked under the **Loading** category of the **Project Properties** window.



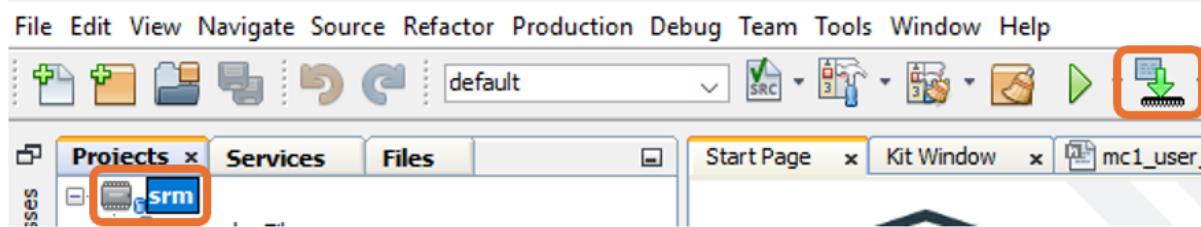
Also, go to **Tools > Options**, and



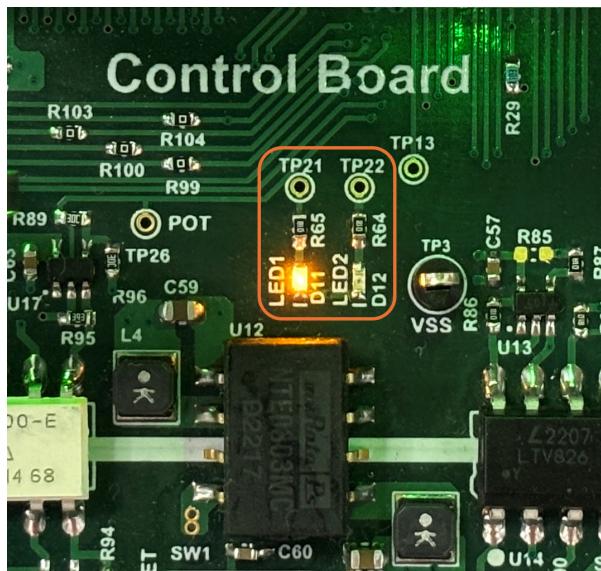
Open the **Embedded > Generic Settings** tab and ensure that the **ELF debug session symbol load methodology (MIPS/ARM)** is selected as **Pre-procesed (Legacy)** from the drop down.



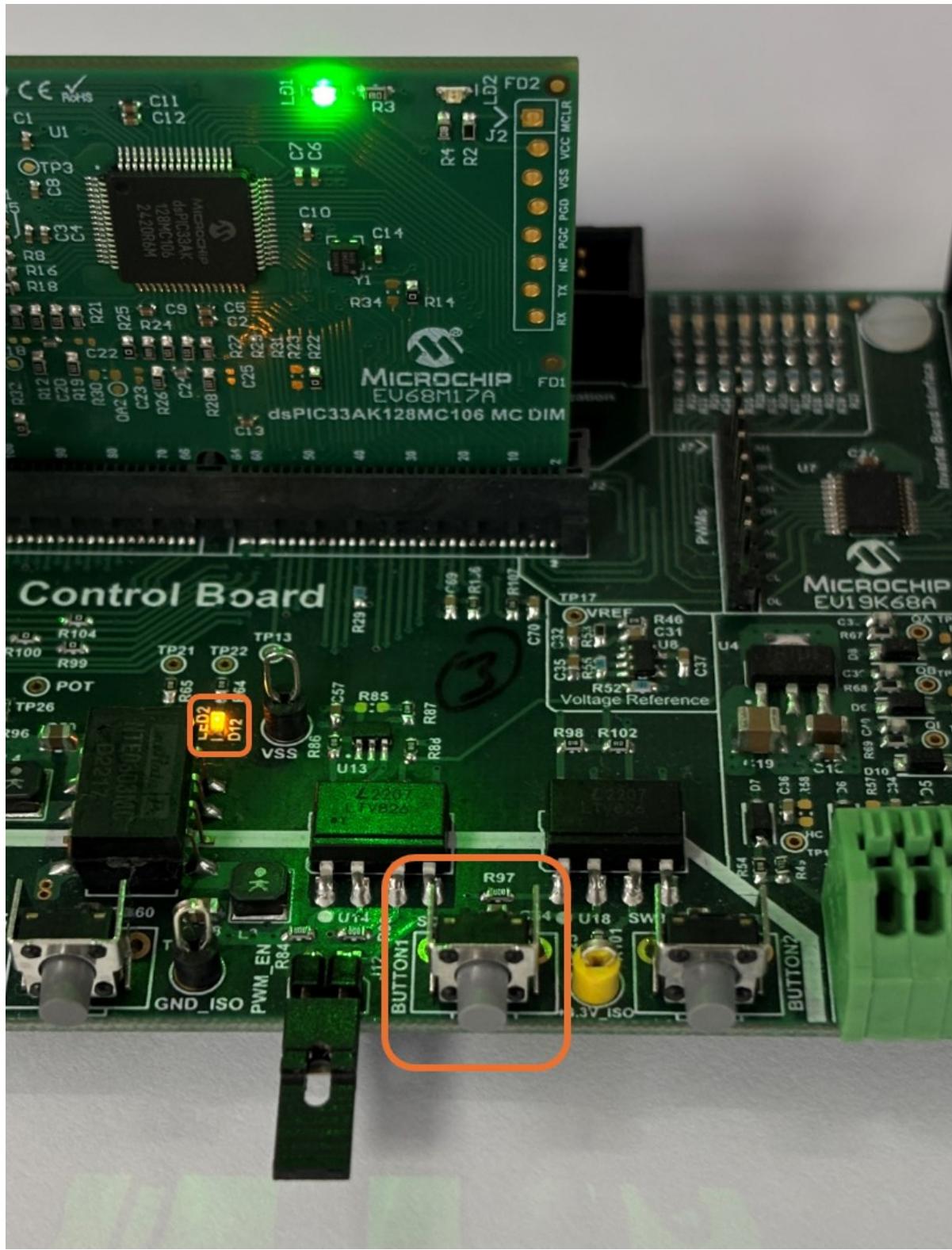
6. To build the project (in this case, **pmsm.X**) and program the device dsPIC33AK128MC106, click **Make and Program Device Main project** on the toolbar



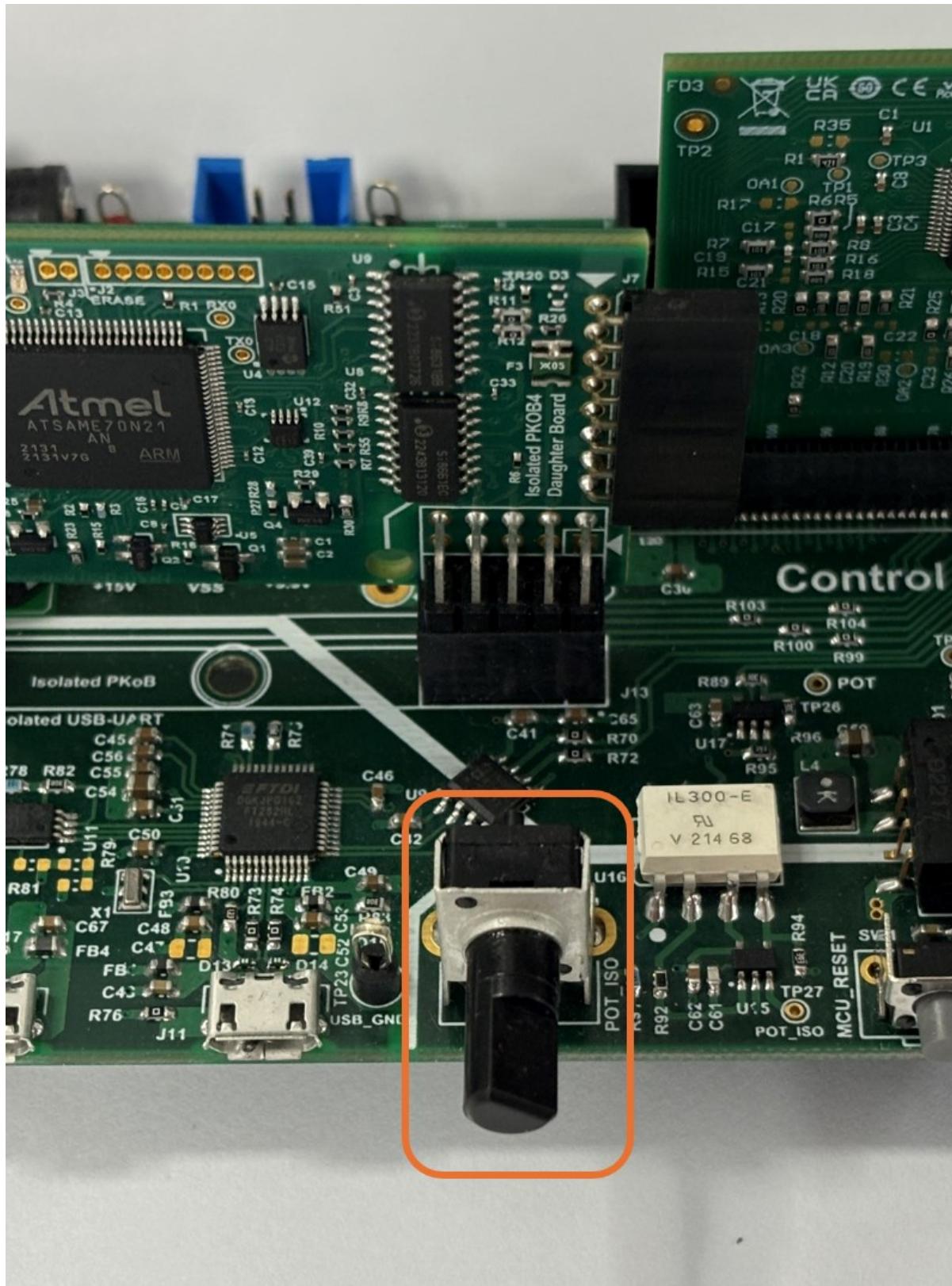
7. If the device is successfully programmed, **LED1 (D11)** will be **blinking**, indicating that the dsPIC® DSC is enabled.



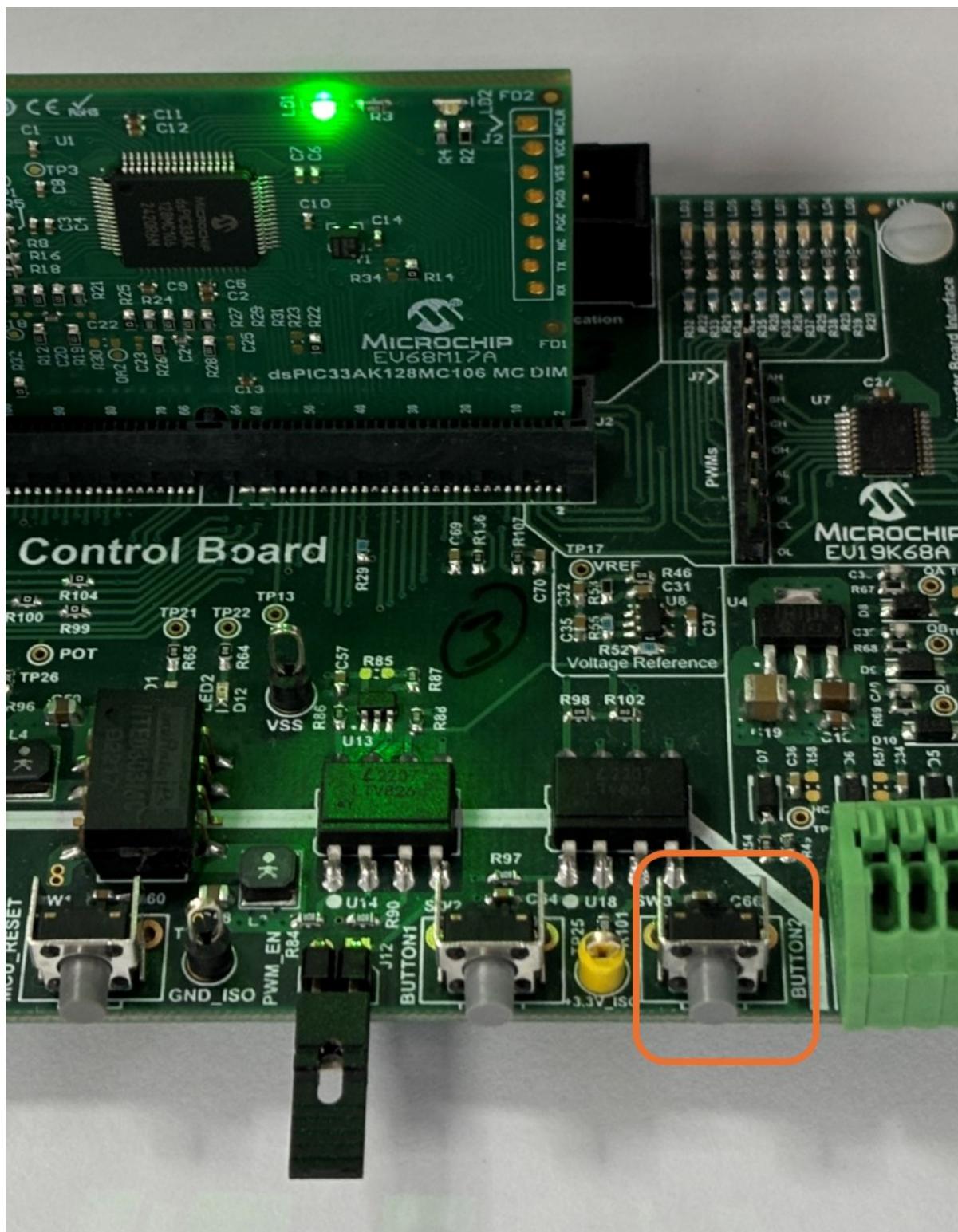
8. Run or stop the motor by pressing the push button **BUTTON 1**. The motor should start spinning smoothly in one direction. The **LED2 (D12)** is turned **ON** to show the button is pressed to start the motor.



9. The motor speed can be varied using the potentiometer (**POT**).



10. Press the push button **BUTTON 2** to change the direction of rotation.



11. Press the push button **BUTTON 1** to stop the motor.

Note:

In the header file `mc1_user_parms.h` included in the project `srm.X`.

- The macros `RATED_CURRENT`, `MINIMUM_SPEED_RPM`, `NOMINAL_SPEED_RPM`, `MAXIMUM_SPEED_RPM`, `MOTOR_MIN_DC_VOLT`, and `MOTOR_MAX_DC_VOLT` are defined as per the Motor manufacturer's specifications. Exceeding manufacture specifications may damage the motor or the board or both.
- In this firmware commutation of phases is defined in the macro `THETA_X_COMMUTATE_CW`, the turn-on and turn-off angles should be defined in the macros `THETA_X_ON_CW` and

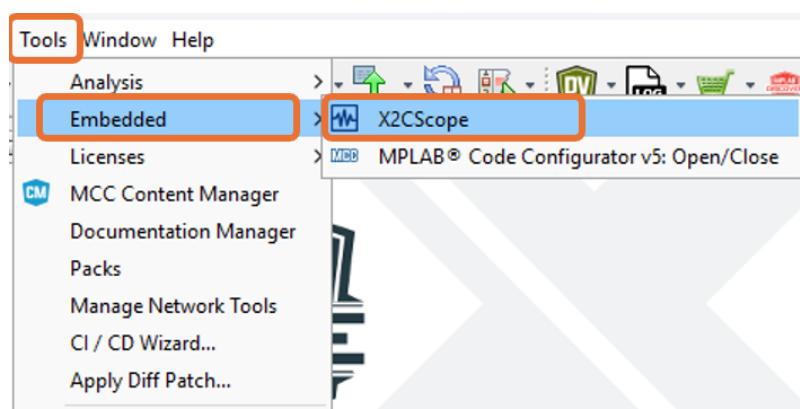
`THETA_X_OFF_CW` respectively. The bootstrap capacitor to be charged should be defined in the macro `THETA_X_CBOOT_CW`. Here "X" is the number of phases, also there are different macros defined for clockwise and counter clockwise direction of rotation.

- The macro `PHASE_OC_THRESHOLD` is defined as a threshold for software phase over current fault detection.

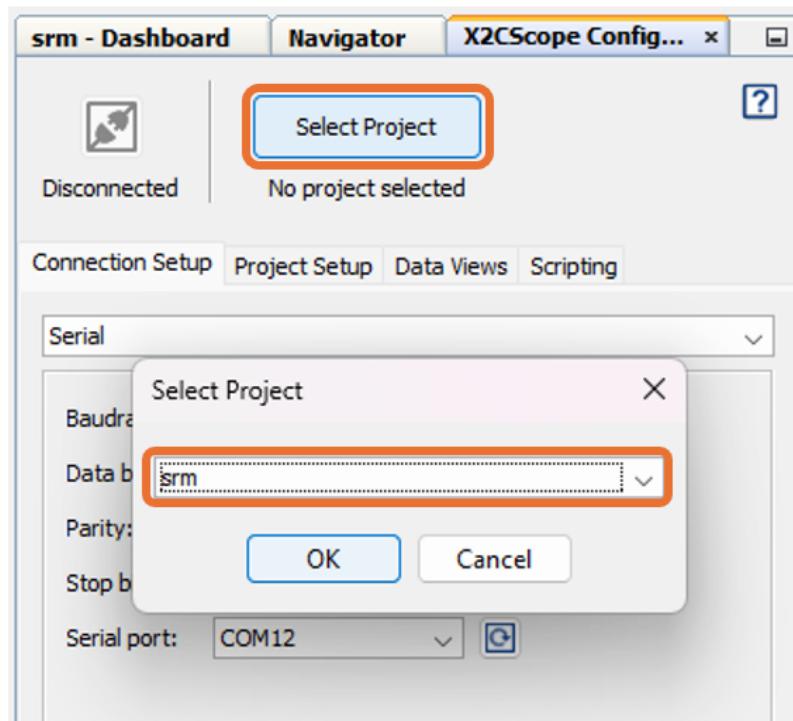
5.3 Data visualization through X2C-Scope Plug-in of MPLAB X

X2C-Scope is a third-party plug-in in MPLAB X, which helps in real-time diagnostics. The application firmware comes with the initialization needed to interface the controller with the host PC to enable data visualization through the X2C-Scope plug-in. Ensure the X2C-Scope plug-in is installed. For more information on how to set up a plug-in, refer to either the [Microchip Developer Help page](#) or the [web page](#).

1. To establish serial communication with the host PC, connect a micro-USB cable between the host PC and connector J11 (**USB-UART**) on the development board. This interface provides an isolated USB-UART communication.
2. Ensure the application is configured and running as described under section [5.2 Basic Demonstration](#) by following steps 1 through 11.
3. Open the **X2C-Scope** window by selecting **Tools>Embedded>X2CScope**.

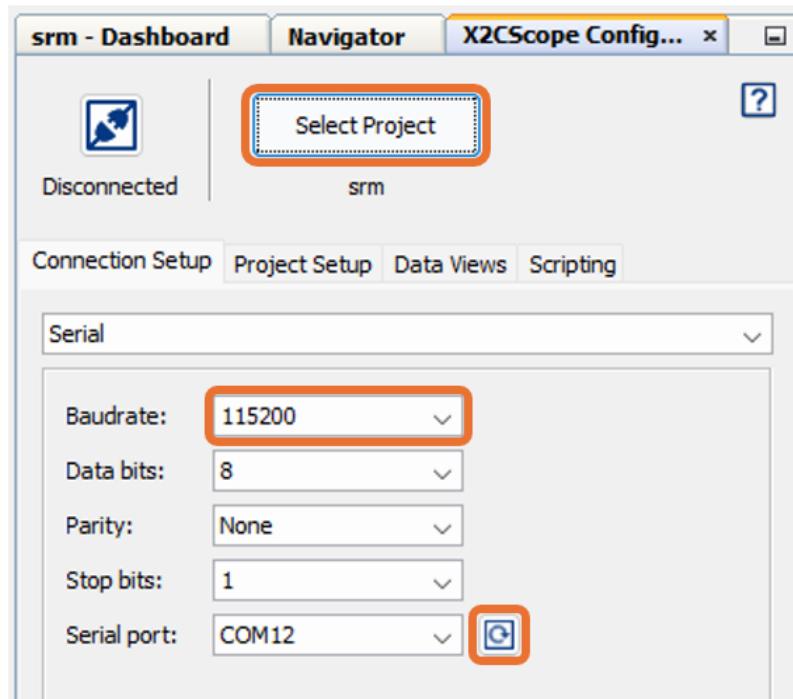


4. In the **X2C-Scope Configuration** window, open the **Connection Setup** tab and click **Select Project**. This opens the drop-down menu **Select Project** with a list of opened projects. Select the specific project **srm** from the list of projects and click **OK**.

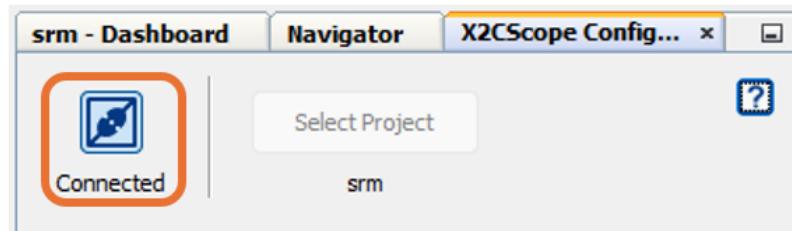


5. To configure and establish the serial communication for **X2C-Scope**, open the **X2CScope Configuration** window, click on the **Connection Setup** tab and:

- Set **Baudrate** as **115200**, which is configured in the application firmware.
- Click on the **Refresh** button to refresh and update the list of the available Serial COM ports connected to the Host PC.
- Select the specific **Serial port** detected when interfaced with the development board. The **Serial port** depends on the system settings

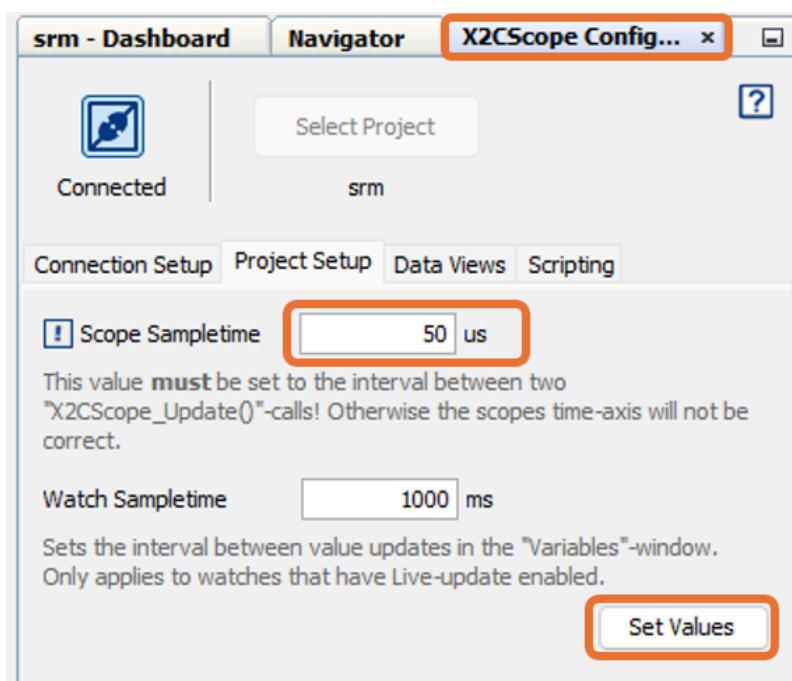


6. Once the **Serial port** is detected, click on **Disconnected** and turn to **Connected**, to establish serial communication between the Host PC and the board.

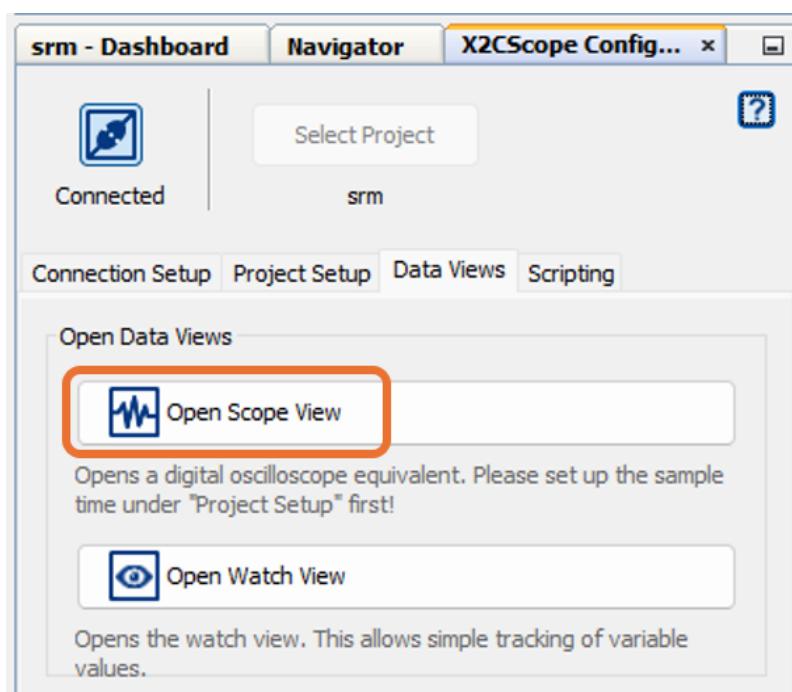


7. Open the **Project Setup** tab in the **X2CScope Configuration** window and,

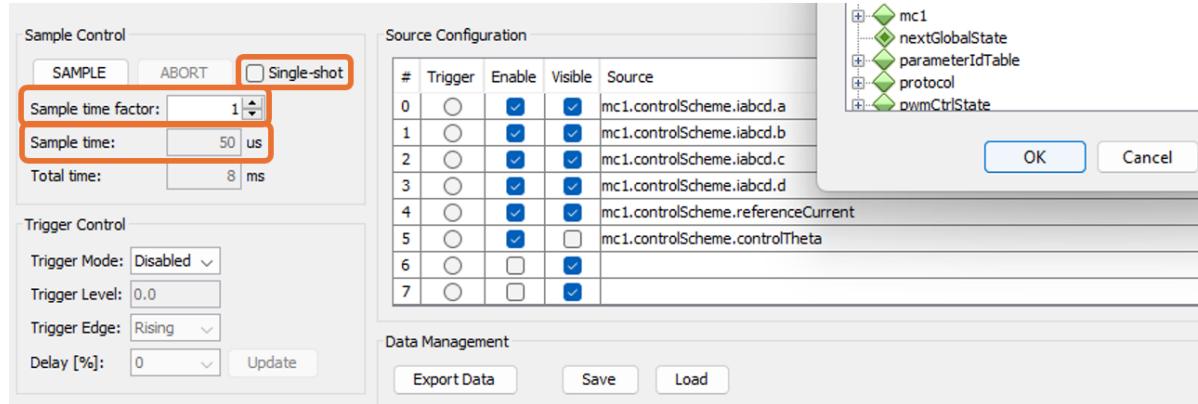
- Set **Scope Sampletime** as the interval at which `X2CScopeUpdate()` is called. In this application, it is every `50μs`.
- Then, click **Set Values** to save the configuration.



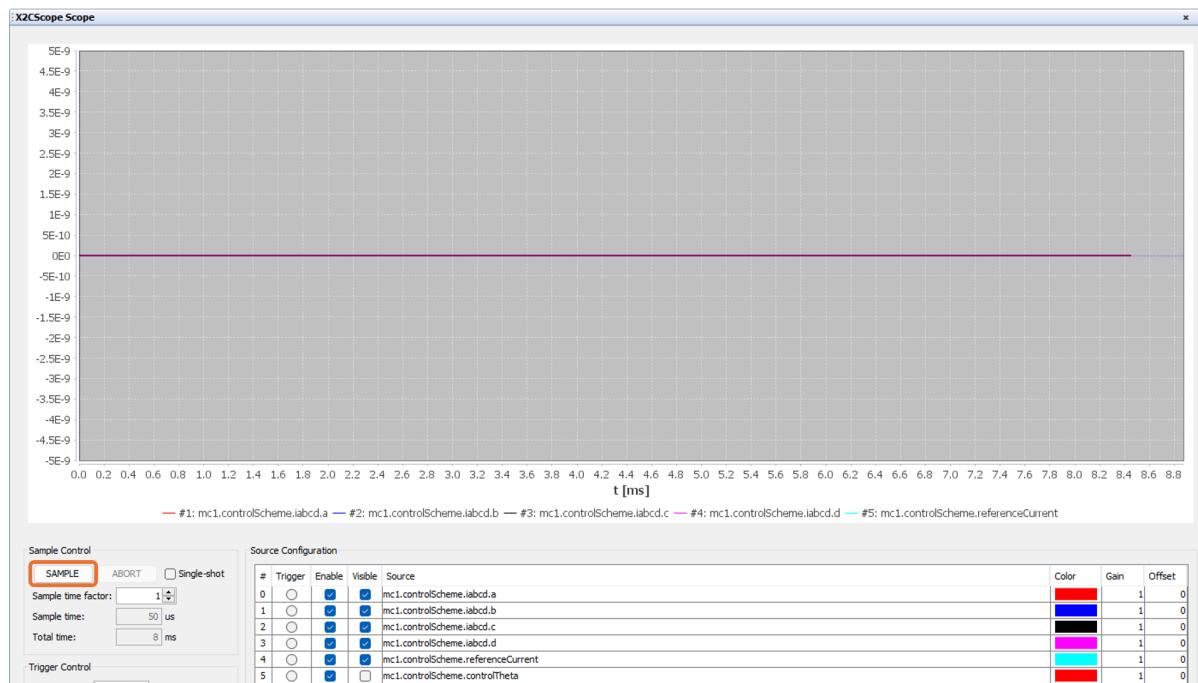
8. Click on **Open Scope View** (in the **Data Views** tab of the **X2CScope Configuration Window**); this opens **Scope Window**.



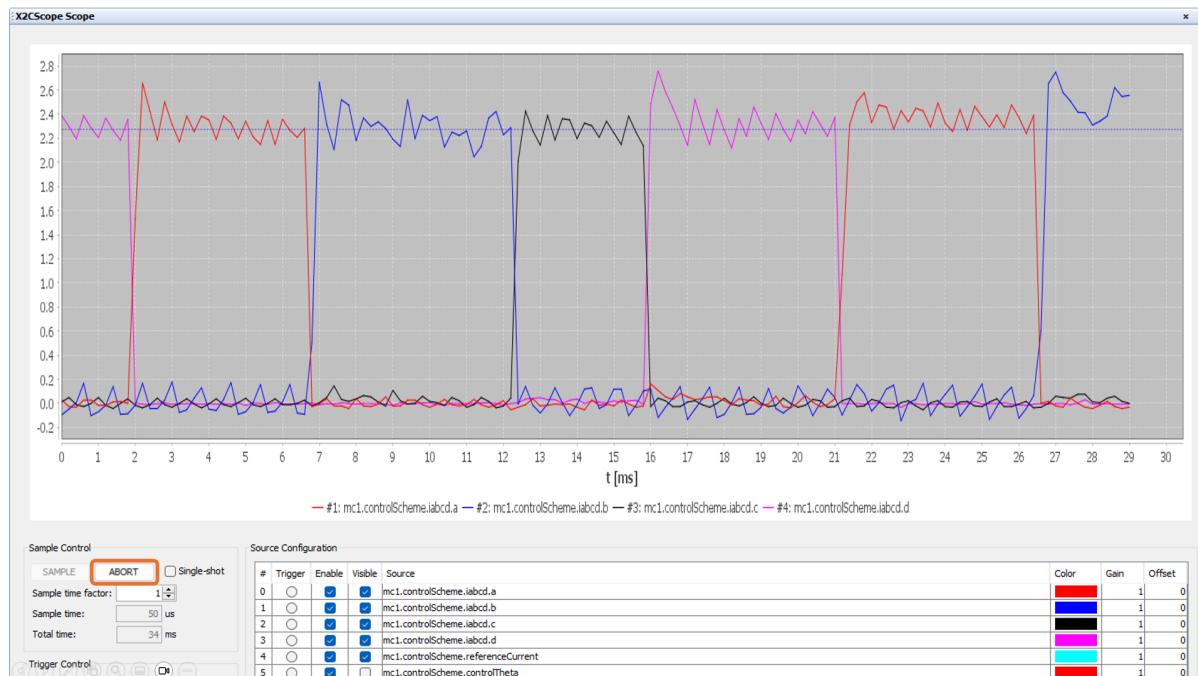
9. In the **Scope Window**, select the variables that must be watched. To do this, click on the **Source** against each channel, and a window **Select Variables** opens on the screen. From the available list, the required variable can be chosen. Ensure checkboxes **Enable** and **Visible** are checked for the variables to be plotted. To view data plots continuously, uncheck **Single-shot**. When **Single-shot** is checked, it captures the data once and stops. The **Sample time factor** value multiplied by **Sample time** decides the time difference between any two consecutive data points on the plot.



10. Click on **SAMPLE**, then the X2C-Scope window plots variables in real-time, which updates automatically.



11. Click on **ABORT** to stop.



6. REFERENCES:

For additional information, refer following documents or links.

1. dsPIC33AK128MC106 Motor Control Dual In-Line Module (DIM) Information Sheet ([DS70005527](#))
2. dsPIC33AK128MC106 Family datasheet ([DS70005539](#))
3. MPLAB® X IDE User's Guide ([DS50002027](#)) or MPLAB® X IDE help
4. [MPLAB® X IDE installation](#)
5. [MPLAB® XC-DSC Compiler installation](#)
6. [Installation and setup of X2Cscope plugin for MPLAB X](#)
7. [Microchip Packs Repository](#)