AN1292 Demonstration ReadMe for the MCLV-48V-300W Inverter Board with the dsPIC33CH512MP508 Motor Control Dual In-line Module (MPLAB® X IDE)

1. INTRODUCTION

This document describes the setup requirements for running the Sensorless FOC algorithm with a PLL Estimator, which is referenced in AN1292 "*Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)*" using MCLV-48V-300W Inverter Board and dsPIC33CH512MP508 Motor Control Dual In-line Module (DIM).

2. SUGGESTED DEMONSTRATION REQUIREMENTS

2.1. Motor Control Application Firmware Required for the Demonstration

To clone or download this application from Github, go to the main page of this repository and then click Clone button to clone this repository or download as zip file.

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

2.2. Software Tools Used for Testing the firmware

- MPLAB X v5.50 or later
- DFP: dsPIC33CH-MP_DFP v1.9.207 or later
- MPLAB® XC16 Compiler v1.70 or later
- MPLAB® X IDE Plugin: X2C-Scope v1.3.0 or later

> **Note:**
>
> The software used for testing the firmware prior to release is listed above. It is recommended to use the version listed above or later versions for building the firmware.

2.3. Hardware Tools Required for the Demonstration

- MCLV-48V-300W Inverter Board (EV18H47A)
- dsPIC33CH512MP508 Motor Control DIM (EV76L31A)
- 24V Power Supply (AC002013)
- 24V 3-Phase Brushless DC Motor (AC300020)

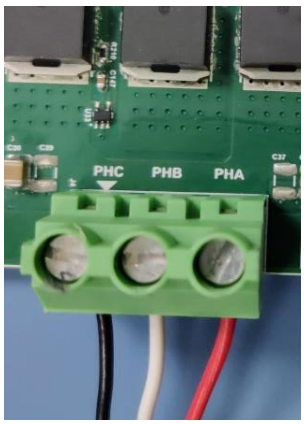
> **Note:**
>
> All items listed under this section MPLAB X v5.50 or later
>
> DFP: dsPIC33CH-MP_DFP v1.9.207 or later

3. HARDWARE SETUP

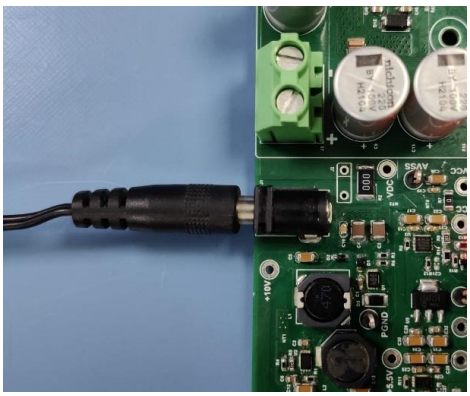
This section describes the hardware setup needed for the demonstration.

1. Motor currents are amplified by the amplifiers (U10 and U11) on the MCLV-48V-300W Inverter Board. The firmware and DIM are configured to sample and convert external amplifier outputs to measure the motor currents needed to implement FOC.
2. Insert the dsPIC33CH512MP508 Motor Control DIM into the DIM Interface Connector J8 provided on the MCLV-48V-300W Inverter Board. Make sure the DIM is placed correctly and oriented before going ahead.

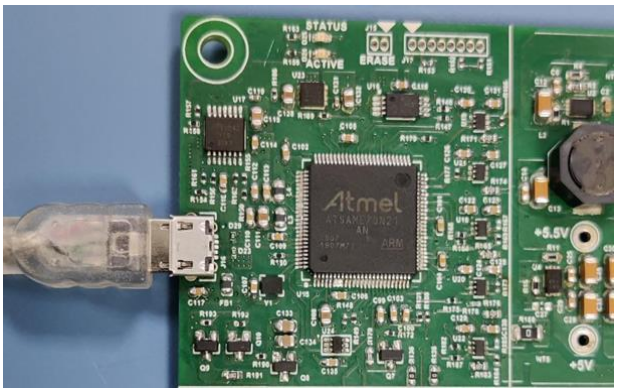
3. Connect the 3-phase wires from the motor to PHC, PHB, and PHA of the J4 connector (no specific order), provided on the MCLV-48V-300W Inverter Board.



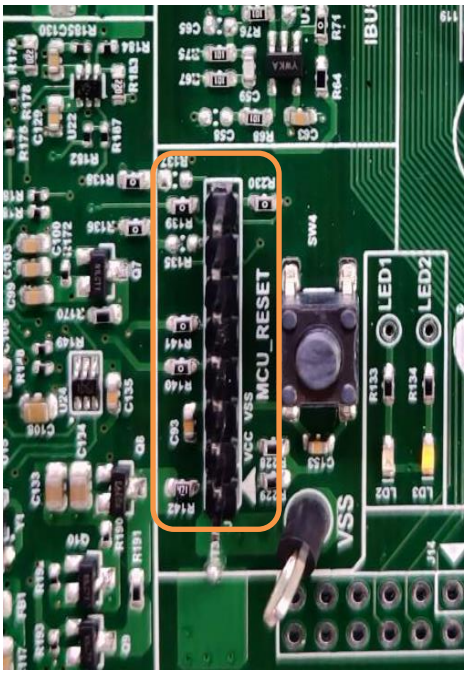
4. Plug in the 24V power supply to connector J1 provided on the MCLV-48V-300W Inverter Board. Alternatively, the Inverter Board can also be powered through Connector J2.



5. The board has an onboard programmer ‘PICKIT™ On Board (PKOBv4)”, which can be used for programming or debugging the dsPIC33CH512MP508. To use an on-board programmer, connect a micro-USB cable between Host PC and Connector J16 provided on the MCLV-48V-300W Inverter Board.



6. Alternatively, connect the Microchip programmer/debugger MPLAB PICkit 4 In-Circuit Debugger to the ICSP header J9 of the MCLV-48V-300W Inverter Board as shown below and to the Host PC used for programming the device. Ensure that PICkit 4 is connected in the correct orientation.



4. SOFTWARE SETUP AND RUN

4.1. Setup: MPLAB X IDE and MPLAB XC16 Compiler

Install MPLAB X IDE and MPLAB XC16 Compiler versions that support the device dsPIC33CH512MP508 and PKOBv4. The tools like MPLAB X IDE, MPLAB XC16 Compiler, and X2C-Scope plug-in used for testing the firmware are mentioned in the Motor Control Application Firmware Required for the Demonstration.

To get help on

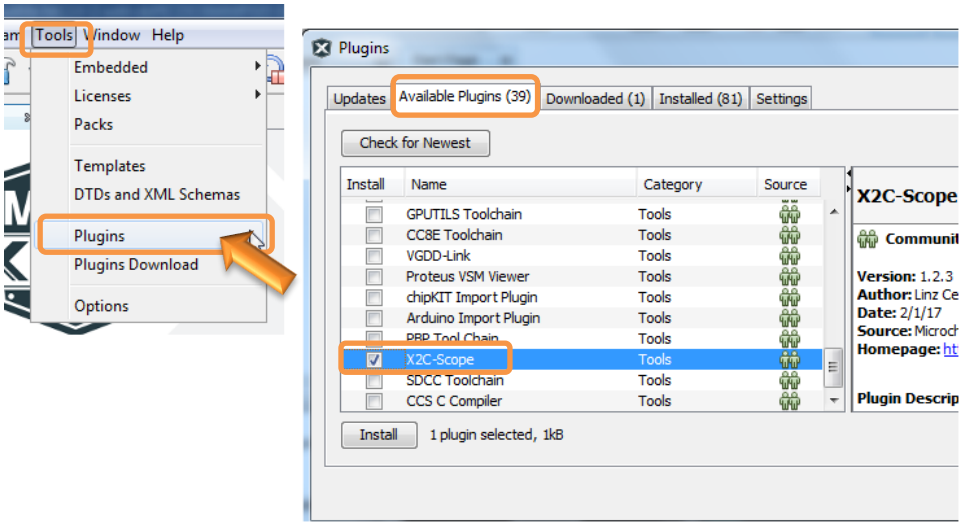
- MPLAB X IDE installation, refer to link
- MPLAB XC16 Compiler installation steps, refer to link

If MPLAB IDE v8 or earlier is already installed on your computer, then run the MPLAB driver switcher (It is installed when MPLAB®X IDE is installed) to switch from MPLAB IDE v8 drivers to MPLAB X IDE drivers. If you have Windows 7 or 8, you must run 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 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 an MPLAB X IDE plugin that allows a developer to interact with an application while the application program is running. X2C-Scope enables you to read, write, and plot global variables in real-time. It communicates with the target using the UART. To use X2C-Scope, the plugin must be installed:

- In MPLAB X IDE, select *Tools->Plugins* and click on the Available Plugins tab.
- Select X2C-Scope plug-in by checking its check box and clicking Install.
- Look for tool X2C-Scope under *Tools->Embedded*.



5. BASIC DEMONSTRATION

5.1. Firmware Description

The firmware version required for the demonstration is mentioned under the "Motor Control Application Firmware Required for the Demonstration'.

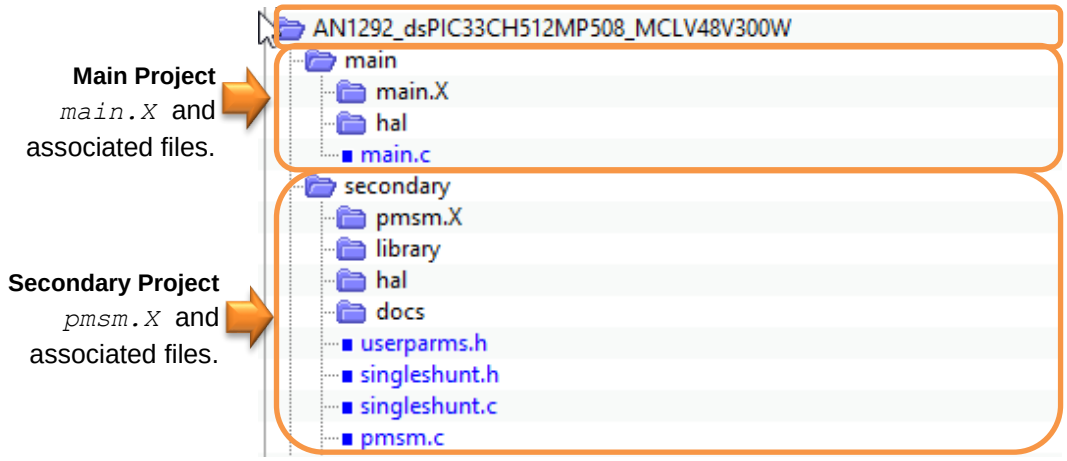
This firmware is implemented to work on Microchip's dual-core 16-bit Digital signal controller (dsPIC® DSC) dsPIC33CH512MP508**.**There are two independent dsPIC DSC cores called 'Main Core' and 'Secondary Core' in the device. For more information, see the *dsPIC33CH512MP508 Family datasheet*.

In MPLAB X IDE, the code for two cores is developed as separate projects with the following device selections.

- Device selection in Main Project (code for Main Core) is dsPIC33CH512MP508
- Device selection in Secondary Project (code for Secondary Core) is dsPIC33CH512MP508S1

Hence the firmware used in this demonstration consists of two MPLAB X projects, `main.X` (Main Project) and `pmsm.X` (Secondary Project).

The firmware directory structure is shown below:



Note:

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

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

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

The function of the Main Core is defined by the Main Project *main.X*. They are:

- To set device configuration bits applicable for both Main and Secondary cores (Configuration bits for Main and Secondary cores exist in Main core). Note that the configuration bits decide the I/O port ownership between Main Core and Secondary Core.
- Configure Main Core Oscillator Subsystem to generate clocks needed to operate Core and its peripherals. In the firmware, Main is configured to use at 90MHz.
- To program and enable the Secondary core by invoking XC16 library (libpic30.h) routines `_program_secondary()` and `_start_secondary()`.

The function of the Secondary Core (as defined in the Secondary Project *pmsm.X*) is:

- To configure Secondary Core Oscillator Subsystem to generate clocks needed to operate core and its peripherals. In the firmware, the Secondary core is configured to operate at 100MHz.
- To configure I/O ports and Secondary Core peripherals (such as PWM Generators PG1, PG2, and PG3, ADC Cores, UART1) required to function the firmware.
- To execute the Motor Control Demo Application based on the Microchip Application note AN1292.

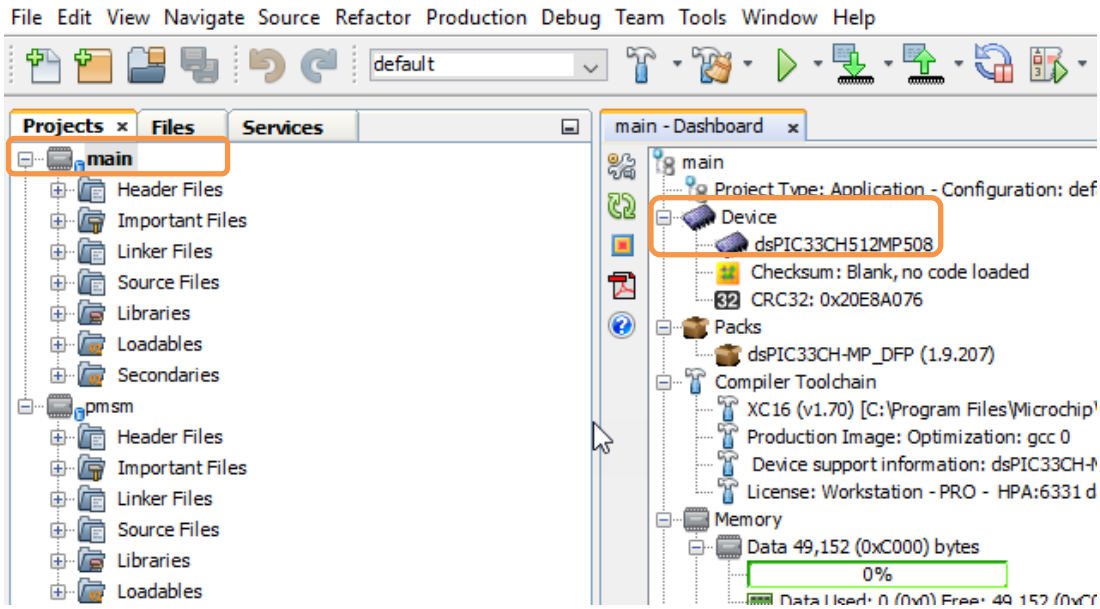
Once Main Core programs and enables the Secondary Core, it can autonomously run the Motor Control Demo application residing in its PRAM. The Motor Control Demo application uses a push button to start or stop the motor and a potentiometer to vary the speed of the motor.

For more details, refer to Microchip Application note AN1292 *"Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)"* available on the website.

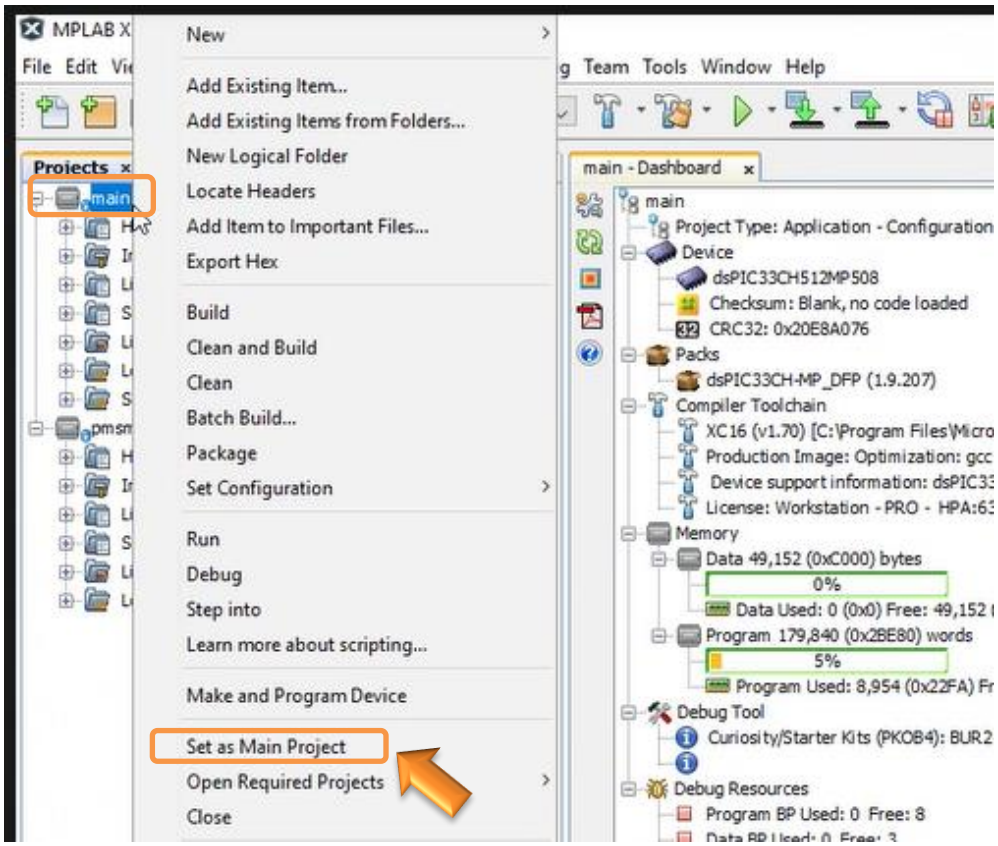
5.2. Basic Demonstration

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

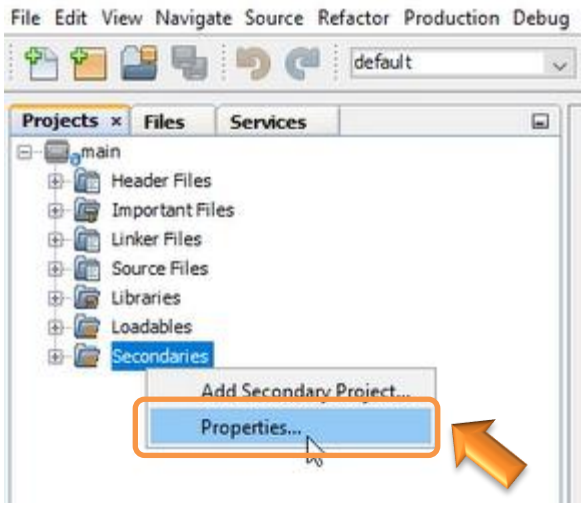
1. Start MPLAB X IDE and open (*File>Open Project)* the Main Project `main.X` with device selection *dsPIC33CH512MP508* (Main Core).



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

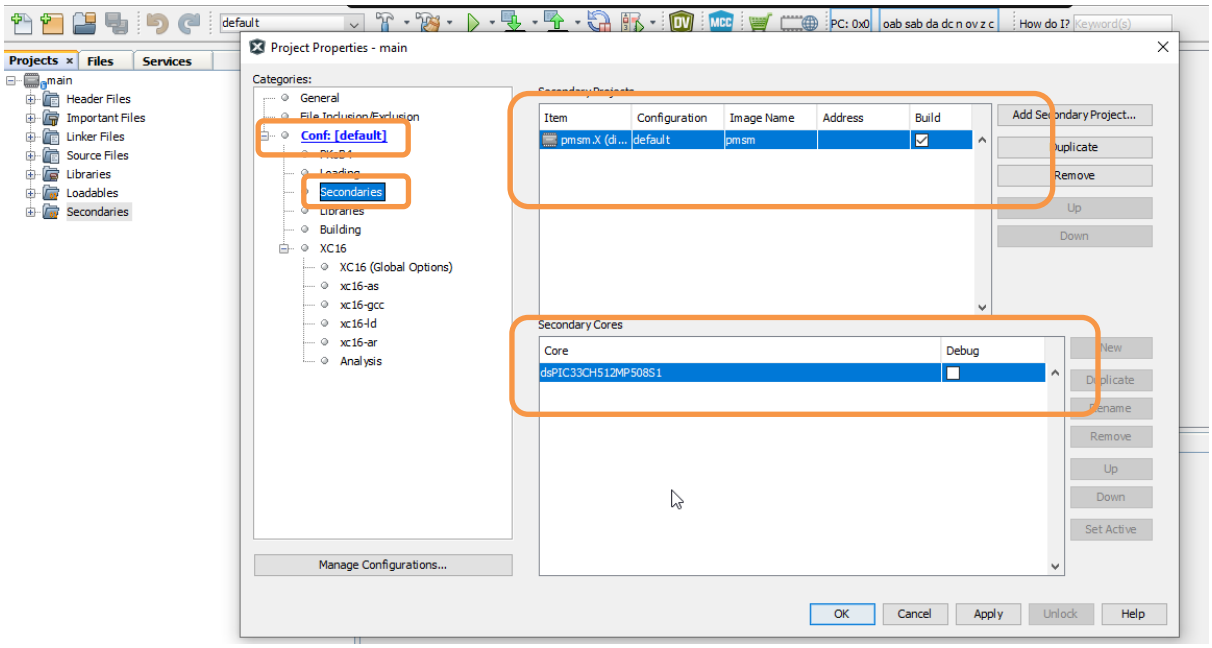


3. In the Projects window, right-click on the Secondaries folder of the project tree (of Main project *main.X)* and select “Properties.” This will open the “Secondaries” category of the Project Properties dialog.



Verify the “Secondaries” category of Project Properties dialog, and ensure details are as follows (see figure):

- Item is *pmsm.X*
- Image name is *“pmsm,”* and
- Check Box *“Build’* is checked
- Check Box *“Debug”* is unchecked

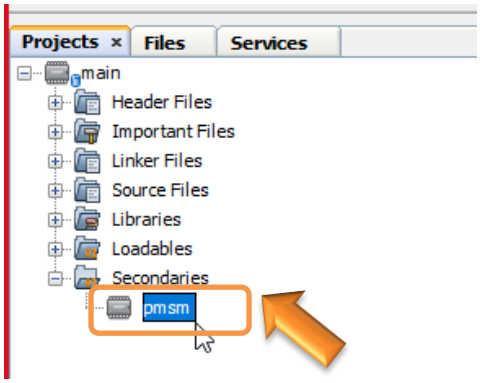


Note:

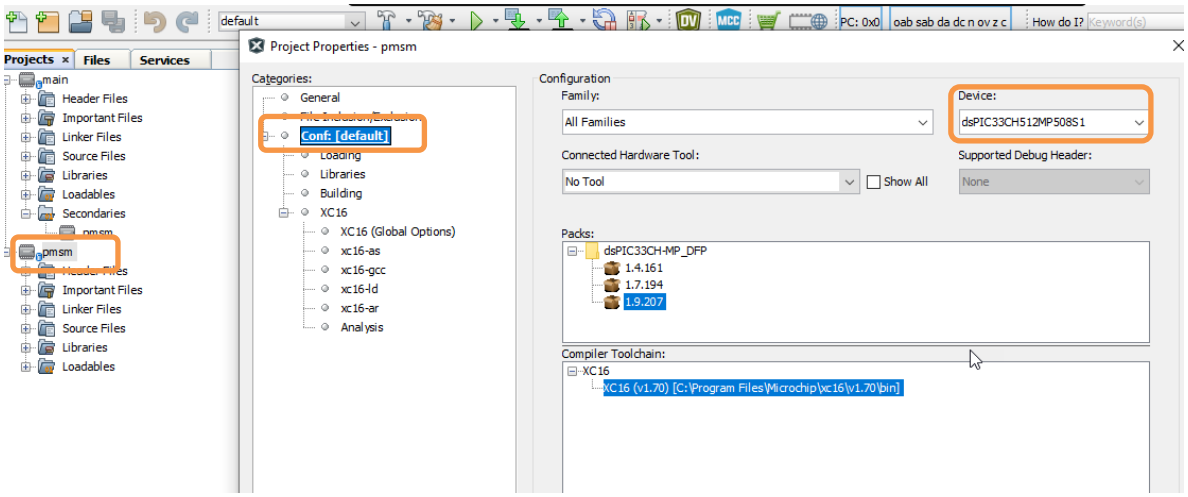
You may encounter build error,

- If any of the values are not as mentioned above
- If the secondary project *pmsm.X* is moved or deleted from the firmware directory

4. In the Projects window, right-click on the Secondaries folder of the project tree (of Main project *main.X)* and select the project "pmsm." This will open the Secondary project *pmsm.X* in the MPLAB X IDE project window. Alternatively, you can open (*File>Open Project*) the Secondary project from its current location like any other MPLAB X project.



The selected device in Secondary project *pmsm. X* can be viewed by opening its Project Property Dialog. As can be seen from the figure below, this device is set as dsPIC33CH512MP508S1(as shown in the figure below), representing the secondary core of the dsPIC33CH512MP508.



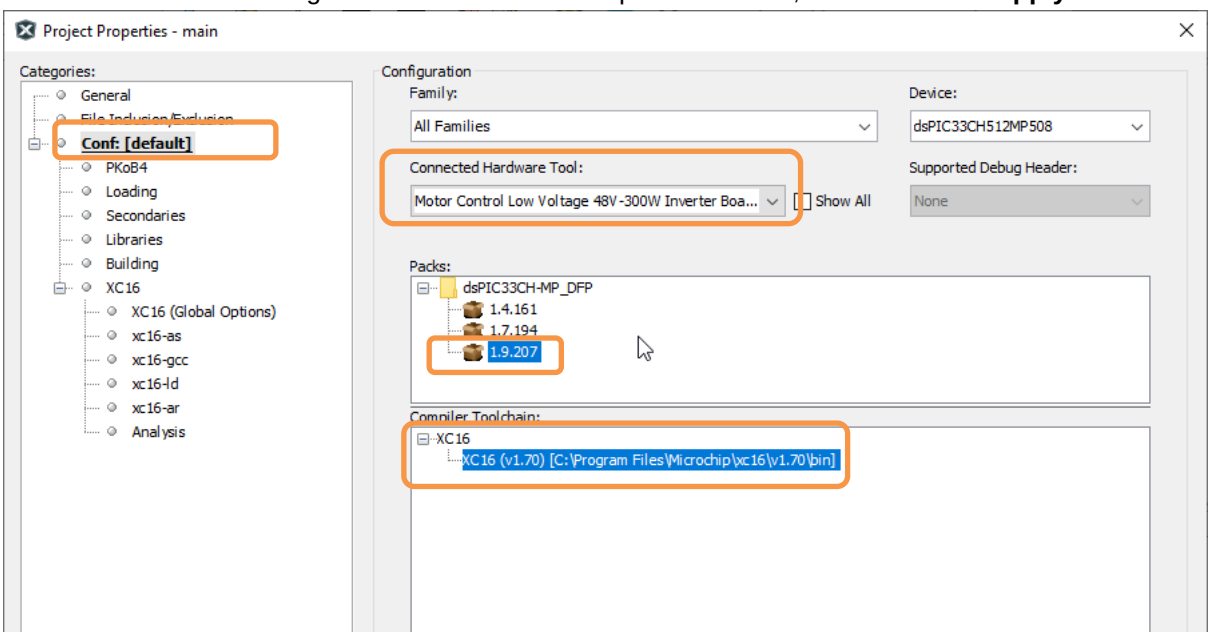
5. Unfold Header Files folder of Secondary project tree and click open the header file `userparms.h` in Editor window. Verify header file `userparms.h` (included in the Secondary project *pmsm.X*) to ensure macro definitions `TUNING`, `OPEN_LOOP_FUNCTIONING`, `TORQUE_MODE` and `SINGLE_SHUNT` is not defined.

```
#undef TUNING

#undef OPEN_LOOP_FUNCTIONING

#undef TORQUE_MODE

#undef SINGLE_SHUNT
```

Note:

The motor phase currents can be reconstructed from the DC Bus current by appropriately sampling it during the PWM switching period, called a single-shunt reconstruction algorithm. The firmware can be configured to demonstrate the single shunt reconstruction algorithm by defining the macro '`SINGLE_SHUNT`' in the header file `userparams.h`
For additional information, refer to Microchip application note **AN1299, "Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM**." By default, the firmware uses phase currents measured across the phase shunt resistors on two of the half-bridges of the three-phase inverter ('dual shunt configuration') to implement FOC.

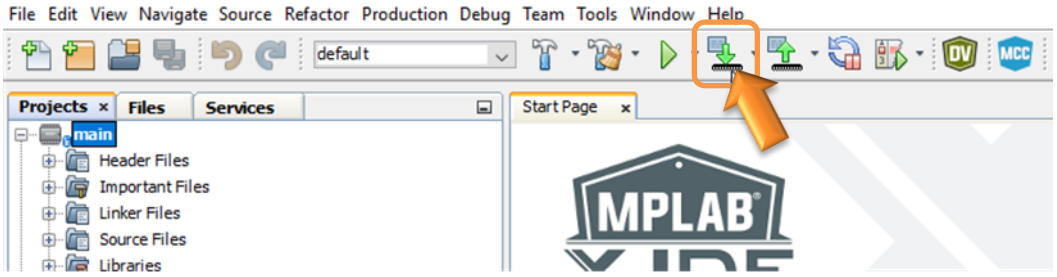
6. Right-click on the Main Project *main.X* and select "Properties" to open its Project Properties Dialog. Click the "Conf: [default]" category to reveal the general project configuration information. The development tools used for testing the firmware are listed in section 2.2. Software Tools Used for Testing the firmware.
   In the ***'Conf-default'*** category window:
   - Select the specific Compiler Toolchain from the available list of compilers. Please ensure MPLAB® XC16 Compiler supports the device dsPIC33CH512MP508.In this case, "XC16(v1.70)" is selected.
   - Select the 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, "dsPIC33CH-MP_DFP 1.9.207" is selected.
   - After selecting Hardware Tool and Compiler Toolchain, click the button **Apply**



7. Right-click on the associated Secondary Project *pmsm.X* and select "Properties" to open its Project Properties Dialog. Click the "Conf: [default]" category to reveal the general project configuration information.
   In the ***'Conf: [default]'*** category window:
   - Select the specific Compiler Toolchain from the available list of compilers. Please ensure MPLAB XC16 Compiler supports the device dsPIC33CH512MP508S1.
   - After selecting Compiler Toolchain, click the button **Apply**

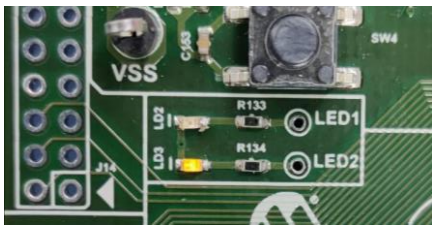
   This step is required to build the Secondary Project with a specific compiler version.

8. To build the Main Project (in this case, *main.X*) and program the device dsPIC33CH512MP508, click "**Make and Program Device Main project**" on the toolbar. Upon this, MPLAB X IDE begin executing the following activities in order:
   - builds Secondary Project *pmsm.X* (linked to Main Project *main.X*)
   - builds Main Project *main.X* and
   - Programs Main flash memory of dsPIC33CH512MP508 with code generated when building the Main Project and the Secondary Project.

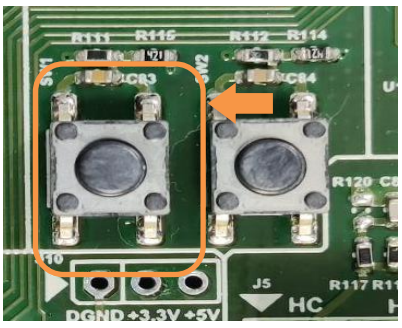


> **Note:**
>
> In this firmware configuration, the Main Core programs the Secondary Core. When device is programmed, the Secondary core image is placed in the Main flash. When the Main Core is powered on and begins execution of code, it transfers the Secondary image from the Main flash to the Secondary PRAM.

9. If the device is successfully programmed, **LD2('LED1')** will be turned ON, indicating that the dsPIC® DSC is enabled.

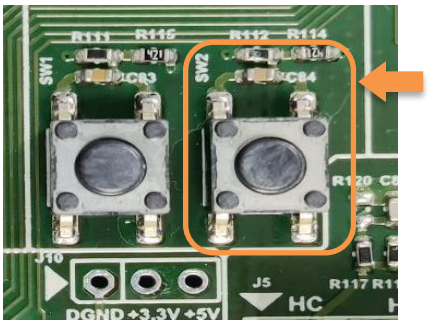


10. Run or stop the motor by pressing the push button **SW1**. The motor should start spinning smoothly in one direction in the 'Normal Speed Range.' Ensure that the motor is spinning smoothly without any vibration. The LED LD3 ('LED2') is turned ON to show the button is pressed to start the motor.



11. If desired, the motor speed can be varied using the potentiometer (‘POT1’).

12. To enter the extended speed range (`NOMINAL_SPEED_RPM` to `MAXIMUM_SPEED_RPM`), press the push button **SW2**. Press the push button **SW2** again to revert the speed of the motor to its normal speed (`END_SPEED_RPM` to `NOMINAL_SPEED_RPM`) range.



13. Press the push button **SW1** to stop the motor.

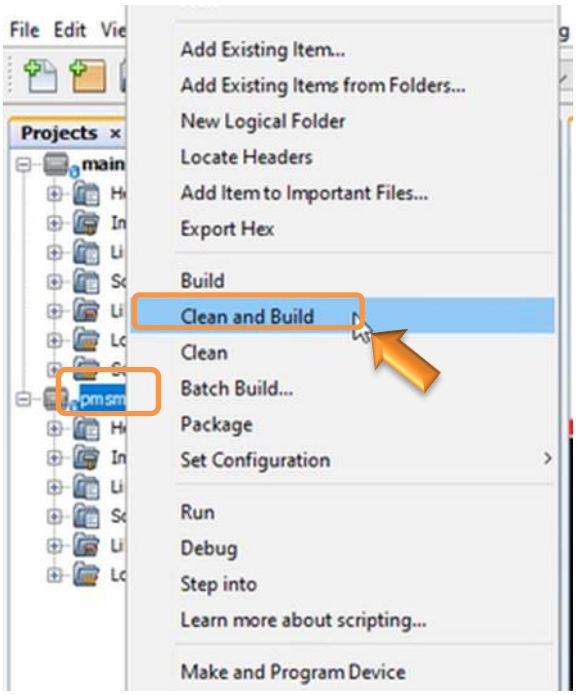
Note:

The macro definitions `END_SPEED_RPM`, `NOMINAL_SPEED_RPM`, and `MAXIMUM_SPEED_RPM` are specified in `userparms.h` file included in the project *`pmsm.X`*. The definitions `NOMINAL_SPEED_RPM`, and `MAXIMUM_SPEED_RPM` are defined as per the specification provided by the motor manufacturer. *Exceeding manufacture specification may lead to damage of the motor or(and) the board.*

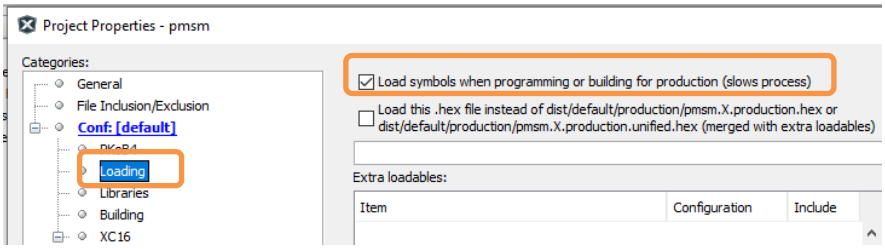
5.3. Data visualization through X2CScope Plug-in of MPLABX

The secondary project (pmsm.X) application firmware comes with the initialization needed to interface Controller with Host PC to enable Data visualization through X2C Scope plug-in. X2C-Scope is a third-party plugin for MPLAB X, which helps real-time diagnostics.

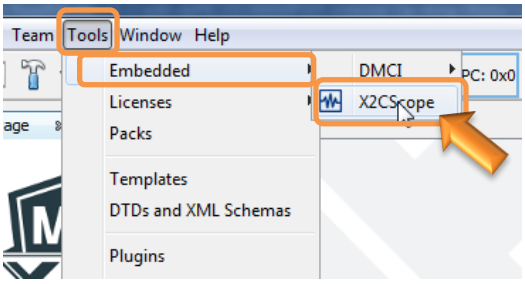
1. Ensure X2C-Scope Plug-in is installed. For more information on how to set up a plug-in, refer to https://microchipdeveloper.com/mplabx:tools-plugins-available
2. To utilize X2C communication for this demonstration, connect a micro-USB cable between the Host PC and the micro-USB connector provided on the MCLV-48V-300W Inverter Board. This interface is used for programming as well.
3. Ensure the application is configured and running as described under Section Basic Demonstration by following steps 1 through 13.
4. Build the secondary project pmsm.X. To do that, right-click on the project pmsm.X and select “Clean and Build.”



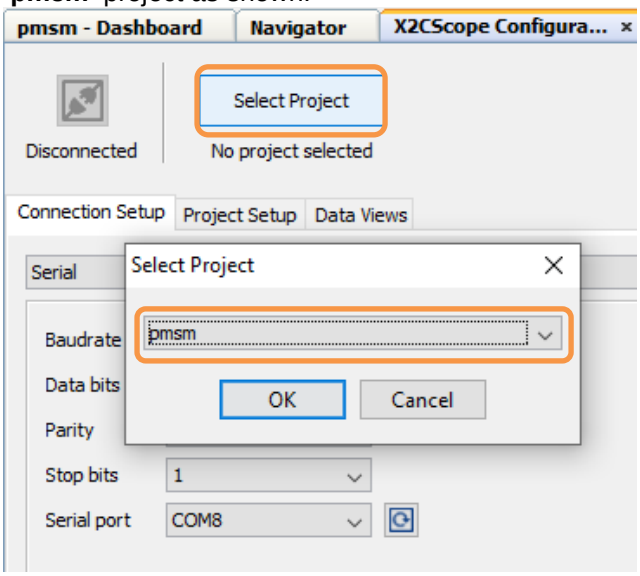
5. Please ensure that the checkbox “Load symbols when programming or building for production (slows process)” is checked, which is under the “Loading” category of the Project Properties window (in the `pmsm.X` project properties).



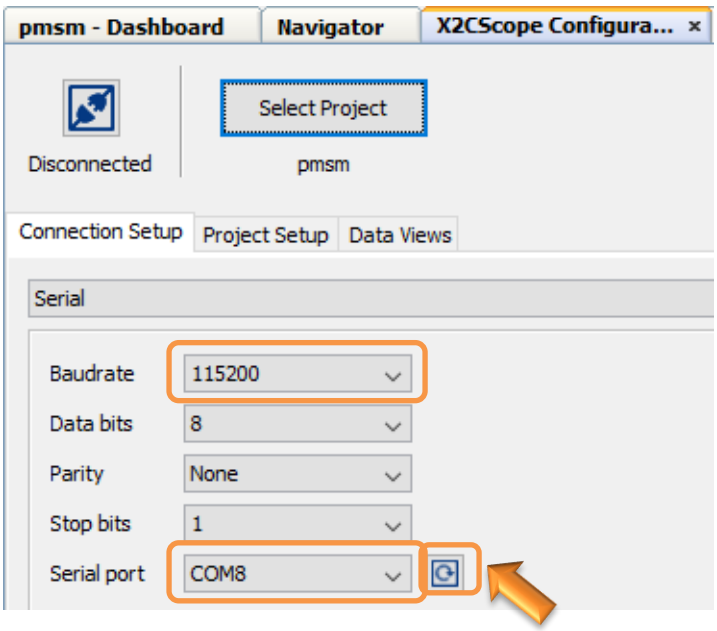
6. Open the X2C-Scope window by selecting `Tools>Embedded>X2CScope`.



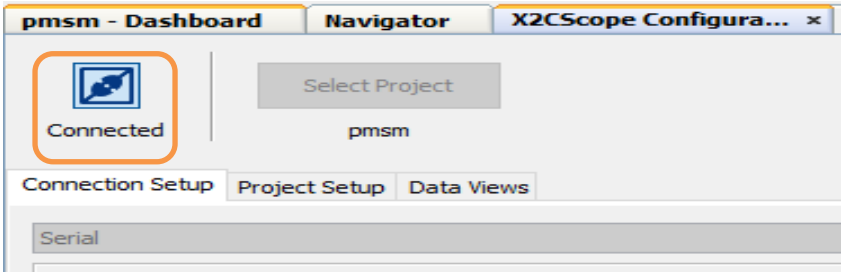
7. In the X2C-Scope Configuration window, using the “Select Project” menu, select the **‘pmsm’** project as shown.



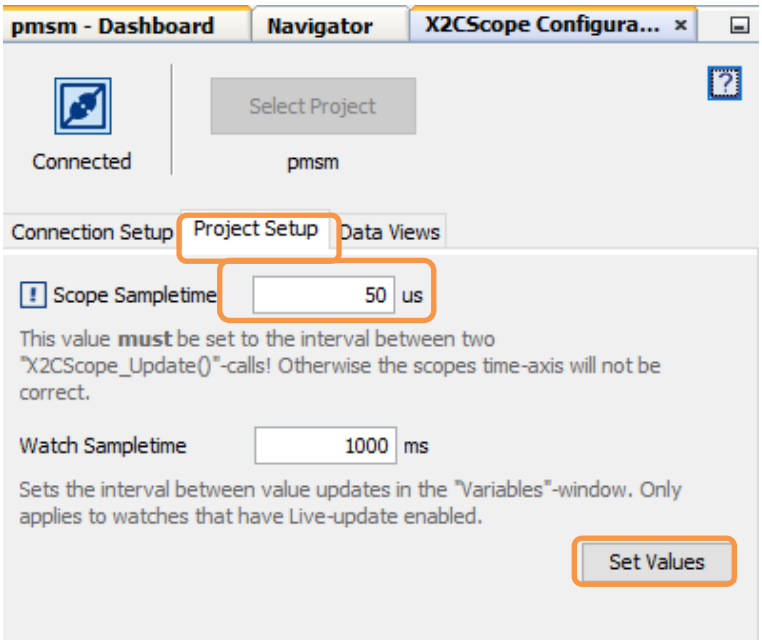
8. Remote Communication needs to be set up, as shown in the following figure. Ensure the communication baud rate is set to 115200 as configured in the application firmware. The COM port used depends on the system settings. `The refresh button` lists the available COM ports. Select the COM port as per the connection.



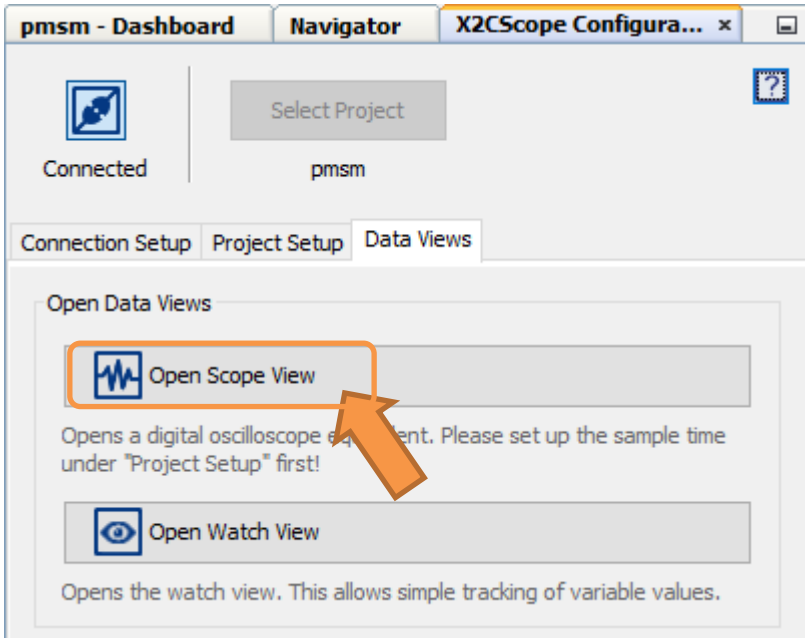
9. Once the COM port is detected, click on “Disconnected” and turn to “Connected” to establish serial communication between the Host PC and the board.



10. Set the “Project Setup” as shown below and click “Set Values.” Set “Scope Sampletime” as the interval at which `X2CScopeUpdate()` is called. In this application, it is every 50µs.

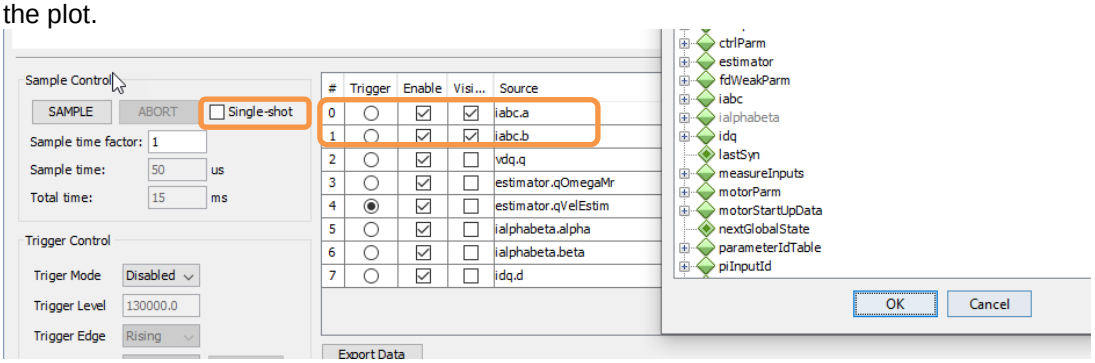


11. Click on ‘Open Scope View’ (under sub-window “Data Views”); this opens ‘Scope Window.’

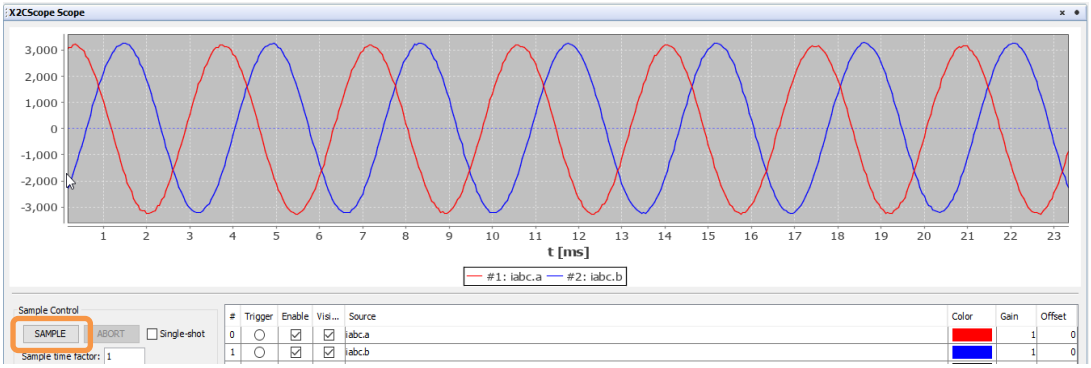


12. In this window, select the variables that need to be watched. To do this, click on the source against each channel, a window Select Variables opens on the screen. From the available list, the required variable can be chosen. Ensure checkboxes Enable & Visible are checked for the variables to be plotted.

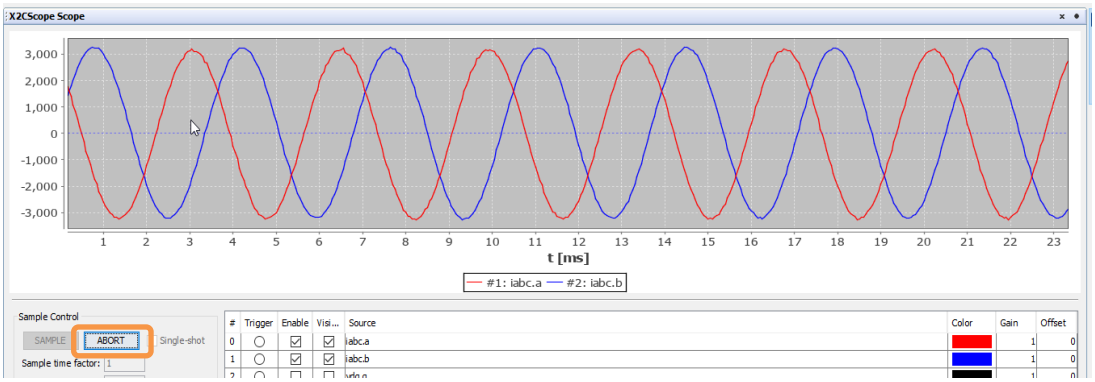
    To view data plots continuously, uncheck `Single-shot`. When `Single-shot` is checked, it captures the data once and stops. The `Sample time factor` value multiplied with Sample time decides the time difference between any two consecutive data points on the plot.



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



14. Click on ABORT to stop.



6. REFERENCES:

For additional information, refer following documents or links.

1. AN1292 Application Note "Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)."
2. AN1299 Application Note "Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM"
3. MCLV-48V-300W Inverter Board User's Guide
4. dsPIC33CH512MP508 Family datasheet.
5. Family Reference manuals (FRM) of dsPIC33CH512MP508 family
6. MPLAB® X IDE User's Guide (DS50002027) or MPLAB® X IDE help
7. MPLAB® X IDE installation
8. MPLAB® XC16 Compiler installation