AN1292 Demonstration ReadMe for MCLV-48V-300W Inverter Board with dsPIC33CH512MP508 Motor Control DIM (MPLAB® X IDE)

1. INTRODUCTION

This document describes the setup requirements for driving a Permanent Magnet Synchronous Motor (PMSM) using Sensorless Field Oriented Control (FOC) and PLL Estimator Algorithms on the hardware platform MCLV-48V-300W Inverter Board and dsPIC33CK512MP508 Motor Control Dual In-line Module (DIM).

For more details about the PLL estimator, 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)."**

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, after this, this firmware package is referred to as **firmware.**

2.2. Software Tools Used for Testing the firmware

- MPLAB® X IDE **v6.00**
- DFP: **dsPIC33CH-MP_DFP v1.11.240**
- MPLAB® XC16 Compiler **v2.00**
- MPLAB® X IDE Plugin: **X2C-Scope v1.3.3**

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 Dual In-line Module (EV76L31A)
- 24V Power Supply (AC002013)
- 24V 3-Phase Brushless DC Motor (AC300020)

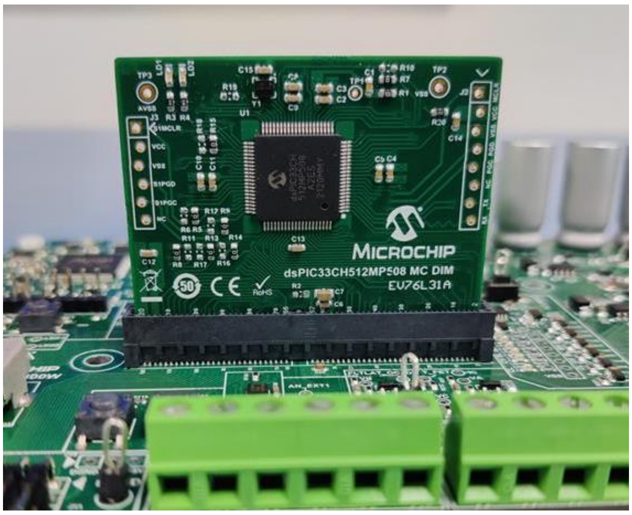
Note:

All items listed under this section Hardware Tools Required for the Demonstration are available at microchip DIRECT.

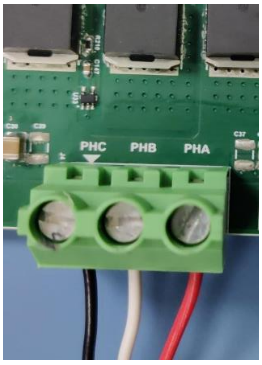
3. HARDWARE SETUP

This section describes the hardware setup needed for the demonstration.

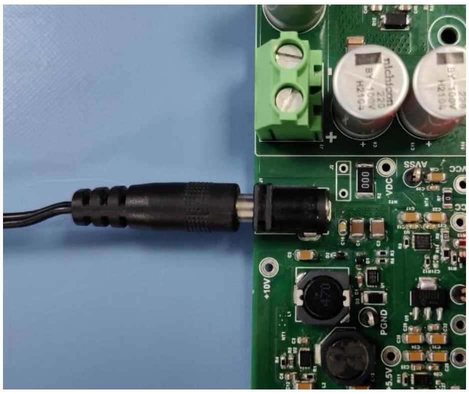
1. Motor currents are amplified on the MCLV-48V-300W Inverter Board. The firmware and DIM are configured to sample and convert external amplifier outputs to measure the motor currents needed to implement FOC.
2. Insert the **dsPIC33CH512MP508 Motor Control DIM** into the DIM Interface **connector J8**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



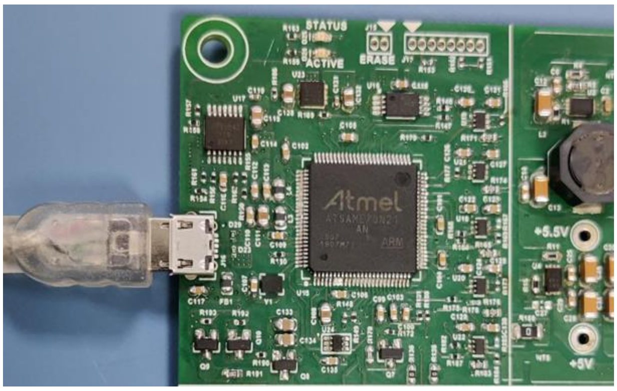
3. Connect the 3-phase wires from the motor to PHC, PHB, and PHA of the **connector J4** (no specific order), provided on the MCLV-48V-300W Inverter Board.



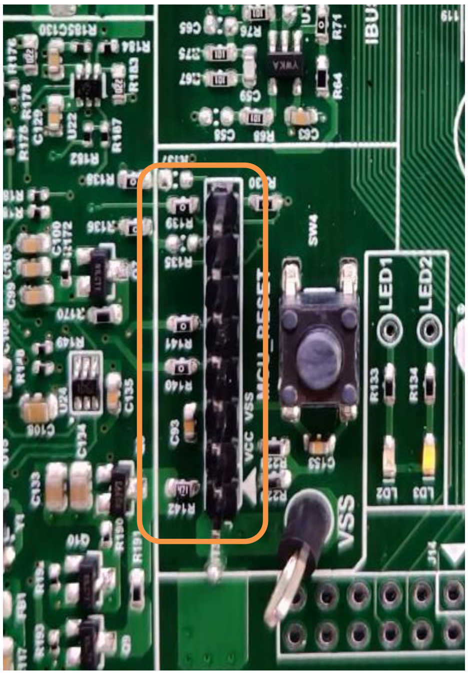
4. Plug in the 24V power supply to **connector J1** provided on the MCLV-48V-300W Inverter Board. Alternatively, the Inverter Board can also be powered through connector J3.

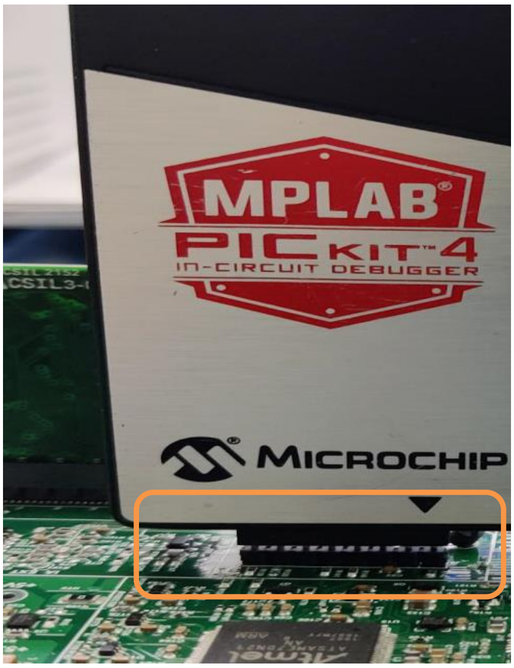


5. The board has an onboard programmer **PICkit™ 4 On Board (PKOBv4)**, which can be used for programming or debugging the microcontroller or dsPIC DSC on the DIM. To use an onboard programmer, connect a micro-USB cable between Host PC and **connector J16** on the MCLV-48V-300W Inverter Board.



6. Alternatively, connect the Microchip programmer/debugger **MPLAB PICkit 4 In-Circuit Debugger** between the Host PC used for programming the device and the **ICSP header J9** on the MCLV-48V-300W Inverter Board (as shown). Ensure that PICkit 4 is oriented correctly before proceeding.





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 (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 an 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 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 (DS70005371)**.

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 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().**

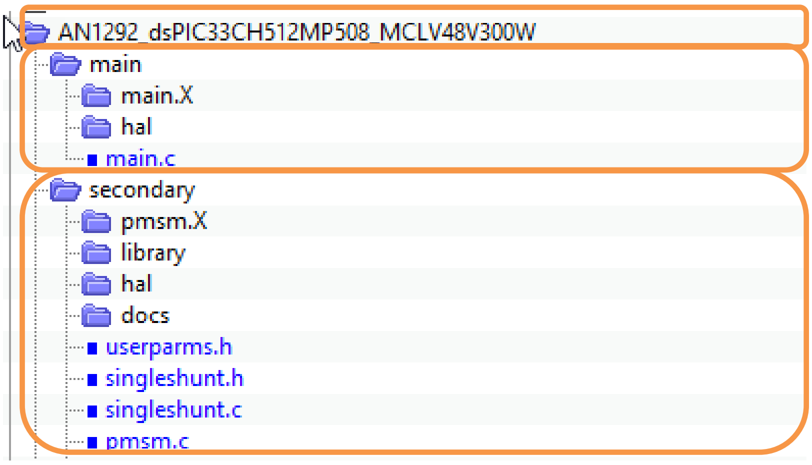
The function of the Secondary Core (as defined in the Secondary Project **pmsm.X)** is:

- 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.

The firmware directory structure is shown below:

Main Project **main.X** and associated files.

Secondary Project **pmsm.X** and associated files.



```
AN1292_dsPIC33CH512MP508_MCLV48V300W
  main
    main.X
    hal
    main.c
  secondary
    pmsm.X
    library
    hal
    docs
    userparms.h
    singleshunt.h
    singleshunt.c
    pmsm.c
```

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.

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, etc. 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 Microchip website.

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:

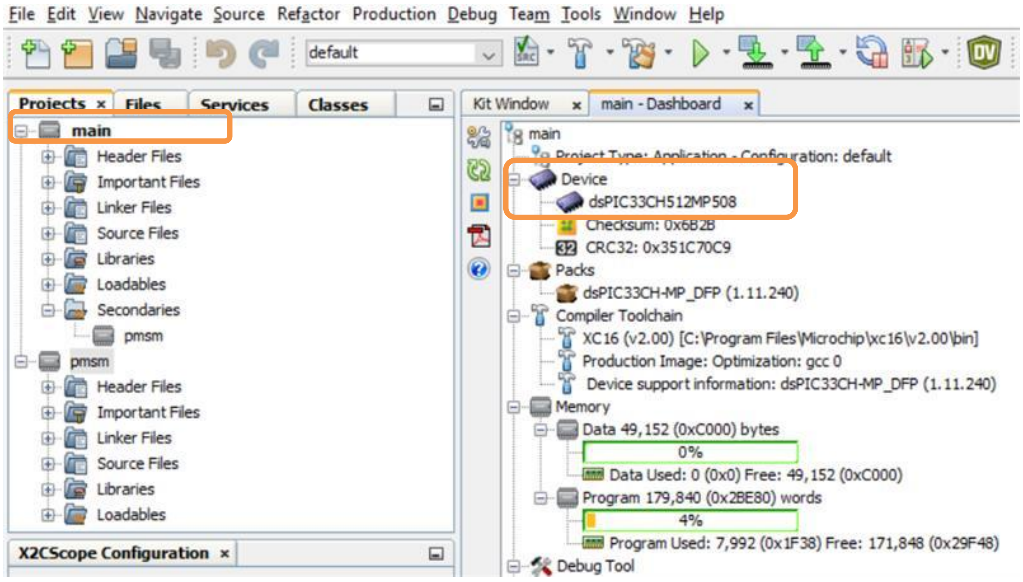
- Shorten the name of the directory containing the firmware used in this demonstration. In case you renamed the directory, consider the new name while reading instructions provided in the upcoming sections of the document.
- Place firmware in a location, such that absolute path length of each file included 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.

5.2. Basic Demonstration

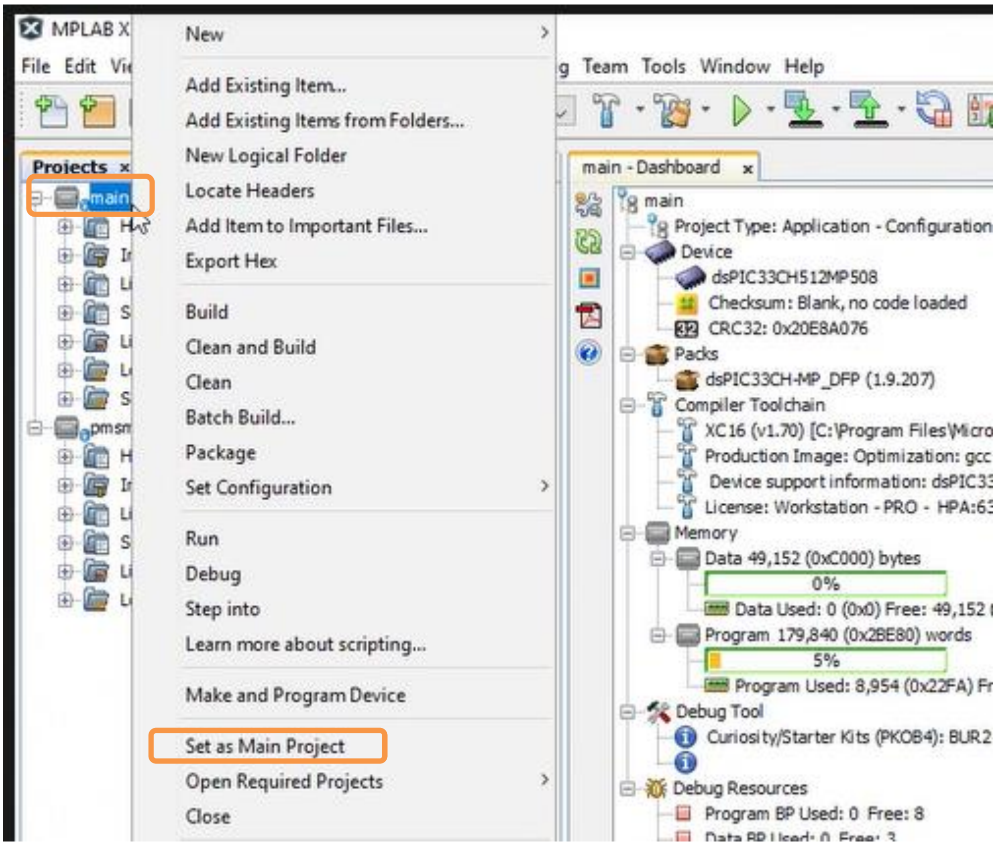
Follow the below instructions step by step to set up and run the motor control demo application:

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



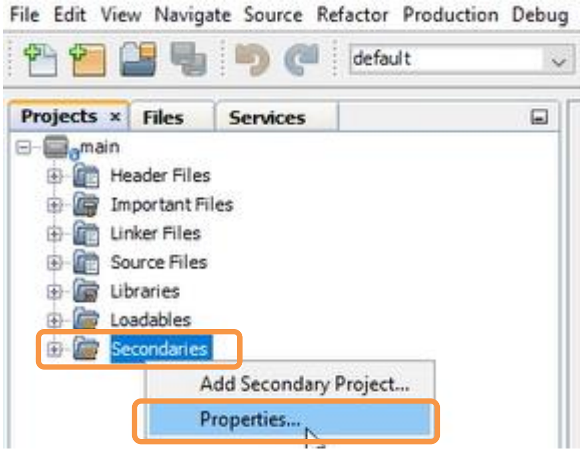
Project Type: Application - Configuration: default
Device
dsPIC33CH512MP508
Checksum: 0x6B2B
CRC32: 0x351C70C9
Packs
dsPIC33CH-MP_DFP (1.11.240)
Compiler Toolchain
XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]
Production Image: Optimization: gcc 0
Device support information: dsPIC33CH-MP_DFP (1.11.240)
Memory
Data 49,152 (0xC000) bytes
0%
Data Used: 0 (0x0) Free: 49,152 (0xC000)
Program 179,840 (0x2BE80) words
4%
Program Used: 7,992 (0x1F38) Free: 171,848 (0x29F48)
Debug Tool

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.X** will then appear in **bold**.



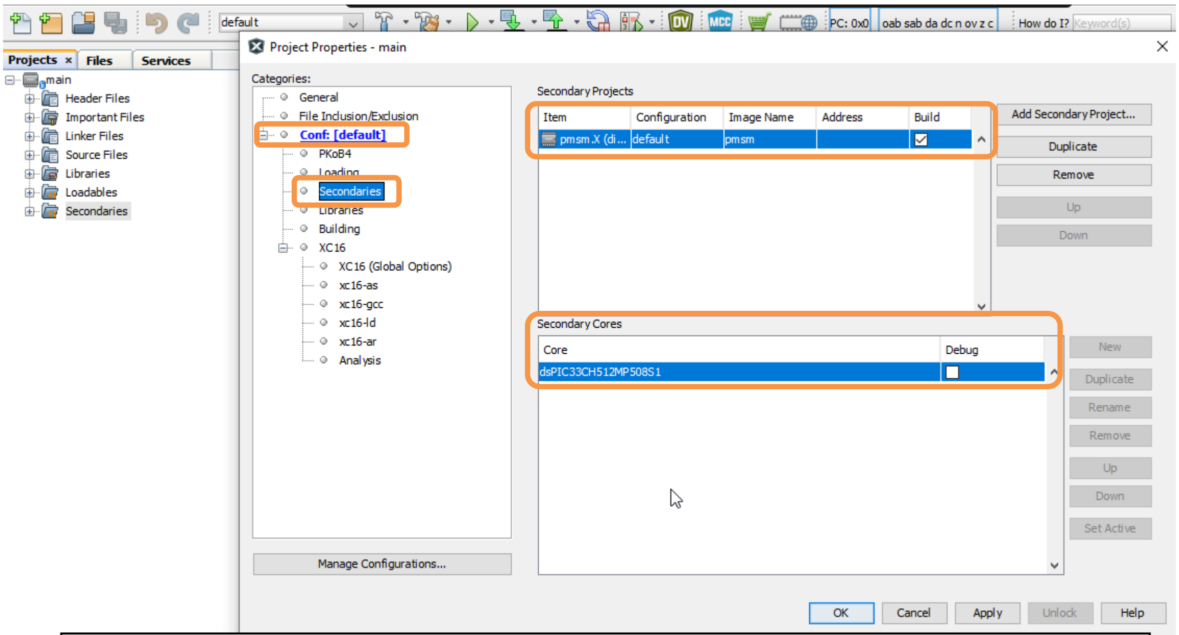
Project Type: Application - Configuration
Device
dsPIC33CH512MP508
Checksum: Blank, no code loaded
CRC32: 0x20E8A076
Packs
dsPIC33CH-MP_DFP (1.9.207)
Compiler Toolchain
XC16 (v1.70) [C:\Program Files\Micro
Production Image: Optimization: gcc
Device support information: dsPIC33
License: Workstation - PRO - HPA:63
Memory
Data 49,152 (0xC000) bytes
0%
Data Used: 0 (0x0) Free: 49,152 (
Program 179,840 (0x2BE80) words
5%
Program Used: 8,954 (0x22FA) Fr
Debug Tool
Curiosity/Starter Kits (PKOB4): BUR2
Debug Resources
Program BP Used: 0 Free: 8
Data BP Used: 0 Free: 3

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**,
- Check Box **Build** is **checked** and
- Check Box **Debug** is **unchecked**



Secondary Projects

Item	Configuration	Image Name	Address	Build
pmsm.X (di...	default	pmsm		☑

Secondary Cores

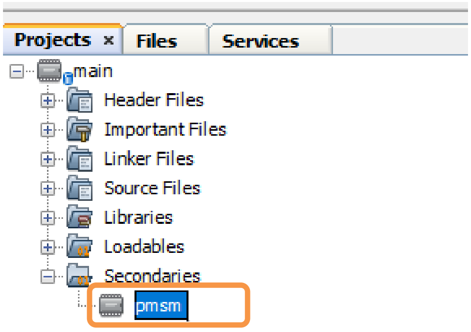
Core	Debug
dsPIC33CH512MP508S1	□

Note:

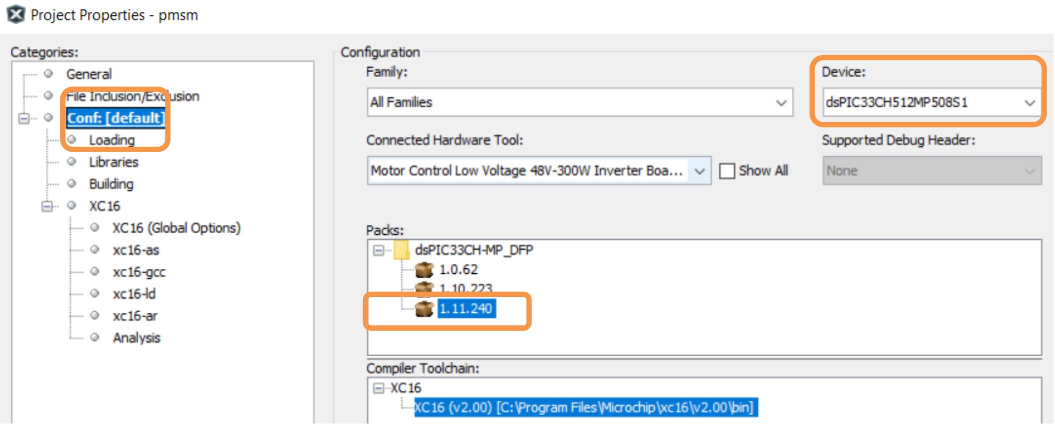
May encounter build error if,

- any of the values are not as mentioned above
- 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 Properties** 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**.



Project Properties - pmsm

Family: All Families

Device: dsPIC33CH512MP508S1

Connected Hardware Tool: Motor Control Low Voltage 48V-300W Inverter Boa... □ Show All

Supported Debug Header: None

Packs:
dsPIC33CH-MP_DFP
1.0.62
1.10.223
1.11.240

Compiler Toolchain:
XC16
XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]

5. Open **userparms.h** (**pmsm.X > Header Files**) in the project **pmsm.X**.

- Ensure that the macros **TUNING**, **OPEN_LOOP_FUNCTIONING**, **TORQUE_MODE**, and **SINGLE_SHUNT** is not defined in the header file **userparms.h**.

```
#undef TUNING

#undef OPEN_LOOP_FUNCTIONING

#undef TORQUE_MODE

#undef SINGLE_SHUNT
```

- When internal amplifiers are used for current amplification (referred to as **internal op-amp configuration**), **define** the macro **INTERNAL_OPAMP_CONFIG in userparms.h.**

```
#define INTERNAL_OPAMP_CONFIG
```

- Otherwise, if external amplifiers are used for current amplification (referred to as **external op-amp configuration**), **undefine** the macro **INTERNAL_OPAMP_CONFIG** in the header file **userparms.h**.

```
#undef INTERNAL_OPAMP_CONFIG
```

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 algorith**m by defining the macro **SINGLE_SHUNT** in the header file **userparms.h**

For additional information, refer to Microchip application note **AN1299, "Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM."**

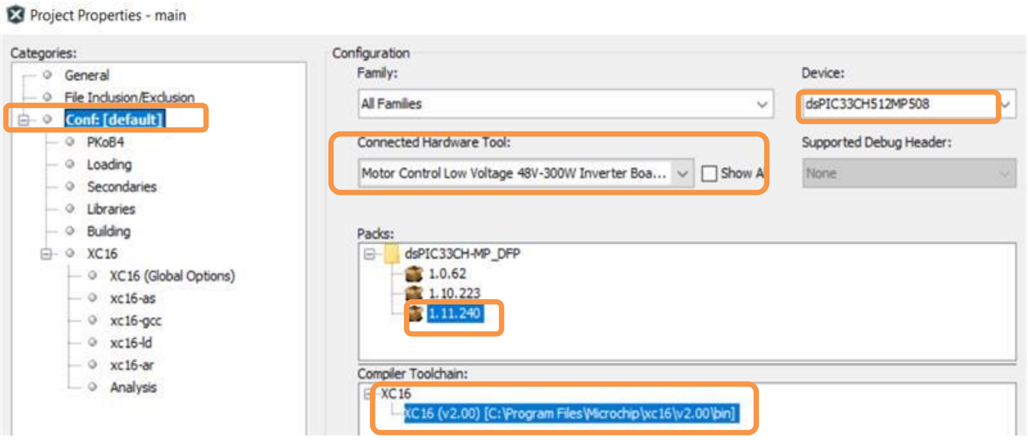
By default, the firmware uses phase currents measured across the phase shunt resistors on two of the half-bridges of the three-phase inverter **('dual shunt configuration'**) to implement FOC.

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:

   - Ensure the selected **Device** is **dsPIC33CH512MP508**.
   - 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, **dsPIC33CH-MP_DFP 1.11.240** is selected.
   - Select the specific **Compiler Toolchain** from the available list of **XC16** compilers. In this case, **XC16(v2.00)** is selected.
   - After selecting Hardware Tool and Compiler Toolchain, click the button **Apply**

   Please ensure that the selected MPLAB® XC16 Compiler and Device Pack support the device configured in the firmware.



Family: All Families

Device: dsPIC33CH512MP508

Connected Hardware Tool: Motor Control Low Voltage 48V-300W Inverter Boa... □ Show A

Supported Debug Header: None

Packs:
dsPIC33CH-MP_DFP
1.0.62
1.10.223
1.11.240

Compiler Toolchain:
XC16
XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]

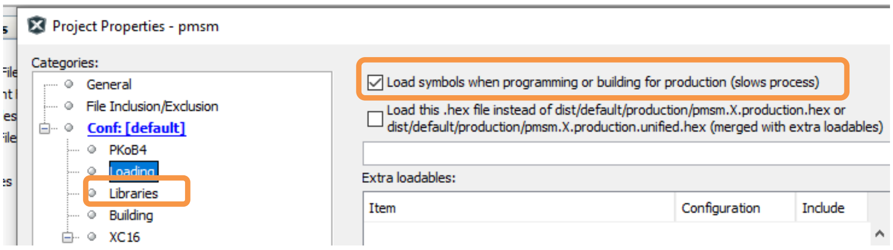
7. Right-click on the associated Secondary Project **pmsm.X** and select **Properties** to open its **Project Properties** Dialog. Click the **Conf: [default]** category to reveal the general project configuration information.

   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. 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 **Secondary** project **pmsm.X**



☑ Load symbols when programming or building for production (slows process)

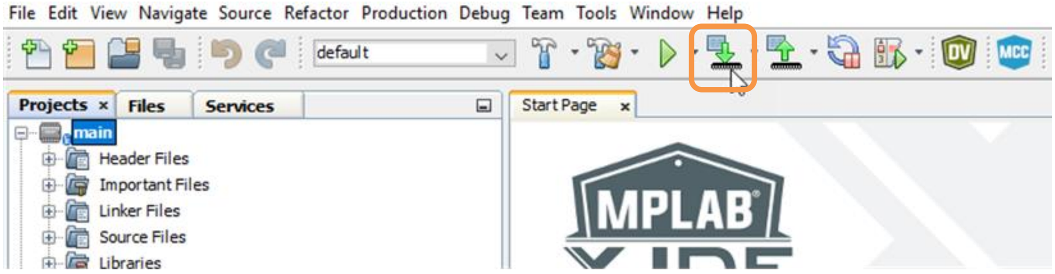
□ Load this .hex file instead of dist/default/production/pmsm.X.production.hex or dist/default/production/pmsm.X.production.unified.hex (merged with extra loadables)

Extra loadables:

Item	Configuration	Include

9. 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:

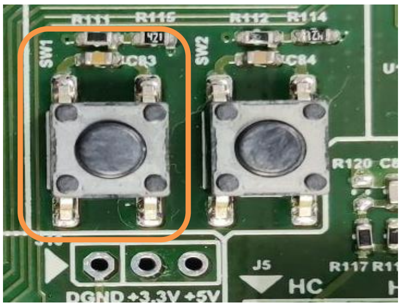
   - Builds Secondary Project **pmsm.X** (linked to Main Project **main.X**)
   - Builds Main Project **main.X** and
   - Programs Main flash memory of dsPIC33CH512MP508 with code generated when building the Main Project and the Secondary Project.



> **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.

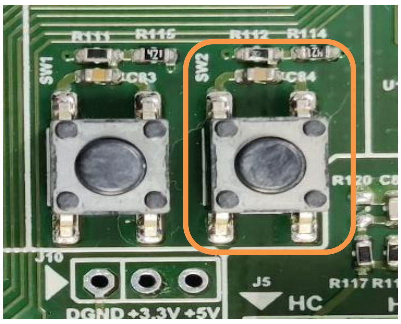
10. If the device is successfully programmed, **LD2(LED1)** will be turned **ON**, indicating that the dsPIC® DSC is enabled.

11. Run or stop the motor by pressing the push button **SW1**. The motor should start spinning smoothly in one direction in the nominal 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.



12. If desired, the motor speed can be varied using the potentiometer (**POT1**).

13. Press the push button **SW2** to enter the extended speed range (NOMINAL_SPEED_RPM to MAXIMUM_SPEED_RPM).
Press the push button **SW2** again to revert the speed of the motor to its nominal speed range (END_SPEED_RPM to NOMINAL_SPEED_RPM)



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

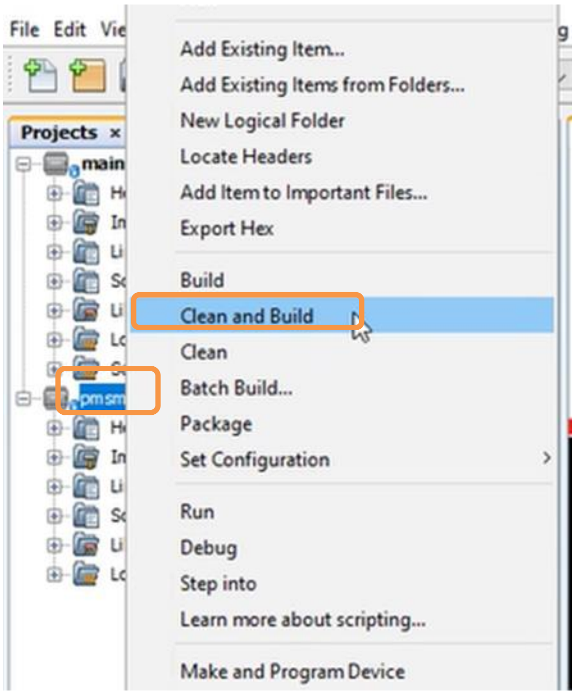
Note:

The macros END_SPEED_RPM, NOMINAL_SPEED_RPM, and MAXIMUM_SPEED_RPM are specified in the header file **userparms. h** included in the project **pmsm.X.** The macros NOMINAL_SPEED_RPM and MAXIMUM_SPEED_RPM are defined as per the Motor manufacturer’s specifications. Exceeding manufacture specifications may damage the motor or the board or both.

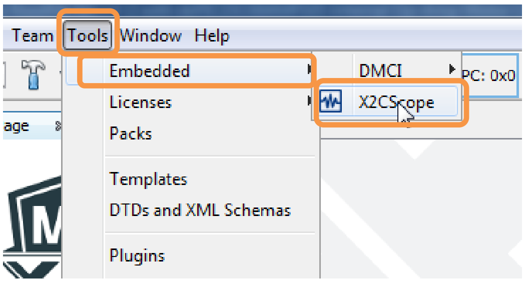
5.3. Data visualization through X2C-Scope Plug-in of MPLAB X

X2C-Scope is a third-party plug-in in MPLAB X, which helps in real-time diagnostics. The secondary project (**pmsm.X**) 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.

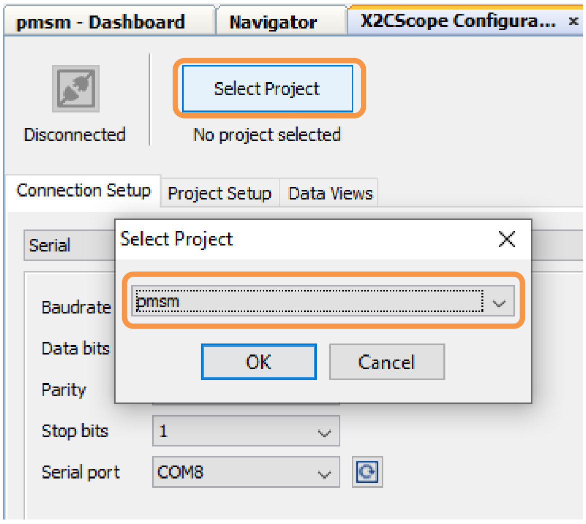
1. To establish serial communication with the host PC, connect a micro-USB cable between the host PC and **connector J16** on the MCLV-48V-300W Inverter Board. This interface is also used for programming.
2. Ensure the application is configured and running as described under section 5.2 Basic Demonstration by following steps 1 through 14.
3. Build the secondary project **pmsm.X**. To do that, right-click on the project **pmsm.X** and select **Clean and Build**.



4. Open the **X2C-Scope** window by selecting **Tools>Embedded>X2CScope**.



5. 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 **pmsm** from the list of projects and click **OK**.



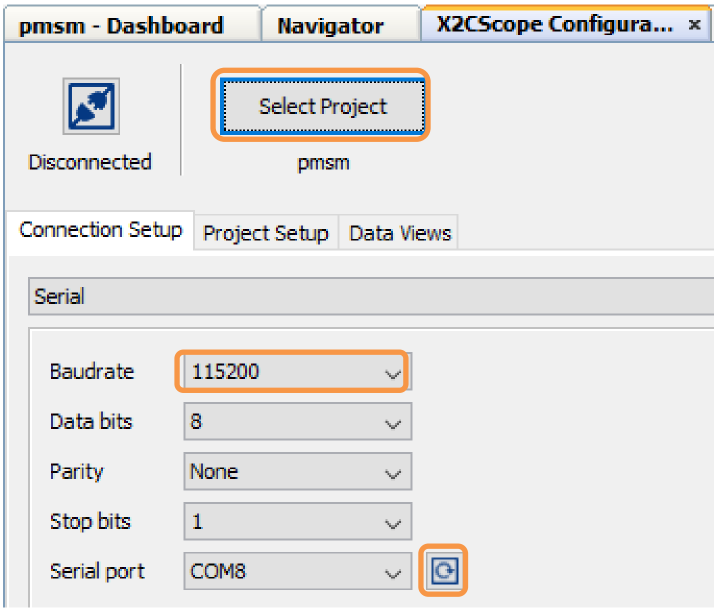
Select Project: pmsm

Stop bits: 1

Serial port: COM8

6. 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 MCLV-48V-300W Inverter Board. The **Serial port** depends on the system settings.



Baudrate: 115200

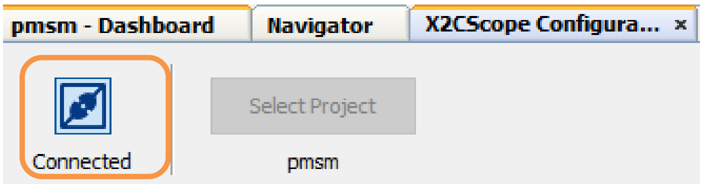
Data bits: 8

Parity: None

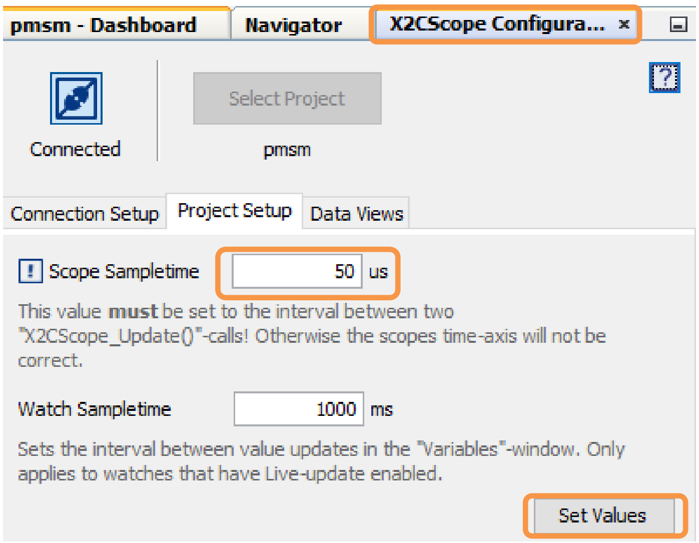
Stop bits: 1

Serial port: COM8

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



8. 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.



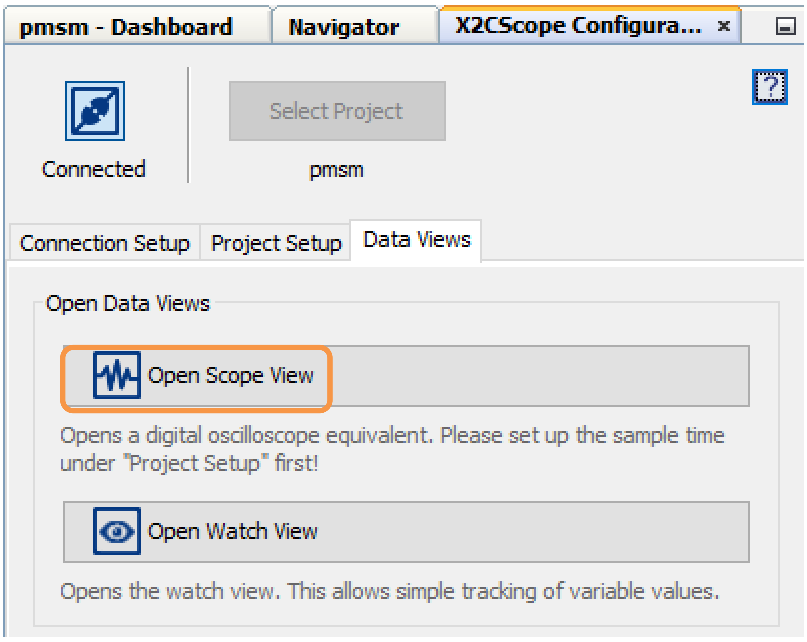
Scope Sampletime 50 us

This value **must** be set to the interval between two "X2CScope_Update()"-calls! Otherwise the scopes time-axis will not be correct.

Watch Sampletime 1000 ms

Sets the interval between value updates in the "Variables"-window. Only applies to watches that have Live-update enabled.

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

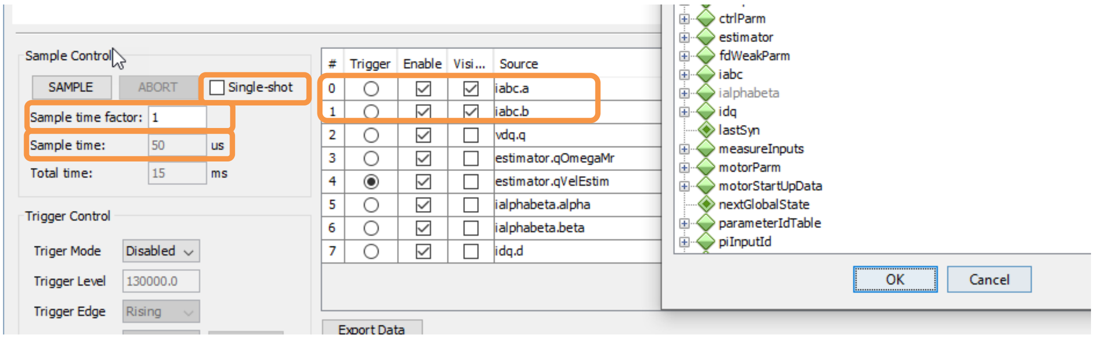


Opens a digital oscilloscope equivalent. Please set up the sample time under "Project Setup" first!

Opens the watch view. This allows simple tracking of variable values.

10. 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.



Sample Control

SAMPLE ABORT ☐ Single-shot

Sample time factor: 1

Sample time: 50 us

Total time: 15 ms

Trigger Control

Triger Mode Disabled

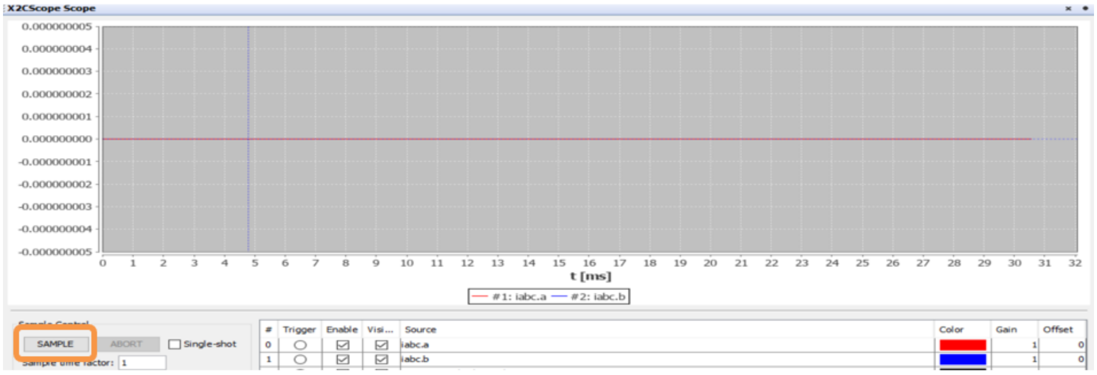
Trigger Level 130000.0

Trigger Edge Rising

#	Trigger	Enable	Visi...	Source
0	○	☑	☑	iabc.a
1	○	☑	☑	iabc.b
2	○	☑	☐	vdq.q
3	○	☑	☐	estimator.qOmegaMr
4	●	☑	☐	estimator.qVelEstim
5	○	☑	☐	ialphabeta.alpha
6	○	☑	☐	ialphabeta.beta
7	○	☑	☐	idq.d

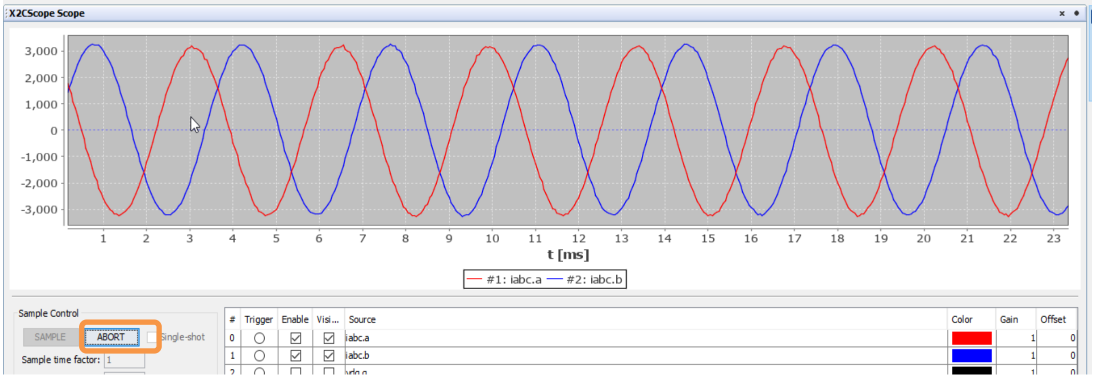
Export Data

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



#	Trigger	Enable	Visi...	Source	Color	Gain	Offset
0	○	☑	☑	iabc.a	🟥	1	0
1	○	☑	☑	iabc.b	🟦	1	0

12. Click on **ABORT** to stop.



#	Trigger	Enable	Visi...	Source	Color	Gain	Offset
0	○	☑	☑	iabc.a	🟥	1	0
1	○	☑	☑	iabc.b	🟦	1	0
2	○	☐	☐	vdq.q	■	1	0

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 (DS50003297)
4. dsPIC33CH512MP508 Motor Control Dual In-Line Module (DIM) Information Sheet (DS50003069)
5. dsPIC33CH512MP508 Family datasheet (DS70005371)
6. dsPIC33CH512MP508 Family Reference Manuals (FRM)
7. MPLAB® X IDE User’s Guide (DS50002027) or MPLAB® X IDE help
8. MPLAB® X IDE installation
9. MPLAB® XC16 Compiler installation
10. Installation and setup of X2Cscope plugin for MPLAB X