# AN1299 Demonstration ReadMe for the dsPICDEM™ MCHV-2 Development Board or dsPICDEM™ MCHV-3 Development Board with the dsPIC33CK256MP508 Internal Op-Amp Motor Control PIM (MPLAB® X IDE)

## 1. INTRODUCTION

This document describes the setup requirements for running the Single-Shunt Reconstruction Algorithm, which is referenced in AN1299 “Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM”.

The demonstration is configured to run on either the dsPICDEM™ MCHV-2 Development Board or the dsPICDEM™ MCHV-3 Development Board in the Internal Op-Amp configuration with the dsPIC33CK256MP508 Internal Op-Amp Motor Control Plug-In Module(PIM).

## 2. SUGGESTED DEMONSTRATION REQUIREMENTS

### 2.1. Motor Control Application Firmware Required for the Demonstration

- `AN1299_dsPIC33CK256MP508_EXT_INT_OPAMP_MCLV2_MCHV2_MCHV3.zip`

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

### 2.2. Software Tools Used for Testing the firmware

- MPLAB® X IDE v5.10
- MPLAB® XC16 Compiler v1.36b
- MPLAB® X IDE Plugin: Data Monitor and Control Interface (DMCI) v2.71
- MPLAB® X IDE Plugin: Latest version of X2CScope Plug-in

> **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

To set up the demonstration, you may use one of the High-Voltage Motor Control Development Boards mentioned below:

- dsPICDEM™ MCHV-2 Development Board (DM330023-2) or
- dsPICDEM™ MCHV-3 Development Board (DM330023-3)

> **Note:**
>
> In this document, hereinafter High-Voltage Motor Control Development Board selected for setting up the demonstration is referred as Development Board**.**

- High Voltage 3-Phase Permanent Magnet Synchronous Motor (AC300025)
- dsPIC33CK256MP508 Internal Op-Amp Motor Control Plug-in module (MA330041-2)

> **Note:**
>
> All items listed under the section 2.3. Hardware Tools Required for the Demonstration are available at microchip DIRECT.

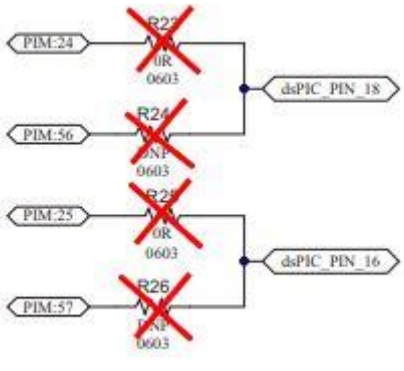
## 3. HARDWARE SETUP

This section describes hardware setup required for the demonstration. Motor phase current feedbacks needed by the firmware are amplified by the operational amplifiers that are internal to the dsPIC33CK256MP508. This is referred as '*Internal amplifier configuration*'.

Refer *dsPICDEM™ MCHV-2 Development Board User's Guide or dsPICDEM™ MCHV-3 Development Board User's Guide*, for any clarification while setting up the hardware.

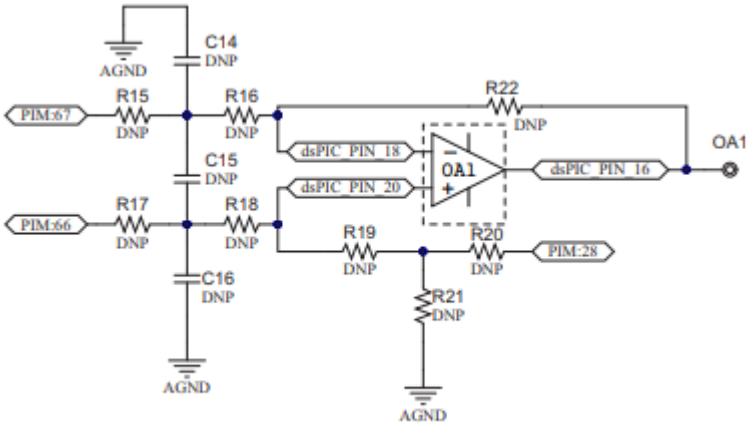
1. By default, internal op-amp which is used for Bus current measurement is not configured. Refer **dsPIC33CK256MP508 Motor Control PIM for Internal Op Amp Configuration.pdf** for more information. Please do as per below note to configure OP-AMP1.

    - As shown in the diagram remove R23, R24, R25, R26 to enable IBUS current to configure Op-AMP1 for amplifying Bus Current.



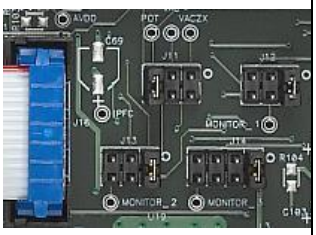

    - Also populate the input and gain resistors of Op-Amp 1

        R22 = R19 = 4.99KΩ
        R20 = 0Ω
        R15 = R17 = 300Ω
        R16 = R18 = 33 Ω
        C15 = 1000pF
        C14, C16, R21 = DNP (Do not populate)




2. ***Before making any connection, verify that the Development Board is not powered and it is fully discharged. This can be done by checking if Power on Status LED D13(Red) is off.***

3. Open the top cover of the enclosure and set up the following jumpers (if they are not in specified positions):

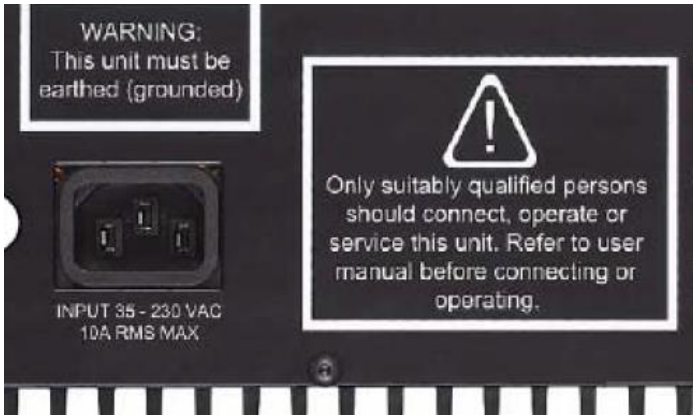
| Jumper | Pins to Short | Board Reference | Remarks |
|---|---|---|---|
| J11 | 3-4 |  | These Jumpers are present on the Development Board.*These can be accessed only after opening the top cover of the enclosure.* |
| J12 | 1-2 | | |
| J13 | 1-2 | | |
| J14 | 1-2 | | |
| PWM OUTPUTS | ENABLE position |  | These Jumpers can be accessed without opening the enclosure, from the front side of the board(or enclosure). |
| USB | FOR USB position |  | |

4. Connect the three phase wires from the motor to M1, M2, and M3 terminals of connector J17(there is no specific order), provided on the Development Board.


5. Insert the ‘Internal Op Amp Configuration Matrix board’ into matrix board header J4. Ensure the matrix board is correctly oriented before proceeding.

6. Insert the dsPIC33CK256MP508 Internal Op-Amp Motor Control PIM into the PIM Socket U11 provided on the Development Board. Make sure the PIM is correctly placed and oriented before proceeding.

7. Close the top cover of the enclosure and secure it with screws.
8. Power Cord Connection. Make sure the power cord is disconnected from the AC mains before connecting the female terminal of the power cable to the AC input connector J1 of the Development Board.




9. To program the device, a mini-USB connection is required between Host PC and the Development Board. Connect a mini-USB cable from your computer to the mini-USB connector “PROGRAM/DEBUG” of the Development Board. The development board features a Built-in isolated Programmer or Debugguer (Microchip Starter Kit).


10. Power up the Development Board by connecting power cord to the mains. To verify the unit is powered, make sure LEDs D6, D13, D16 and D18 are ON.

# 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 dsPIC33CK256MP508 assembled on the Plug-in Module (PIM). The version of the MPLAB X IDE, MPLAB XC16 Compiler and DMCI plug-in used for testing the firmware are mentioned in the section Motor Control Application Firmware Required for the Demonstration. To get help on

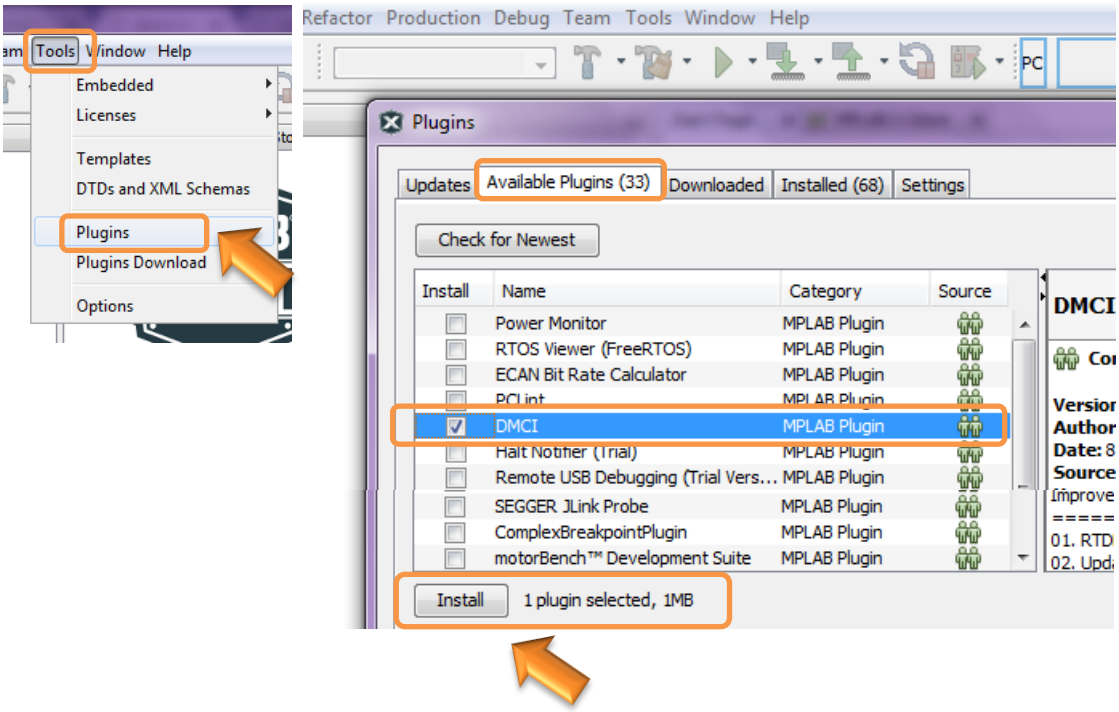
- MPLAB X IDE installation, refer link
- MPLAB XC16 Compiler installation steps, refer link

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 additional details refer 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: Data Monitor and Control Interface (DMCI)

The Data Monitor and Control Interface (DMCI) is a MPLAB X IDE plugin that allows a developer to interact with an application while the application program is running. DMCI provides a graphical user interface which operates within the MPLAB X IDE enabling the developer to examine or modify the contents of application variables without having to halt the application during a debug session. For additional information on DMCI follow the link. To use DMCI, the plugin must be installed:

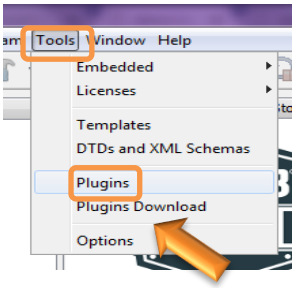
- In MPLAB X IDE, select `Tools>Plugins` and click on the **Available Plugins** tab.
- Select DMCI plug-in by checking its check box, and then click **Install**.
- Look for your tool DMCI under `Tools>Embedded`. If you do not see it, you may need to close and re-open MPLAB X IDE.

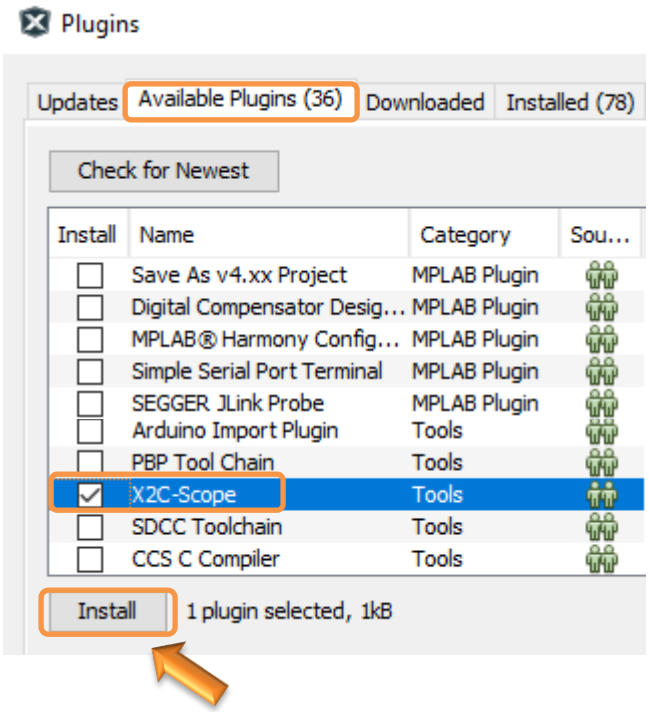


## 4.3. SETUP: X2C - SCOPE

X2C - SCOPE is a 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 (for motor control) in real time. It communicates with the target using the UART. To use X2C, 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 then click 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 section Motor Control Application Firmware Required for the Demonstration.

This firmware is implemented to work on Microchip's 16-bit Digital signal controller (dsPIC® DSC) dsPIC33CK256MP508. For more information, see the *dsPIC33CK256MP508 Family datasheet (DS70005349)*.

The Motor Control Demo application uses push button to start or stop the motor and potentiometer to vary speed of the motor.

This Motor Control Demo Application configures and uses peripherals like PWM, ADC, Op-Amp, UART etc. required for implementing Sensor-less Field Oriented Control (FOC) of Permanent Magnet Synchronous Motor (PMSM) based on the motor control application AN1299 & AN1292.

For more details refer Microchip Application note AN1299 "Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM" and AN1292 *"Sensorless Field Oriented Control(FOC) for a Permanent Magnet Synchronous Motor(PMSM) using a PLL Estimator and Field Weakening(FW)"* available at Microchip web site

**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:

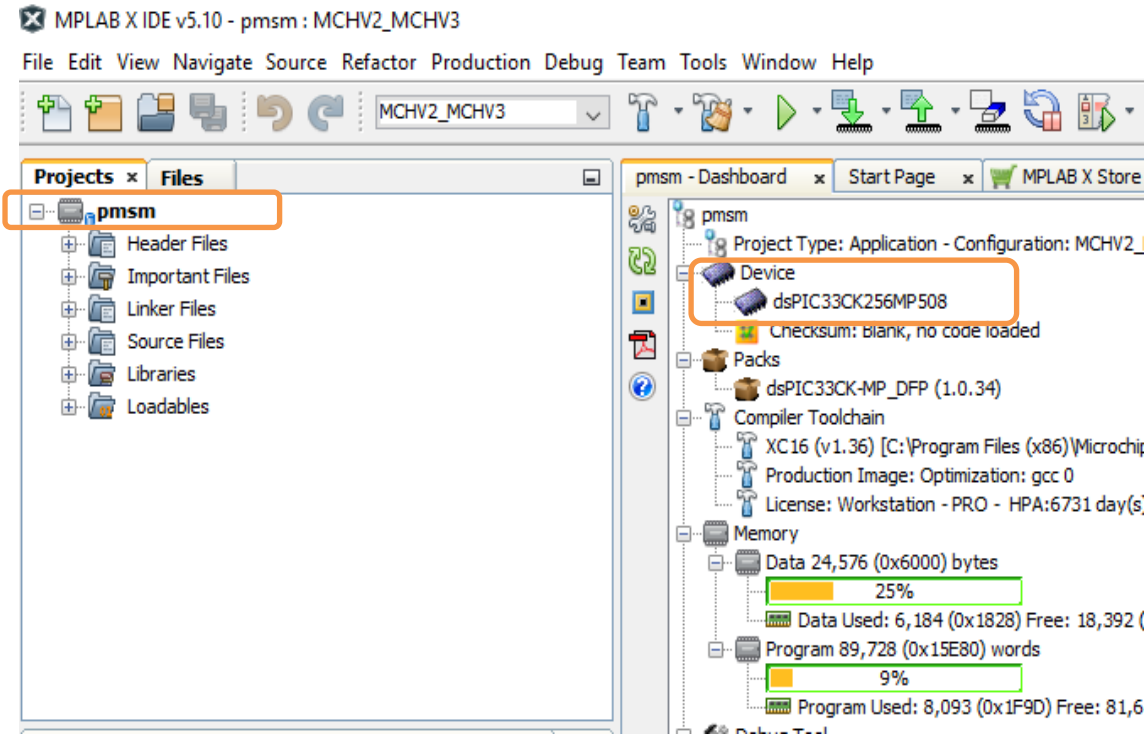
- Shorten the name of the directory containing the firmware used in this demonstration. In this case, rename directory `AN1299_dsPIC33CK256MP508_EXT_INT_OPAMP_MCLV2_MCHV2_MCHV3` to more appropriate shorter name. In case you renamed the directory, consider the new name while reading instructions provided 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.

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

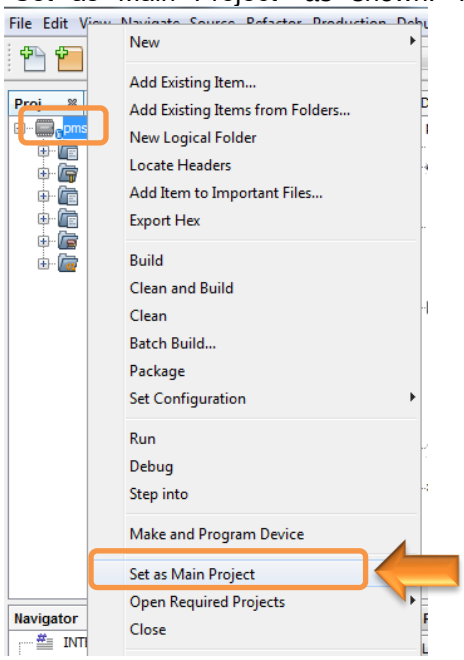
## 5.2. Basic Demonstration

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

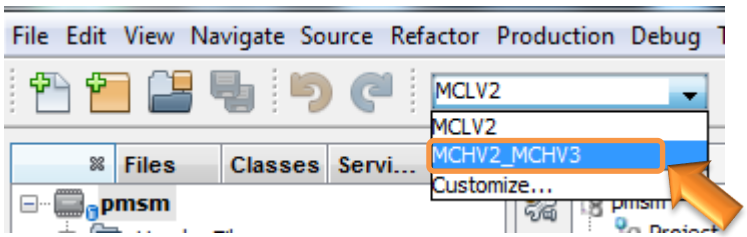
1. Start MPLAB X IDE and open (`File>Open Project`) the project *`pmsm.X`* (*`..\AN1299_dsPIC33CK256MP508_EXT_INT_OPAMP_MCLV2_MCHV2_MCHV3\pmsm.X`*) with device selection *dsPIC33CK256MP508*




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



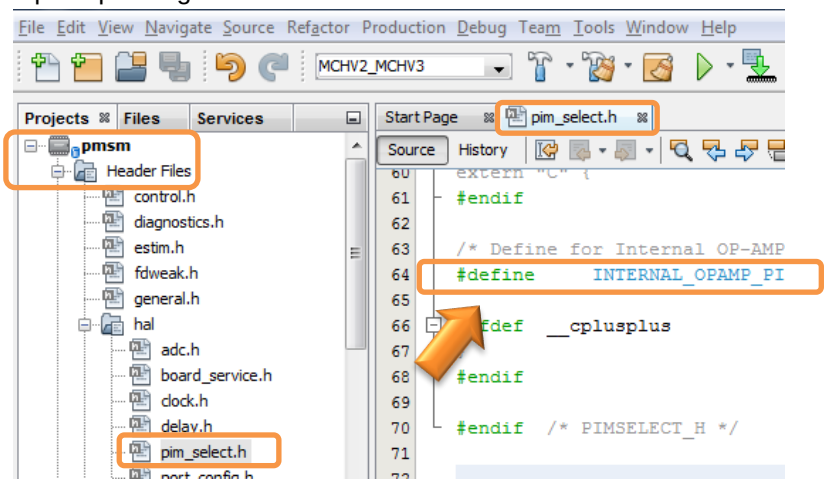
3. Select project configuration as "MCHV2_MCHV3" from the Project Configuration drop down box on the toolbar as shown:



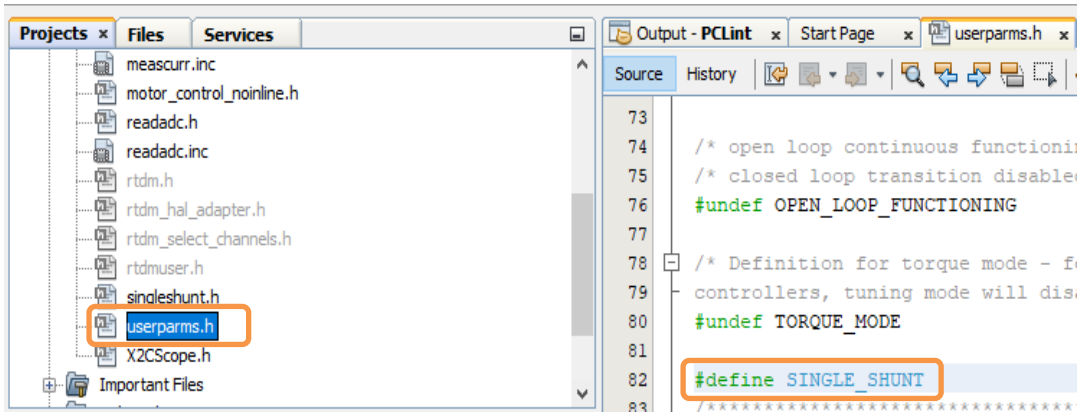

As shown in the figure above, there may be multiple project configurations available for *pmsm.X*.

4. Open userparams.h (under pmsm.X -> headerfiles) in the project pmsm.X and ensure that TUNING, OPEN_LOOP_FUNCTIONING, and TORQUE_MODE is not defined.

5. Open pim_select.h (under pmsm.X->headerfiles->hal) in the project pmsm.X and ensure INTERNAL_OPAMP_PIM is defined as this demonstration is for Internal Op-Amp configuration.



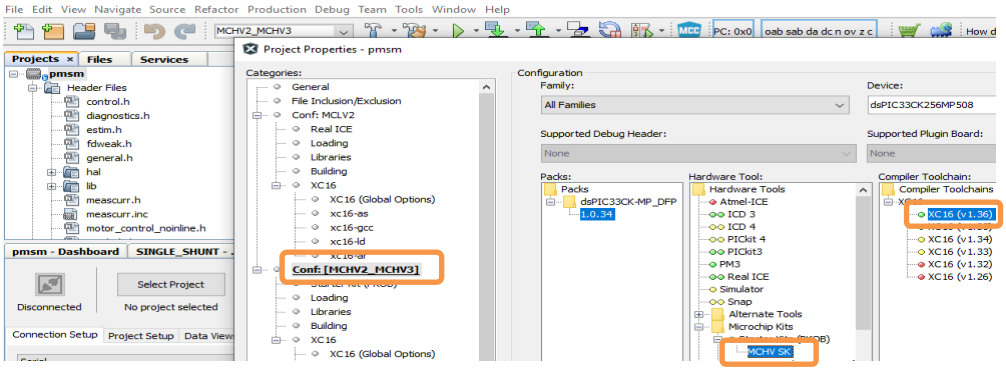

6. Open userparams.h (under pmsm.X -> headerfiles) in the project pmsm.X and ensure SINGLE_SHUNT is defined as this demonstration is for single shunt configuration. Undef SINGLE_SHUNT to work with dual shunt configuration.



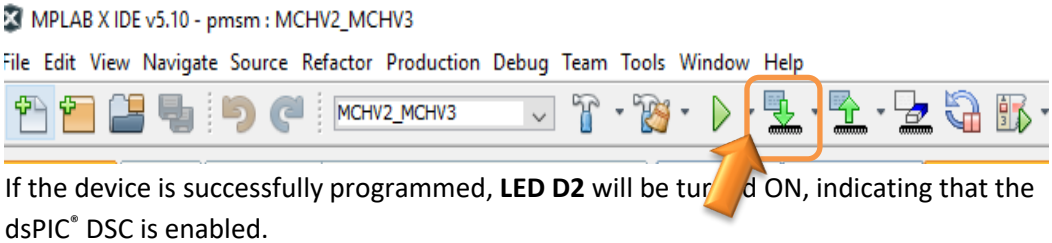

7. Right click on the project *pmsm.X* and select "Properties" to open its Project Properties Dialog. Click the "Conf: [MCHV2_MCHV3]" category to reveal the general project configuration information.

In the ***'Conf-MCHV2_MCHV3'*** category window:

- Select the specific Compiler Toolchain from the available list of compilers. Please ensure MPLAB® XC16 Compiler supports the device dsPIC33CK256MP508.In this case “XC16(v1.36)” is selected. The compiler used for testing the firmware is listed in the section 2.2 Software Tools Used for Testing the firmware.
- lect the Hardware Tool to be used for programming and debugging. In this case, “MCHV-SK” is selected as the programmer from Microchip Starter Kits section.
- After selecting Hardware Tool and Compiler Toolchain, click button **Apply.**




8. To build the project (in this case *pmsm.X*) and program the device dsPIC33CK256MP508, click “**Make and Program Device Main project**” on the toolbar.




If the device is successfully programmed, **LED D2** will be turned ON, indicating that the dsPIC® DSC is enabled.

9. If the device is successfully programmed, **LED D2** will be turned ON, indicating that the dsPIC® DSC is enabled.
10. Run or Stop the motor by pressing the push button **S1**(labeled as **“PUSHBUTTON”**) on the front panel of the Board. The function of the pushbutton (Run/Stop of the motor) is indicated by turning ON or OFF **LED D19**.


11. If desired, the motor speed can be varied using the potentiometer (labeled **“POT**”).


12. Press push button **S1**(labeled as **“PUSHBUTTON”** on the front panel of the Board) to stop the motor.

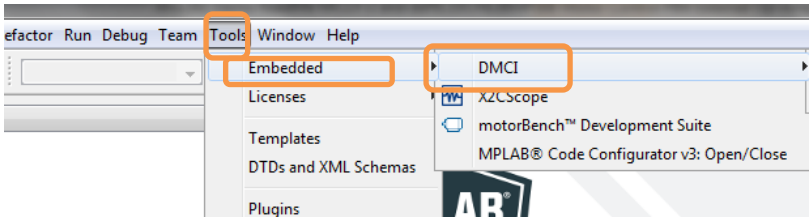
**Note:**

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 DMCI Plug-in of MPLABX

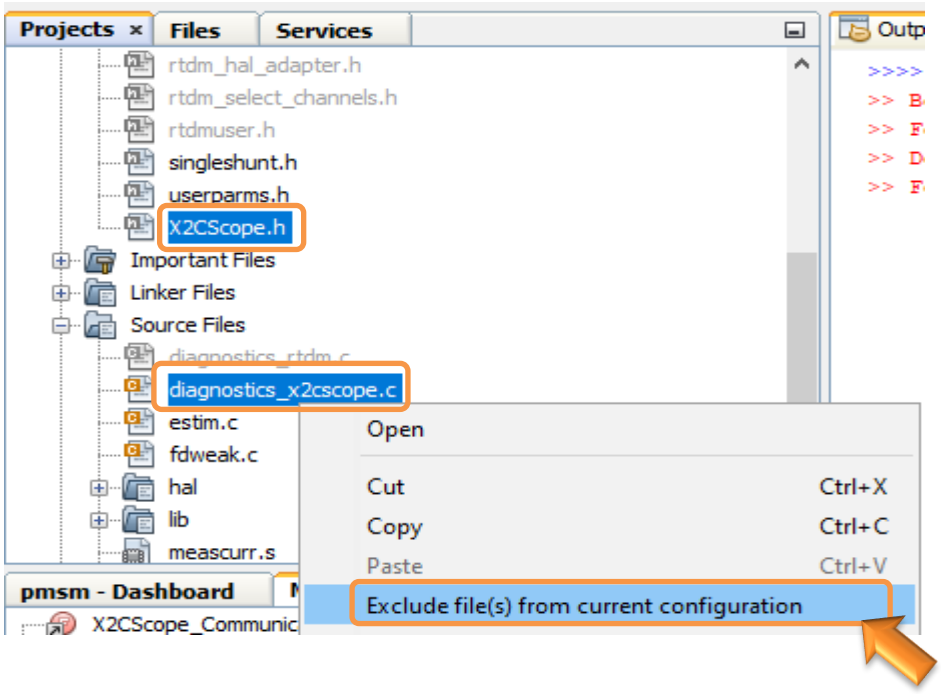
The project firmware comes with the software library for Real Time Data Monitoring (RTDM), needed to interface with DMCI Plug-in available in the MPLAB X IDE. RTDM, along with DMCI creates a communication link between a host PC and a target device for debugging applications in real-time. For additional information on DMCI, click on the link .For additional information on RTDM click on the link.

1. Ensure DMCI Plug-in is installed. Look for DMCI under `Tools>Embedded`. If you do not see it, follow instructions provided in the section Setup: Data Monitor and Control Interface (DMCI) to install the plug-in.

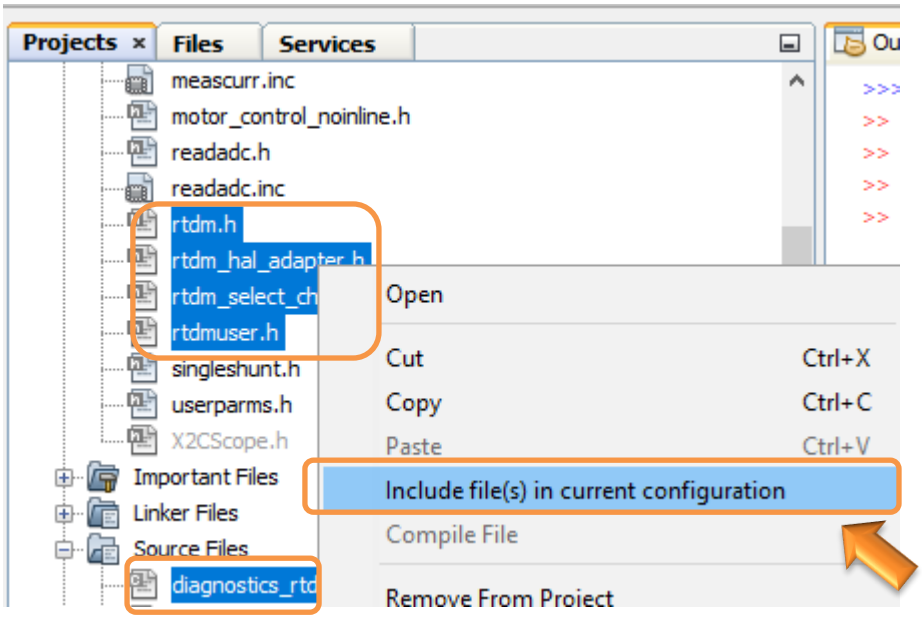



2. To utilize RTDM communication for this demonstration, a USB connection is required between Host PC and the Development Board. Connect a mini-USB cable from your computer to the J6 connector (labeled as “USB” on the front panel of the board enclosure) of the Development Board.

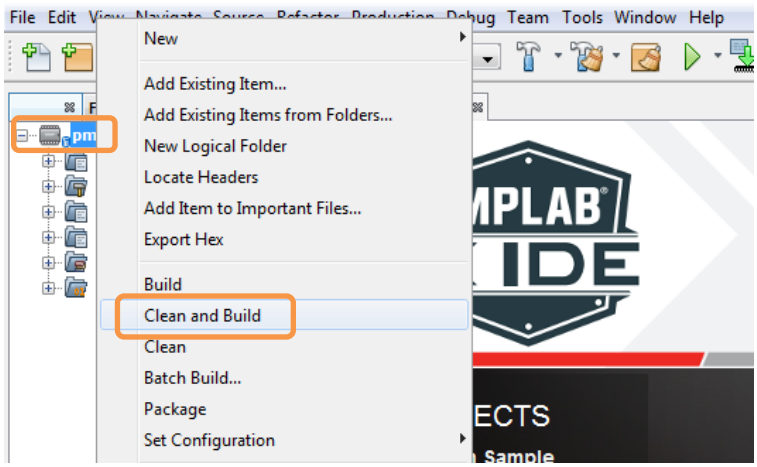

3. Ensure application is configured and running as described under Section Basic Demonstration by following steps 1 through 12.
4. Select files **X2Cscope.h** and **diagnostics_x2cscope.c**, then right click and set Exclude file(s) from current configuration.



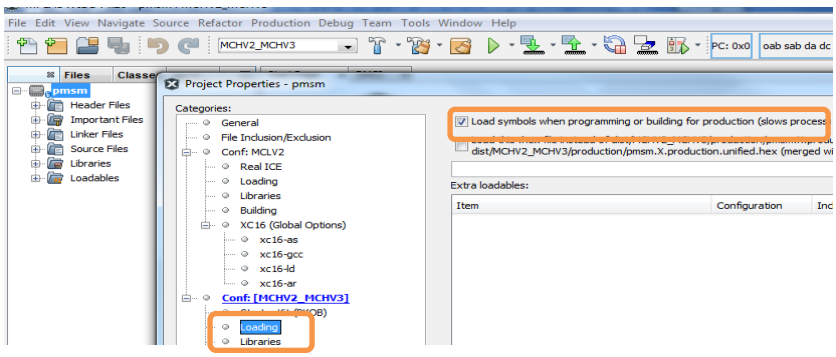
5. Select files rtdm.h , rtdm_hal_adapter.h , rtdm_select_channels.h , rtdmuser.h,and diagnostics_rtdm.c , then rightclick and set Include file(s) from current configuration to add these files as part of current Project Configuration. This will allow RTDM interface related files to be added to the project, required to enable RTDM interface.



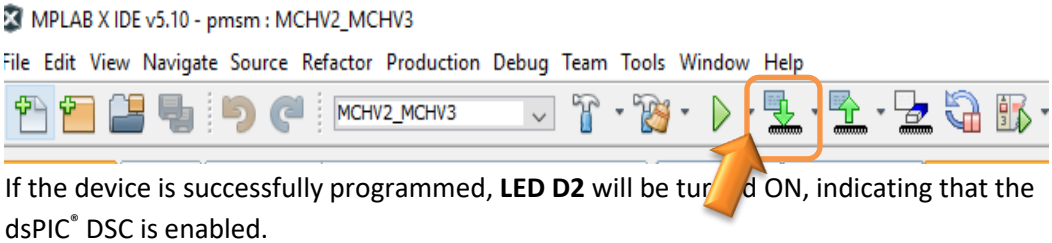

6. Build the project *pmsm.X*. To do that right click on the project *pmsm.X* and select “Clean and Build”.




7. 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 Property dialog

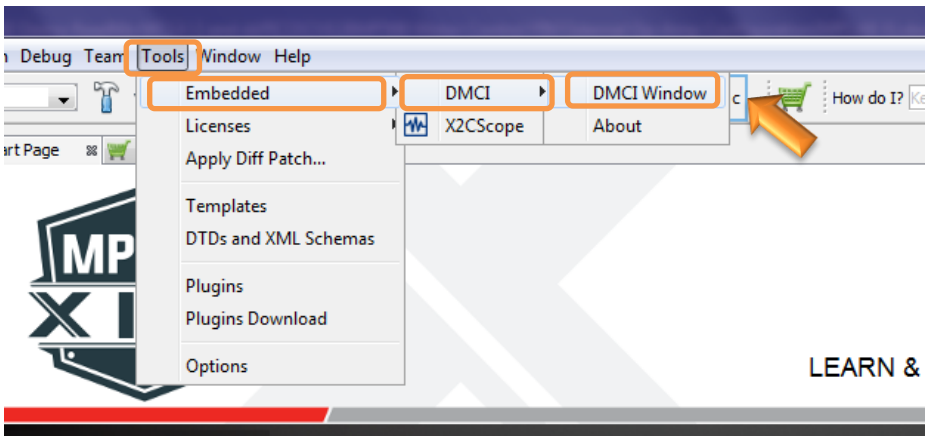


8. To build the project (in this case *pmsm.X*) and program the device dsPIC33CK256MP508, click "**Make and Program Device Main project**" on the toolbar.

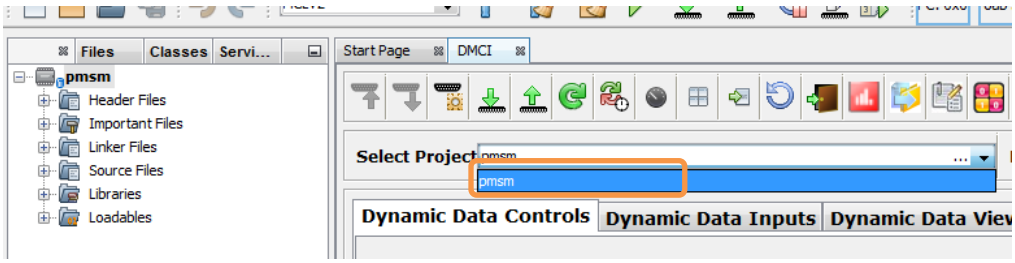



If the device is successfully programmed, **LED D2** will be turned ON, indicating that the dsPIC® DSC is enabled.

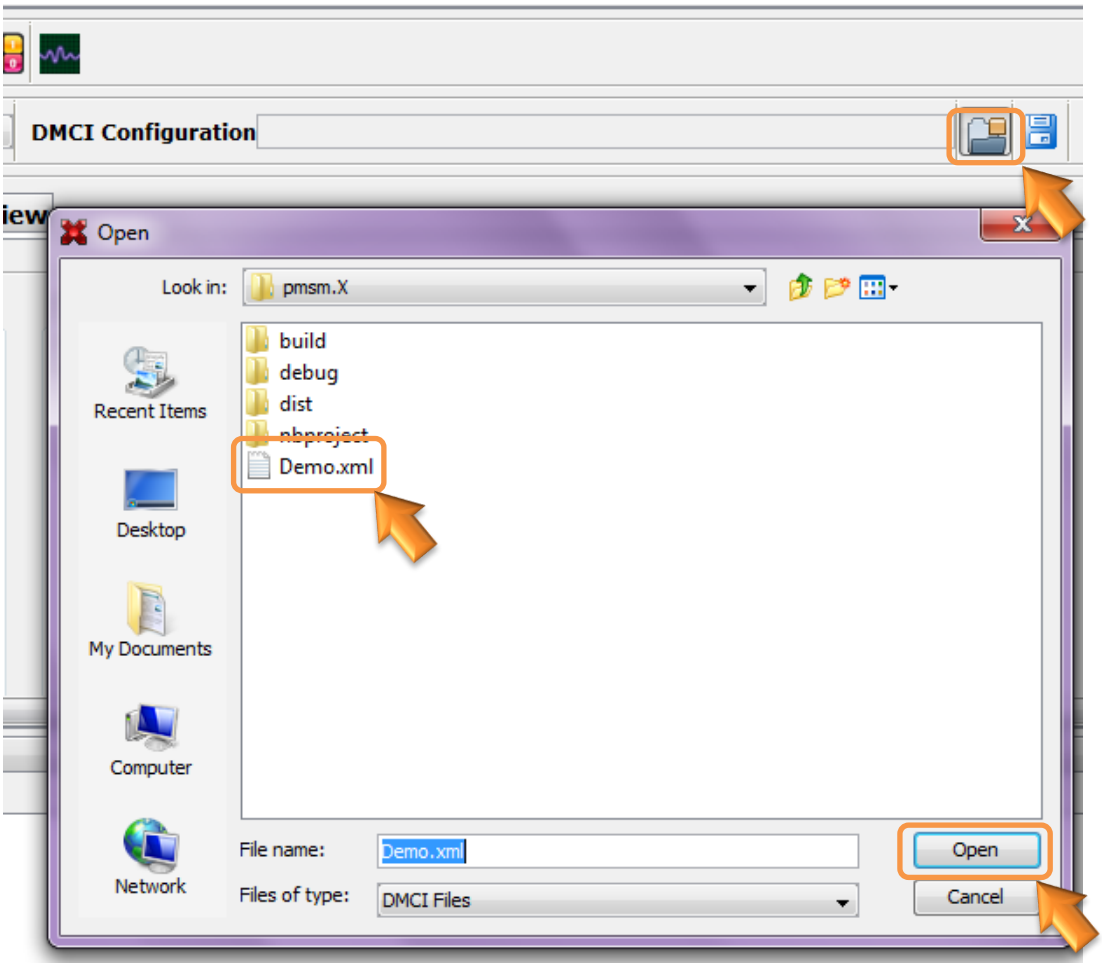
9. Open the DMCI window by selecting `Tools>Embedded>DMCI>DMCI Window.`



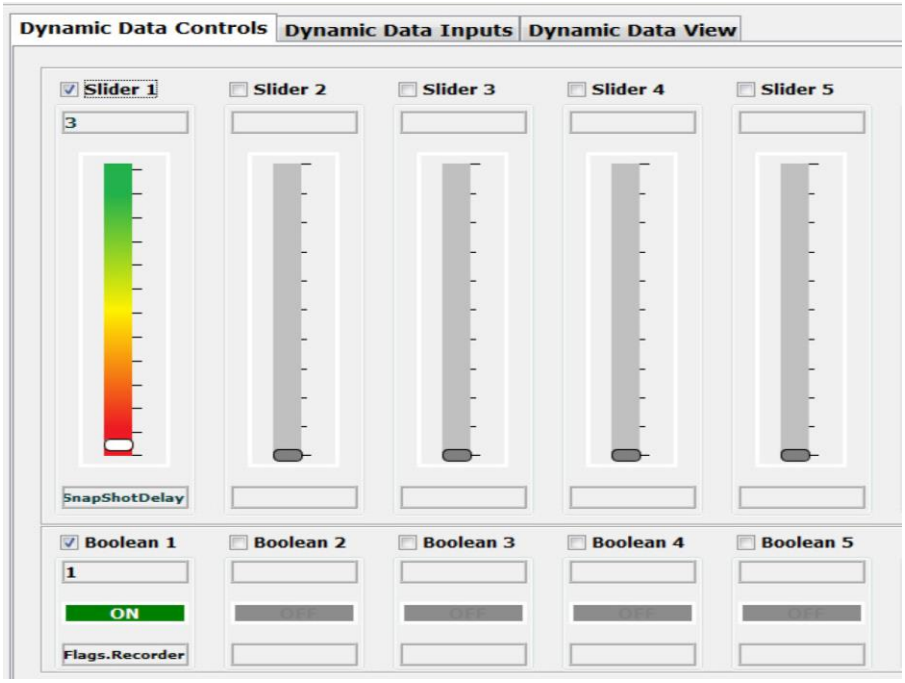

10. From the Select Project drop down menu availble in the DMCI window , select project '*pmsm*'



11. Click the **Load Profile** icon, and load `Demo.xml` from the directory where project *`pmsm.X`* is located. The `Demo.xml` file contains a previously configured profile.

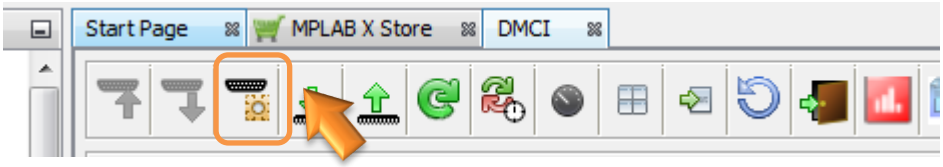



12. The DMCI window appears as follows:

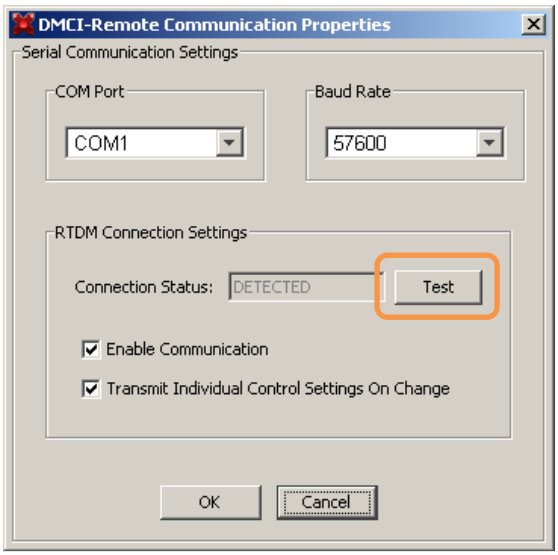


Please refer to the *"Real-Time Data Monitor User's Guide"* (DS70567) for additional settings needed for a RDTM connection.

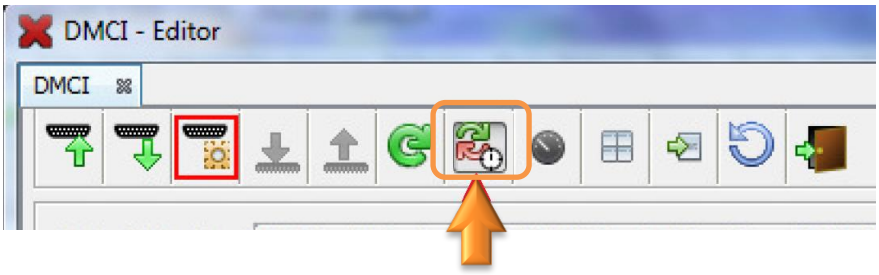
13. Click **Serial Settings** to connect RTDM with your computer..



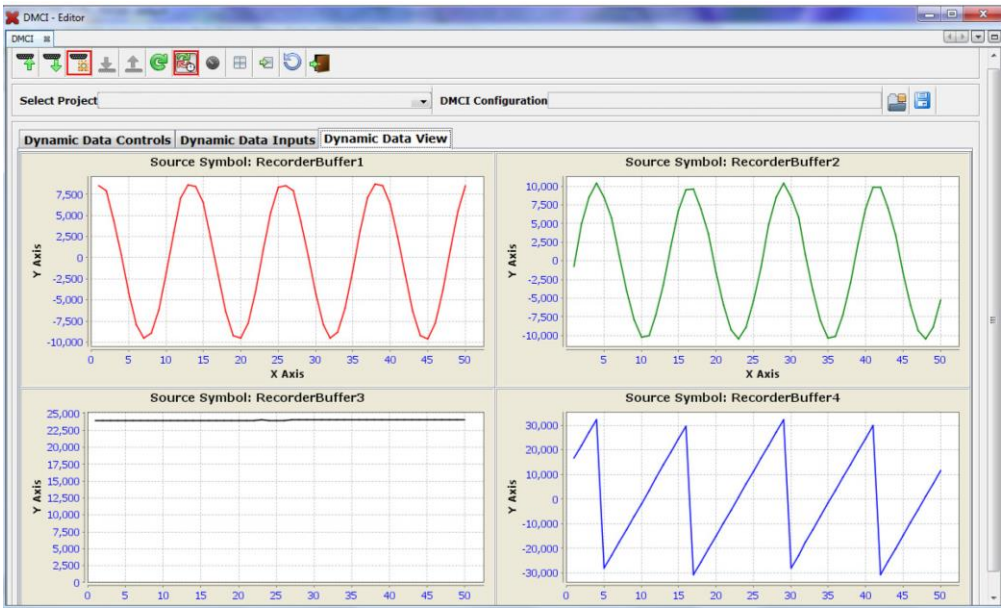

14. Remote Communication needs to be established, as indicated in the following figure (the communication baud rate should be set to 57600, while the COM port used depends on the your connection. Click on Test Box to detect the communication link.



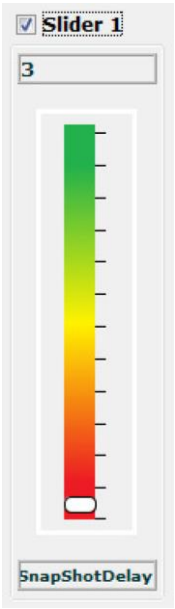

15. Once communication is detected, make sure the "Enable Communication" box is checked and click **OK**.
16. To plot variables in real time, enable "Automated Event Control" by clicking **Automatic Event Execution** icon found on the toolbar.



17. The DMCI window shows variables plotted in real time, which is updated automatically.




18. To change the time window to see data over larger time span, change the value of the `'SnapShotDelay'`, which controls how the buffers are being filled.



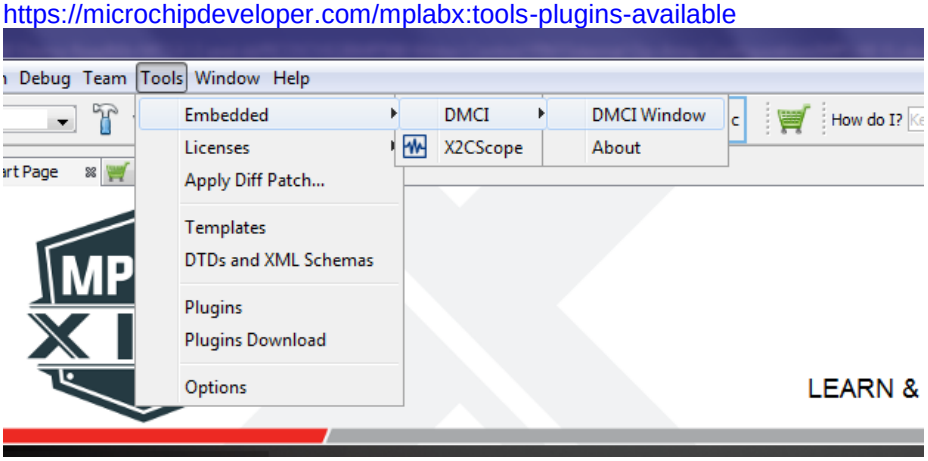

19. The variables displayed through DMCI/RTDM plug-in is set in rtdm_select_channels.c,user may change the varibles to be displayed by modifying the address assigned to the arrays addresses[0],addresses[1],addresses[2] and addresses[3].The array addresses[0] corresponds to Graph1, addresses[1] corresponds to Graph2 ,and so on.

```
void RTDM_SelectChannels(const volatile int16_t *addresses[4])
{
    addresses[0] = &iabc.a;
    addresses[1] = &iabc.b;
    addresses[2] = &estimator.qVelEstim;
    addresses[3] = &estimator.qRho;
}
```

## 6. Data visualization through X2CScope Plug-in of MPLABX

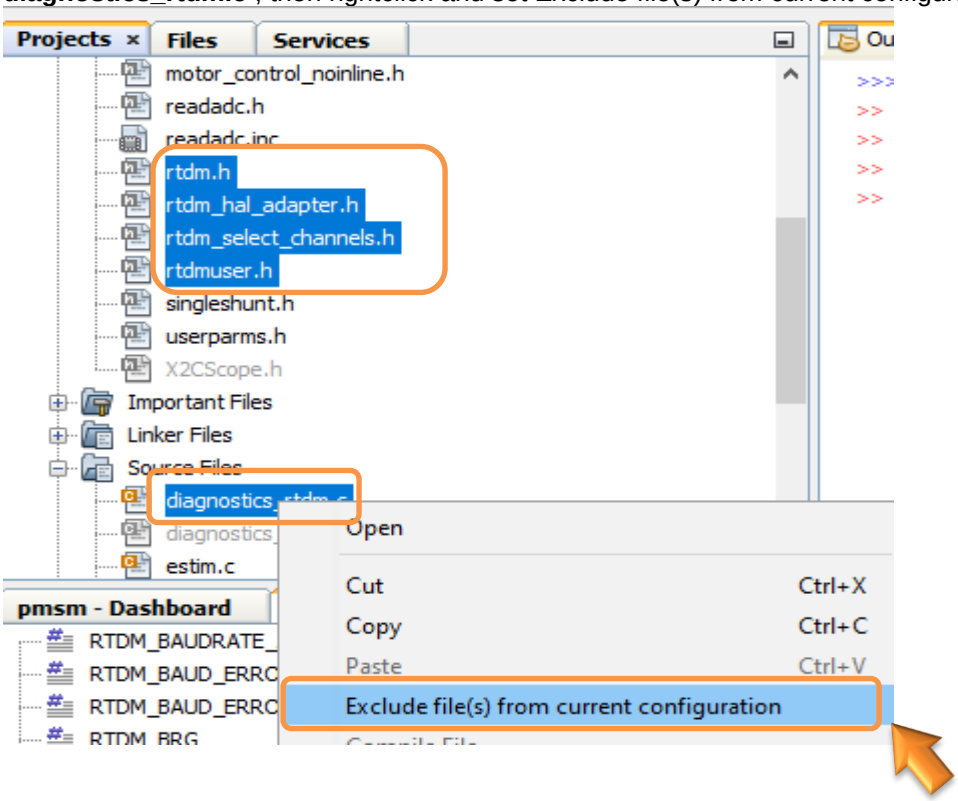
The application firmware comes with initialization required to interface Controller with Host PC to enable Data visualization through X2C Scope plug-in. X2C-Scope is a third-party plugin for MPLAB X which facilitates real-time diagnostics.

1. Ensure X2C Scope Plug-in is installed. For additional information on how to set up a plug-in refer to
   https://microchipdeveloper.com/mplabx:tools-plugins-available

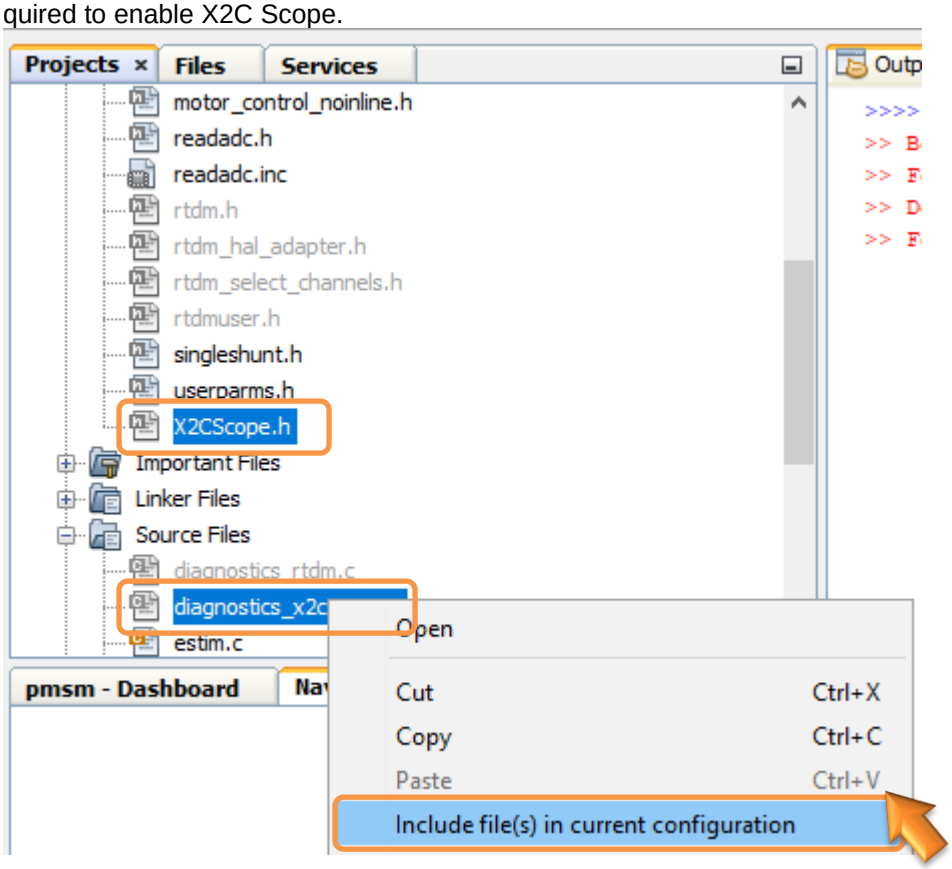



2. To utilize X2-C communication for this demonstration, a mini-USB connection is required between Host PC and dsPICDEM™ MCLV-2 Development Board. Connect a mini-USB cable from your computer to the J8 connector of the dsPICDEM™ MCLV-2 Development Board.

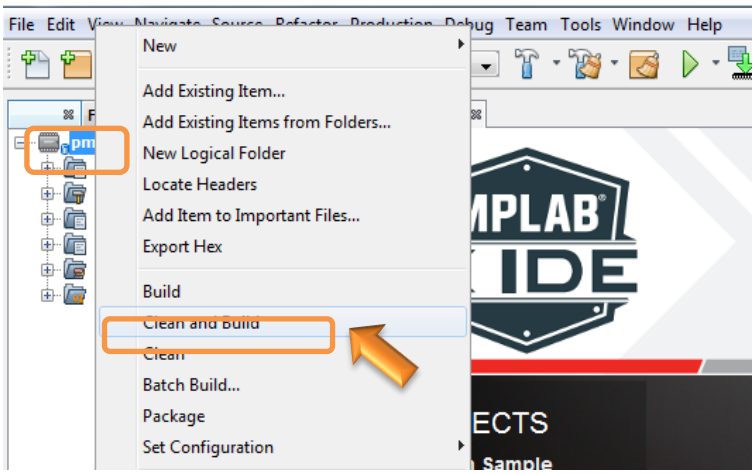
3. Ensure application is configured and running as described under Section Basic Demonstration by following steps 1 through 12.
4. Select files **rtdm.h** , **rtdm_hal_adapter.h** , **rtdm_select_channels.**h , **rtdmuser.h**,and **diagnostics_rtdm.c** , then rightclick and set Exclude file(s) from current configuration.



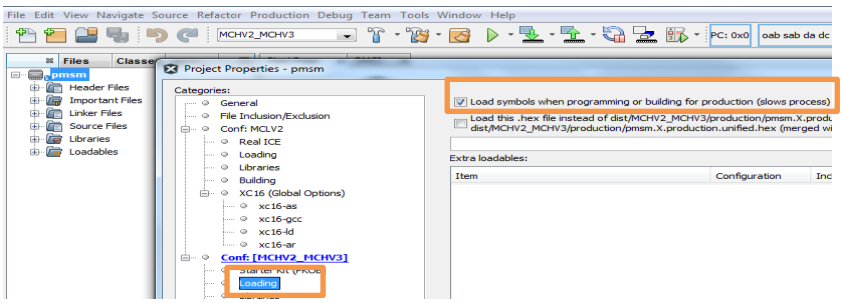
5. Select files **X2Cscope.h and diagnostics_x2cscope.c**, then right click and set **Include file(s) from current configuration** to add these files as part of current Project Configuration. This will allow X2C Scope interface related files to be added to the project and is required to enable X2C Scope.



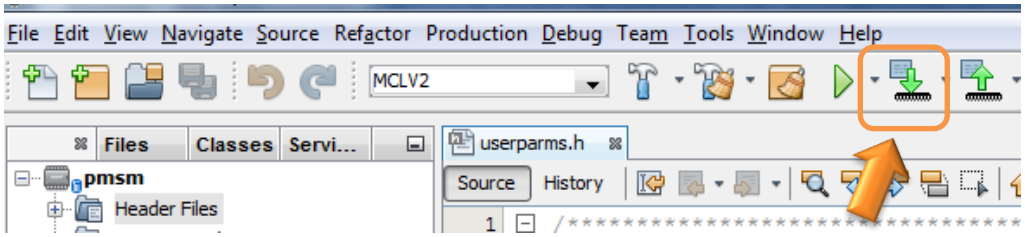

6. Build the project *pmsm.X*. To do that right click on the project *pmsm.X* and select “Clean and Build”.



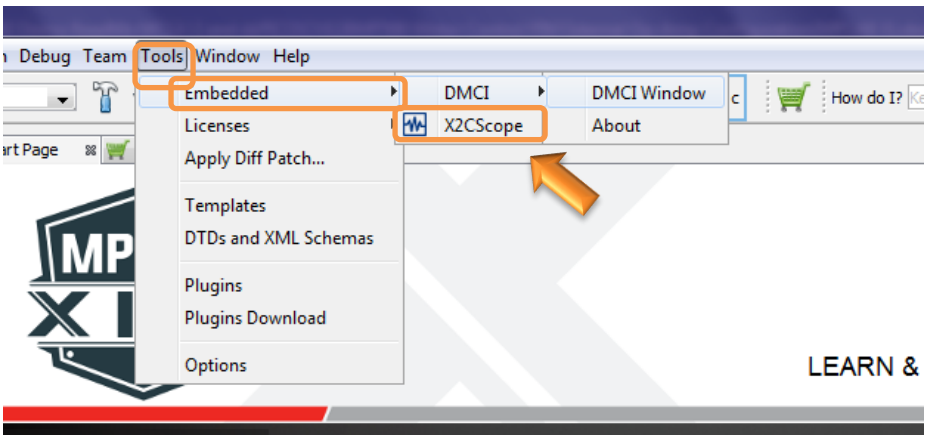

7. Please ensure that the checkbox “Load symbols when programming or building for production (slows process)” is checked, which is under the “Loading” category of the Project Property dialog.



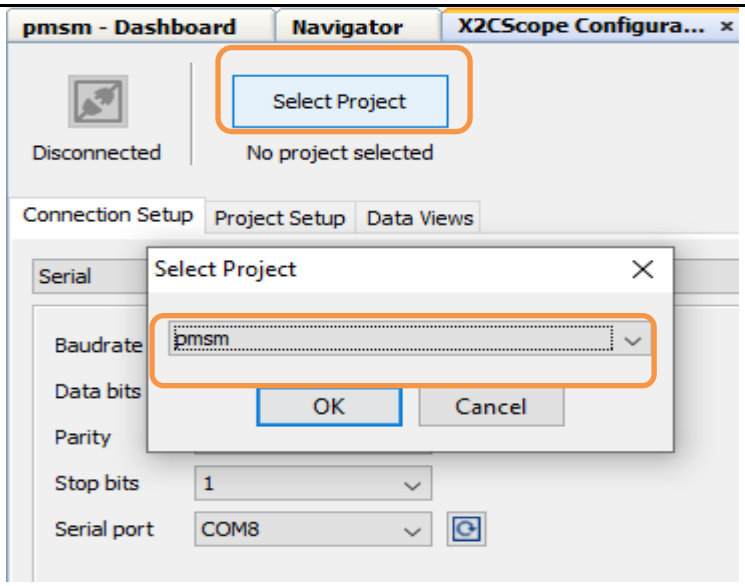

8. To build the project (in this case *pmsm.X*) and program the device dsPIC33CK256MP508, click "**Make and Program Device Main project**" on the toolbar.



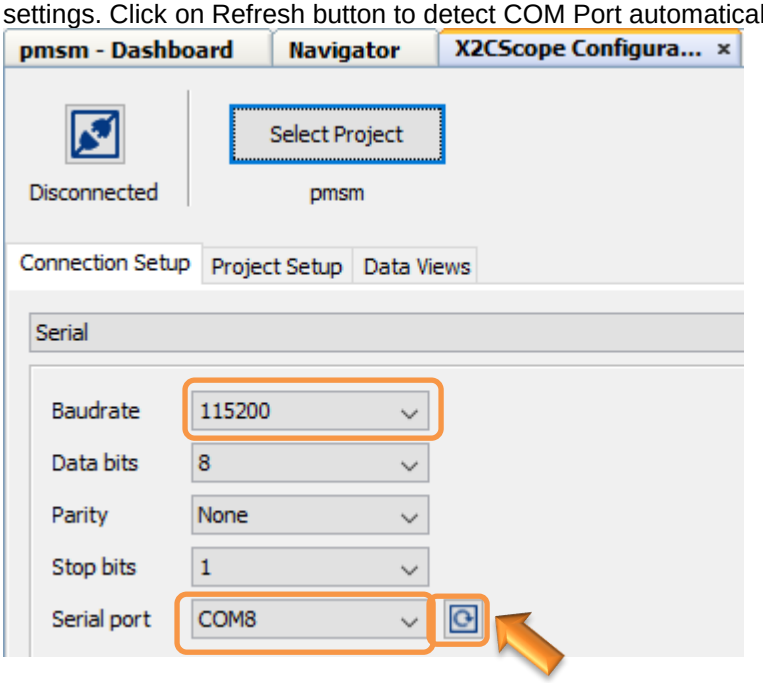

9. Open the X2C window by selecting `Tools>Embedded>X2CScope`.



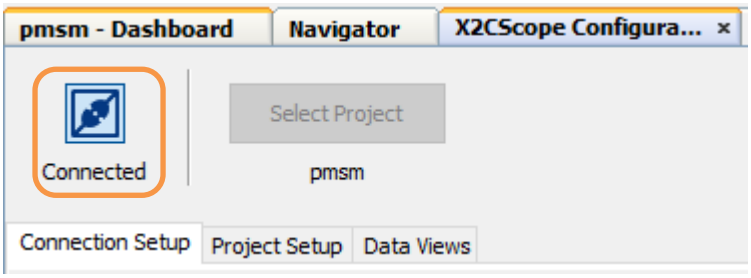

10. Open the X2CScope Configuration window and in "Select project" menu, select pmsm.X project as shown.



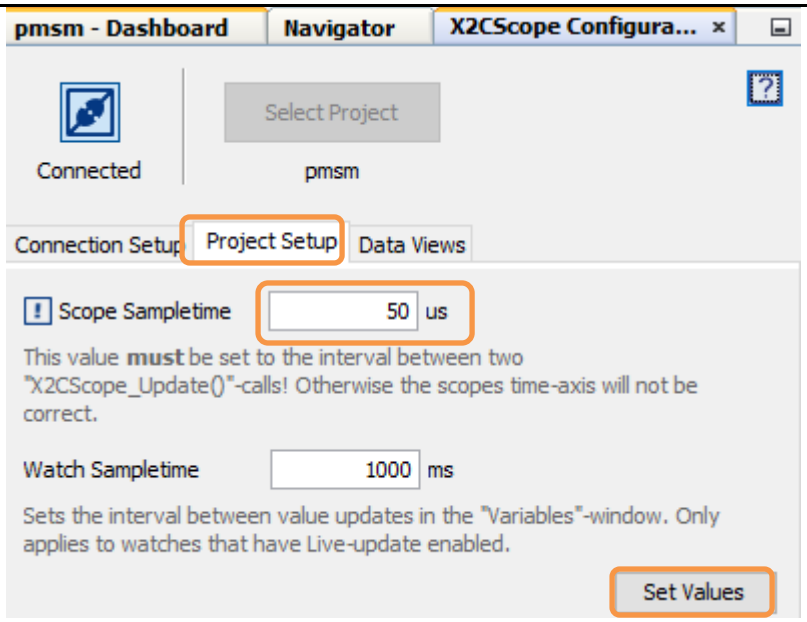

11. Remote Communication needs to be established, as indicated in the following figure (the communication baud rate should set to 115200, while COM port used depends on your settings. Click on Refresh button to detect COM Port automatically.



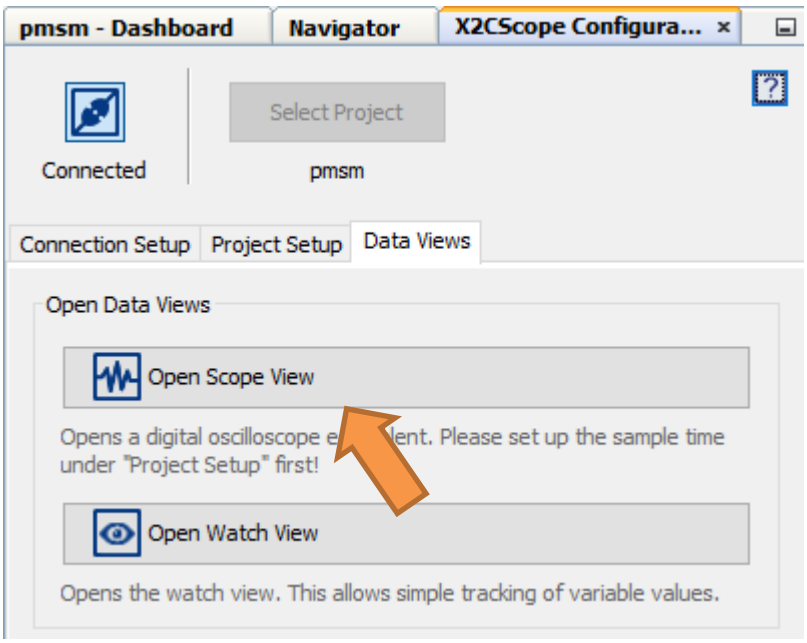

12. Once COM port detected, click on "**Disconnected**", and it will be turn into "**Connected**", if the link is established as programmed.




13. Set the "Project Setup" as shown below and click "Set Values". Set Scope sampling time as interval at which X2CScopeUpdate() is called. In this application it is every 20kHz (50Us).

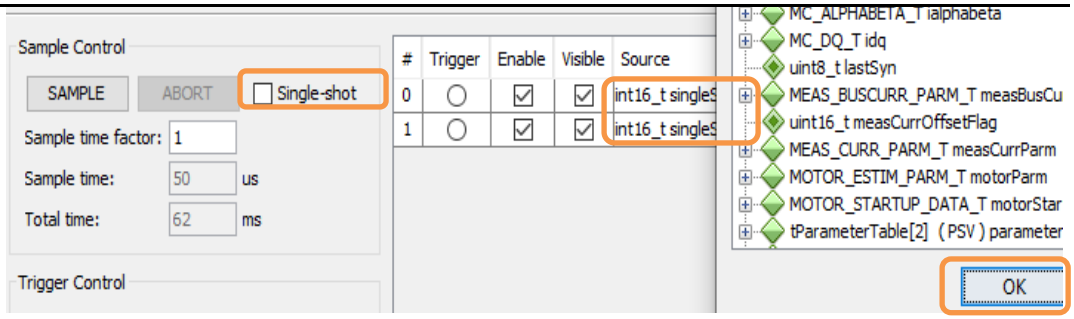



14. When the setup is established, click on open scope View (under sub window “Data Views”), this open Scope Window.

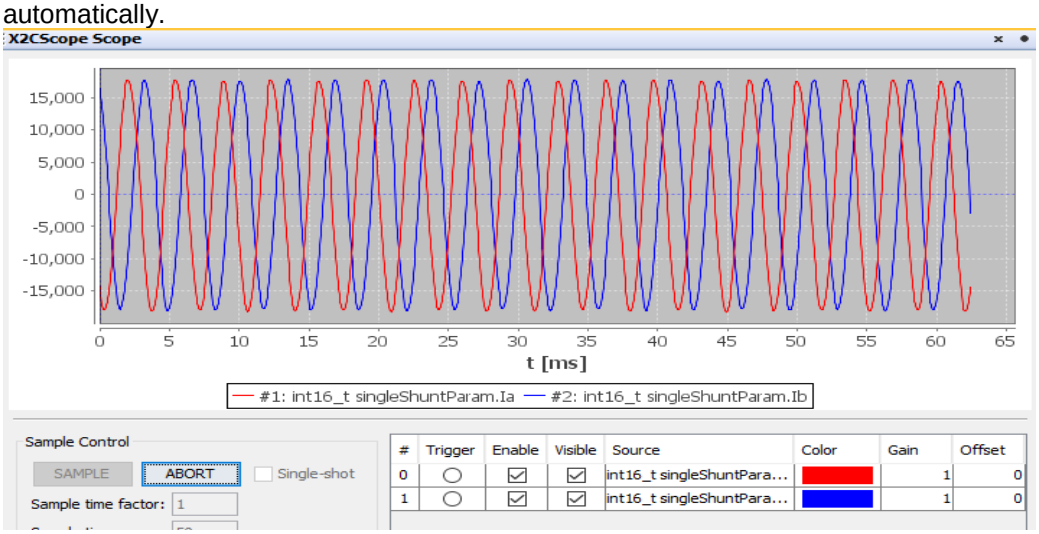



15. In the window select the variables user may want to watch. Click on the source, a window Select Variables opens upon the screen. From the select variables list, choose the variable that you want to view. Then ensure Enable, Visible check boxes are checked as shown.

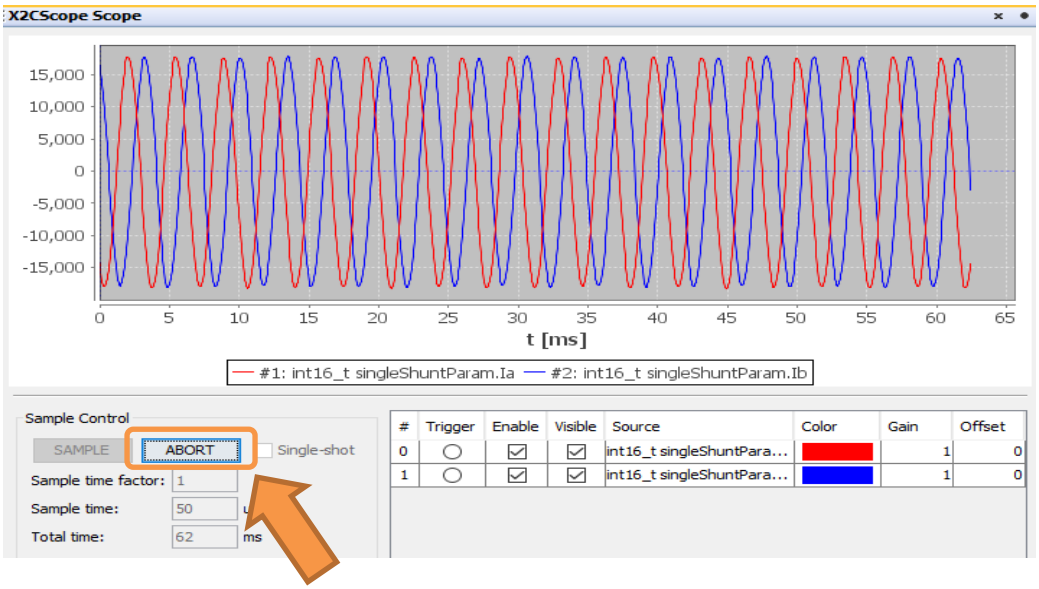
    To view data plots continuously uncheck Single-shot. When Single-shot is checked it captures the data once and stops, if trigger occurs. Change the Sample time factor value to change the time window.




16. Click on SAMPLE, then X2C scope window shows variables in real time, which is updated automatically.




17. Click on ABORT to stop.




# 7. dsPIC® DSC RESOURCE USAGE SUMMARY

## 7.1. Device Pin Mapping and Its Functionality in the Firmware:

The following table summarizes device pins configured and used in the AN1299 motor control application firmware demonstrated using the Development Board and the dsPIC33CK256MP508 Internal Op-Amp Motor Control PIM(MA330041-2). Refer "*dsPIC33CK256MP508 Internal Op-Amp Motor Control Plug-in-Module (PIM) Information Sheet(DS50002756)"* for more information.

| Functional Description | PIM PIN Number | Device PIN Number | Device Pin Name | Signal Type | Remarks |
|---|---|---|---|---|---|
| **Motor Control PWMs and Fault Input** | | | | | |
| PWM1H | PIM:94 | 1 | RP46/**PWM1H**/PMD5/**RB14** | PWM Output | Controls Hex Bridge MOSFET Q5 |
| PWM1L | PIM:93 | 3 | RP47/**PWM1L**/PMD6/**RB15** | PWM Output | Controls Hex Bridge MOSFET Q6 |
| PWM2H | PIM:99 | 78 | TDI/RP44/**PWM2H**/PMD3/**RB12** | PWM Output | Controls Hex Bridge MOSFET Q3 |
| PWM2L | PIM:98 | 80 | RP45/**PWM2L**/PMD4/**RB13** | PWM Output | Controls Hex Bridge MOSFET Q4 |
| PWM3H | PIM:03 | 75 | TMS/RP42/**PWM3H**/PMD1/**RB10** | PWM Output | Controls Hex Bridge MOSFET Q1 |
| PWM3L | PIM:100 | 76 | TCK/RP43/**PWM3L**/PMD2/**RB11** | PWM Output | Controls Hex Bridge MOSFET Q2 |
| FAULT_MC | PIM:18 | 49 | RP72/SDO2/**PCI19**/**RD8** | PWM Input | Connected to Over Current Fault Output |
| **Analog Inputs – Phase Currents, Speed Reference** | | | | | |
| POT | PIM:32 | 36 | **AN19**/CMP2C/RP75/PMA0/PMALL/PSA0/**RD11** | Analog Input | Speed Reference Connected to Potentiometer POT1 |
| IBUS+ | PIM:66 | 20 | OA1IN+/AN9/PMA6/RA2 | Analog Input | Op-Amp 1 Non-Inverting Input (Internal to dsPIC33CK256MP508) |
| IBUS- | PIM:67 | 18 | OA1IN-/ANA1/RA1 | Analog Input | Op-Amp 1 Inverting Input (Internal to dsPIC33CK256MP508) |
| IBUS (Amplified Bus Current) | Not Applicable | 16 | OA1OUT/AN0/CMP1A/IBIAS0/RA0 | Analog Input | Op-Amp 1 Output (Internally connected to dsPIC33CK256MP508's ADC) |
| IA+ | 74 | 45 | PGC2/**OA2IN+**/RP36/RB4 | Analog Input | Op-Amp 2 Non-Inverting Input (Internal to dsPIC33CK256MP508) |
| IBUS+ | 66 | 43 | PGD2/**OA2IN-**/AN8/RP35/RB3 | Analog Input | Op-Amp 2 Inverting Input (Internal to dsPIC33CK256MP508) |
| IMOTOR1 (Amplified IA) | Not Applicable | 41 | **OA2OUT**/**AN1**/AN7/ANA0/CMP1D/CMP2D/CMP3D/RP34/SCL3/INT0/RB2 | Analog Output | Op-Amp 2 Output (Internally connected to dsPIC33CK256MP508's ADC) |
| IB+ | 73 | 29 | **OA3IN**+/AN14/CMP2B/ISRC1/RP50/PMD13/PMA13/RC2 | Analog Input | Op-Amp 3 Non-Inverting Input (Internal to dsPIC33CK256MP508) |
| IBUS+ | 66 | 28 | **OA3IN**-/AN13/CMP1B/ISRC0/RP49/PMA7/RC1 | Analog Input | Op-Amp 3 Inverting Input (Internal to dsPIC33CK256MP508) |
| IMOTOR2 (Amplified IB) | Not Applicable | 23 | OA3OUT/AN4/CMP3B/IBIAS3/RA4 | Analog Ouput | Op-Amp 3 Output (Internally connected to dsPIC33CK256MP508's ADC) |
| **Miscellaneous Signals** | | | | | |
| BTN_1 | PIM:83 | 54 | RP69/PMA15/PMCS2/**RD5** | Digital Input | Connected to Push Button S2 |
| BTN_2 | PIM:84 | 39 | **RE7** | Digital Input | Connected to Push Button S3 |
| Debug LED1 | PIM:60 | 42 | **RE8** | Digital Output | Connected to LED D17 |
| Debug LED2 | PIM:01 | 44 | **RE9** | Digital Output | Connected to LED D2 |

| | | | | | |
|---|---|---|---|---|---|
| RX (UART) | PIM:49 | 52 | **RP71**/PMD15/**RD7** | UART1 Input | Connected to UART-USB converter to establish serial communication interface between Host PC and the dsPIC® DSC as needed by DMCI-RTDM. |
| TX (UART) | PIM:50 | 53 | **RP70**/PMD14/**RD6** | UART1 Output | Connected to UART-USB converter to establish serial communication interface between Host PC and the dsPIC® DSC as needed by DMCI-RTDM. |

## 8. References:

For additional information, refer following documents or links.

1. AN1299 Application Note “Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM”
2. AN1292 Application Note “Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)”
3. dsPICDEM™ MCHV-2 Development Board User’s Guide(DS52074)
4. dsPICDEM™ MCHV-3 Development Board User’s Guide(DS50002505)
5. dsPIC33CK256MP508 Internal Op-Amp Motor Control Plug-in-Module (PIM) Information Sheet(DS50002756)
6. dsPIC33CK256MP508 Family datasheet(DS70005349).
7. Family Reference manuals (FRM) of dsPIC33CK256MP508 family
8. MPLAB® X IDE User’s Guide (DS50002027) or MPLAB® X IDE help
9. Real-Time Data Monitor User’s Guide (DS70567) or Real Time Data Monitoring Tool - RTDM
10. Data Monitor and Control Interface - Developer Help
11. MPLAB® X IDE installation
12. MPLAB® XC16 Compiler installation