AN1307 Demonstration ReadMe for the dsPICDEM™ MCSM Development Board with the dsPIC33CK64MP105 External Op Amp Motor Control PIM (MPLAB® X IDE)

1. INTRODUCTION

This document describes the setup requirements for running the Stepper motor control, which is referenced in AN1307 "*Stepper Motor Control with dsPIC® DSCs*" using a dsPICDEM™ MCSM Development Board in the External Op Amp configuration.

The demonstration is configured to run on the dsPICDEM™ MCSM Development Board in External Op Amp configuration with the dsPIC33CK64MP105 External Op Amp Motor Control Plug-In Module(PIM).

2. SUGGESTED DEMONSTRATION REQUIREMENTS

2.1. Motor Control Application Firmware Required for the Demonstration

- AN1307_dsPIC33CK64MP105_MCSM.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 v1.36
- MPLAB® X IDE Plugin: Data Monitor and Control Interface (DMCI) v2.71

2.3. Hardware Tools Required for the Demonstration

- dsPICDEM™ MCSM Development Board (DM330022-1)
- 24V Power Supply (AC002013)
- Leadshine Stepper Motor (AC300024)
- dsPIC33CK64MP105 External Op Amp Motor Control Plug-in module (MA330050-1)
- Microchip Programmer tools like MPLAB® REAL ICE™ In-Circuit Emulator (DV244005) or MPLAB® ICD 3 (DV164035) etc.

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

3. HARDWARE SETUP

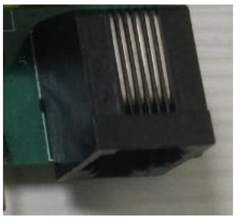
This section describes hardware setup required for the demonstration. The following hardware setup allows the stepper motor control algorithm to run on the dsPICDEM MCSM Development Board Kit.

1. Connect the Stepper Motor to the output header J8 with following motor connection.

J8 Pin Number	J8 Pin Name	Wire Color for Bipolar Parallel Connection
1	NC	---
2	M1	Black + Yellow
3	DC+	---
4	M2	Green + Orange
5	M3	Red + White
6	DC+	---
7	M4	Blue +Brown
8	NC	---

2. Insert the dsPIC33CK64MP105 External Op Amp Motor Control PIM into the PIM Socket U2 provided on the dsPICDEM™ MCSM Development Board. Make sure the PIM is correctly placed and oriented before proceeding.
3. Plug in the 24V power supply to connector J6 provided on the dsPICDEM™ MCSM Development Board.

4. Connect the Microchip programmer/debugger tools like MPLAB REAL ICE™ or MPLAB ICD-3™ to the Connector J1 of the dsPICDEM™ MCSM Development Board and to the Host PC used for programming the device.



5. Using a mini-USB cable, connect the computer to the J4 mini-USB Connector.

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 dsPIC33CK64MP105 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

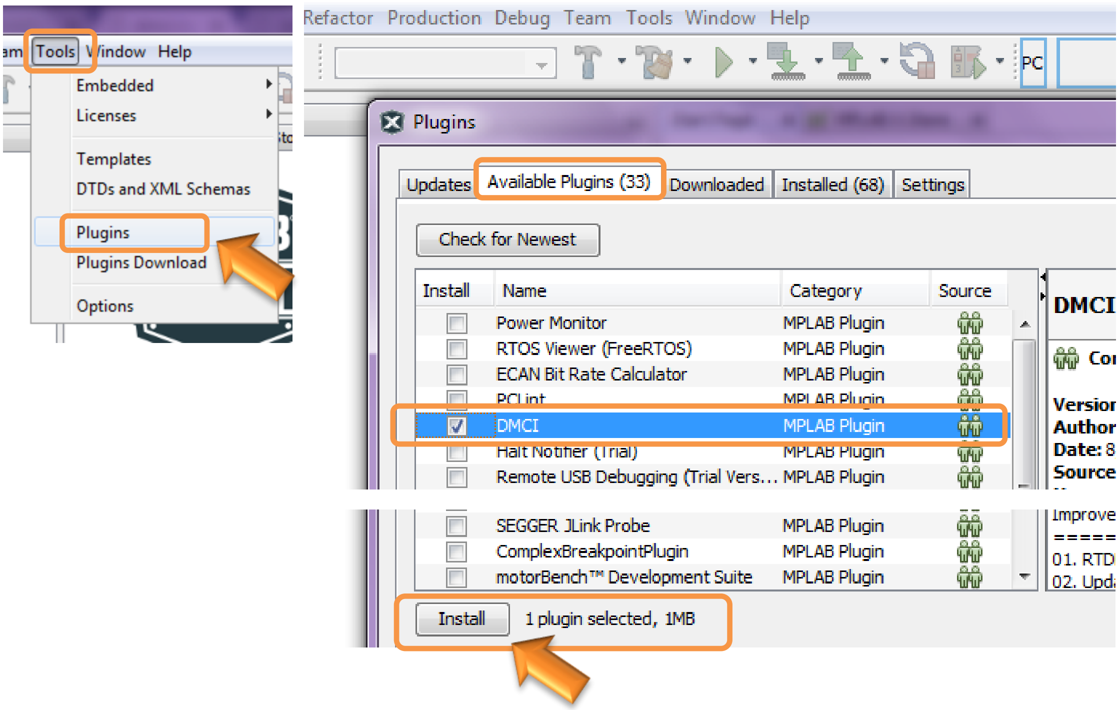
If MPLAB IDE v8 or earlier is already installed on your computer, then run the MPLAB driver switcher (It is installed when MPLAB®X IDE is installed) to switch from MPLAB IDE v8 drivers to

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 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.



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) dsPIC33CK64MP105. For more information, see the *dsPIC33CK64MP105 Family datasheet(DS70005363)*.

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, UART etc. required for implementing Stepper Motor Control based on the motor control application AN1307.

For more details refer Microchip Application note AN1307 *"Stepper Motor Control with dsPIC® DSCs"* 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:

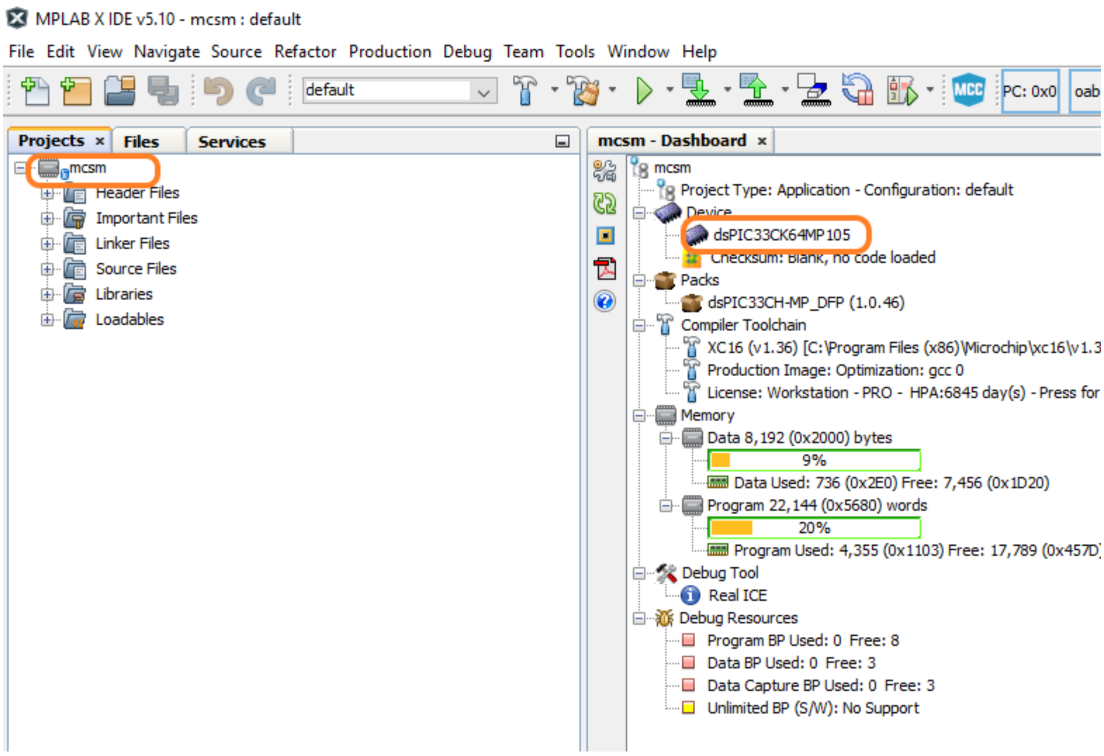
- Shorten the name of the directory containing the firmware used in this demonstration. In this case, rename directory AN1307_dsPIC33CK64MP105_MCSM to more appropriate shorter name. In case you renamed the directory, consider the new name while reading instructions provided in the upcoming sections of 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".*

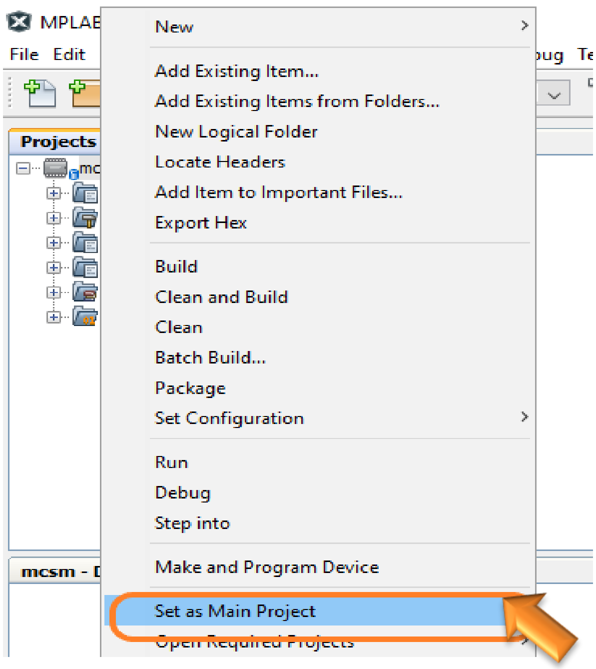
5.2. Basic Demonstration

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 *`mcsm.X`* *`(..\AN1307_dsPIC33CK64MP105_MCSM\mcsm.X)`* with device selection *dsPIC33CK64MP105*.



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



3. Open `userparams.h` (under `mcsm.X -> headerfiles`) in the project `mcsm.X` and ensure that BIPOLAR is defined.

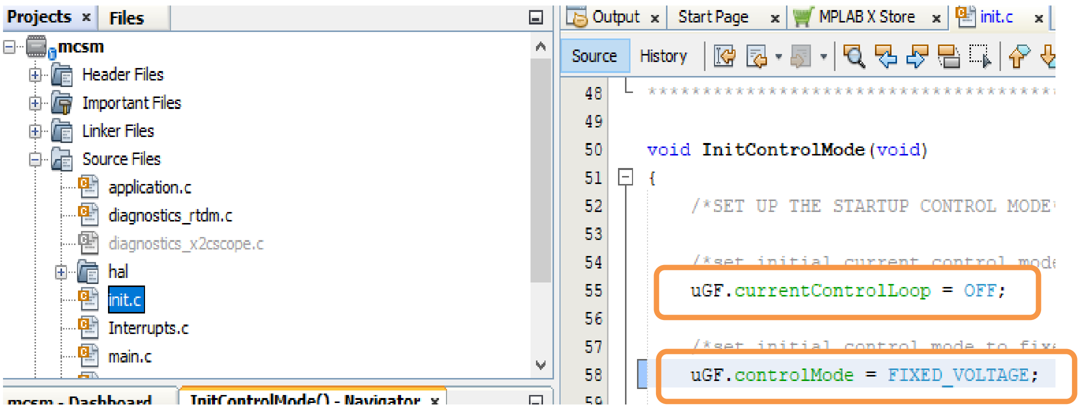
4. Open `int.c` (under `mcsm -> Sourcefiles->init`) in the project `mcsm.X` and select the required control mode.in **InitControlMode()** function.

   - For Open loop Voltage Control, Configure as below
     uGF.currentControlLoop = OFF;
     uGF.controlMode = FIXED_VOLTAGE;

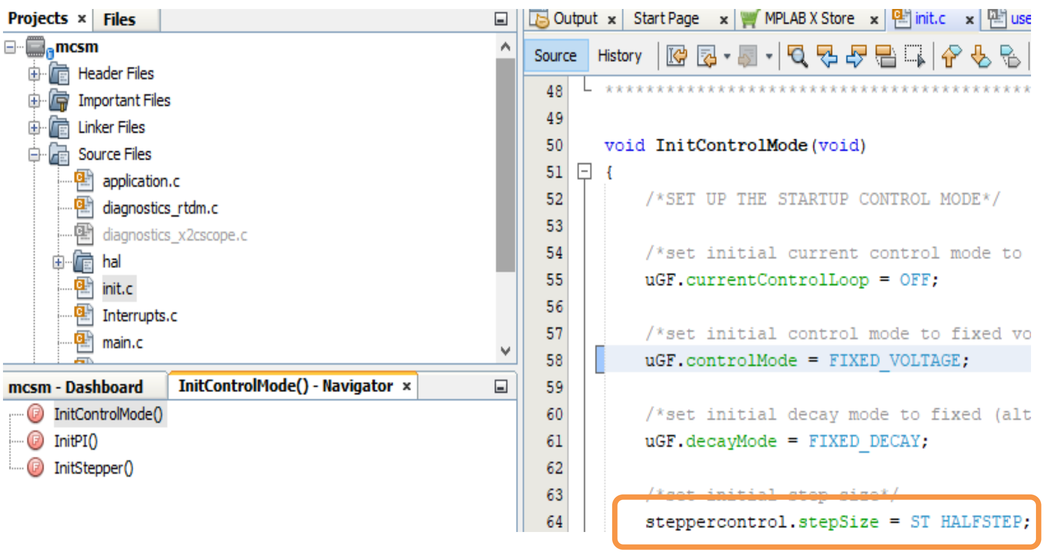
   - For Open loop Current Control, Configure as below
     uGF.currentControlLoop = OFF;
     uGF.controlMode = FIXED_CURRENT;

   - For Closed loop Current Control, Configure as below
     uGF.currentControlLoop = ON;
     uGF.controlMode = FIXED_CURRENT;

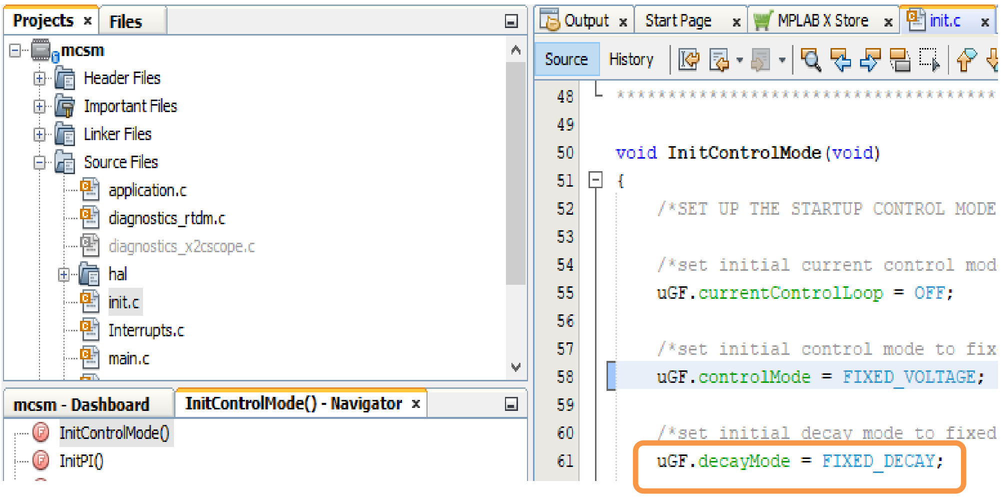
   In this Demo, Control mode is configured as Open loop voltage Control.



5. Open `int.c` (under `mcsm -> Sourcefiles->init`) in the project `mcsm.X` and select the required step size in **InitControlMode()** function.
   In this Demo, step size is configured as HalfStep.

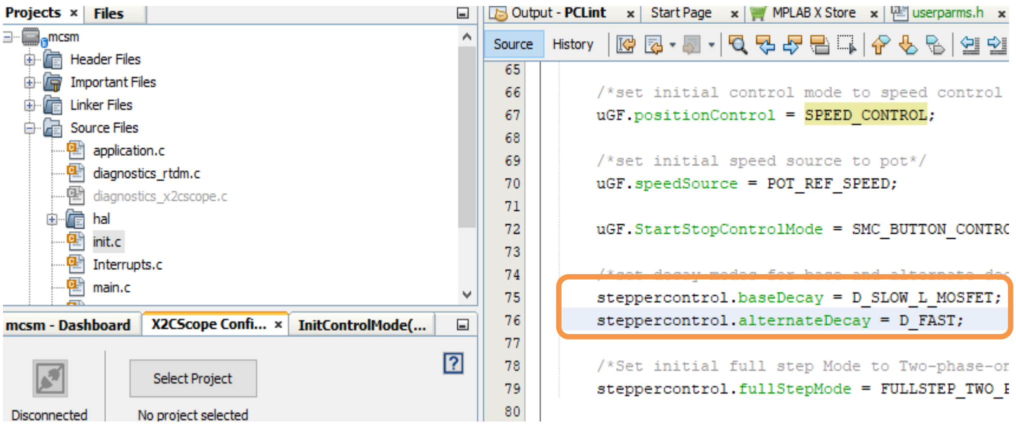


6. Open `int.c` (under `mcsm -> Sourcefiles->init`) in the project `mcsm.X` and select the decay mode (fixed decay or alternate decay) in **InitControlMode()** function.
   In this Demo, decay mode is configured as fixed decay mode.



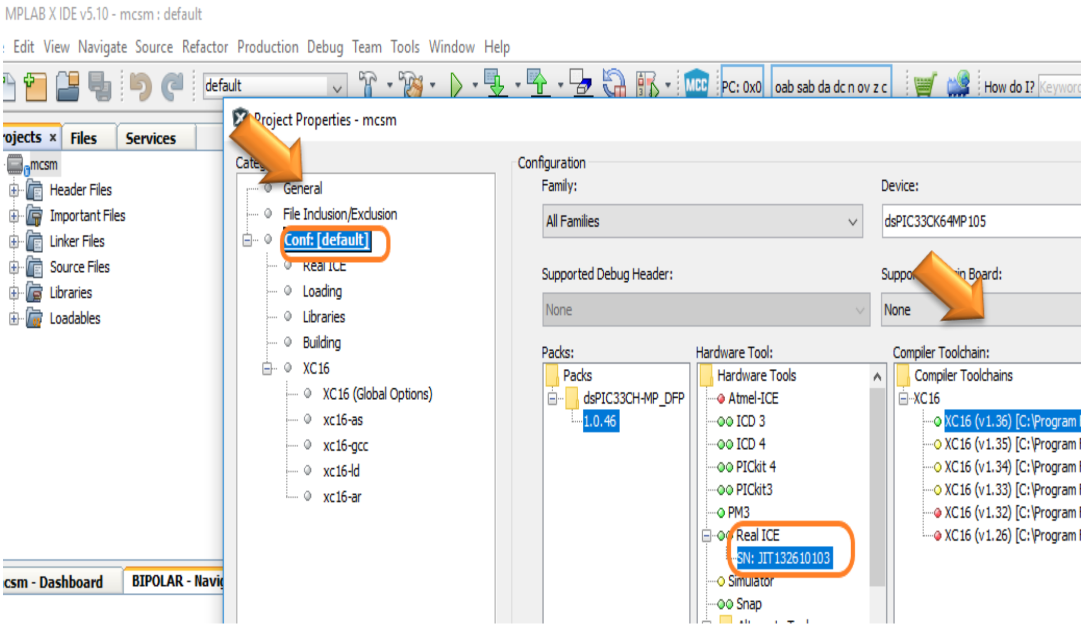
7. Similarly select the base decay and alternate decay type, if decay mode is fixed decay mode then only base decay will be used, else if decay mode is alternate decay mode then both base and alternate decay types will be used.

   In this demo, base decay is selected as slow decay lower mosfet recirculation and alternate decay is selected as fast decay.

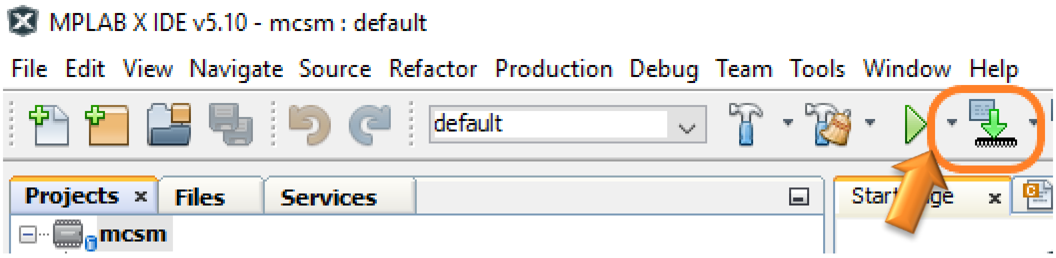


8. Right click on the project *mcsm.X* and select "Properties" to open its Project Properties Dialog. Click the 'Conf: [default]' category to reveal the general project configuration information.

   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 dsPIC33CK64MP105.In this case "XC16(v1.36)" is selected.
   - Select the Hardware Tool to be used for programming and debugging. In this case, "Real ICE" is the selected programmer.
   - After selecting Hardware Tool and Compiler Toolchain, click button **Apply**



9. To build the project (in this case *mcsm.X*) and program the device dsPIC33CK64MP105, click “**Make and Program Device Main project**” on the toolbar.



10. Run or Stop the motor by pressing the push button **S1**. The motor will run on the following modes as the button is pressed. The sequence is completed once the button is pressed nine times, and then it will return to the initial state (1/2 Step).

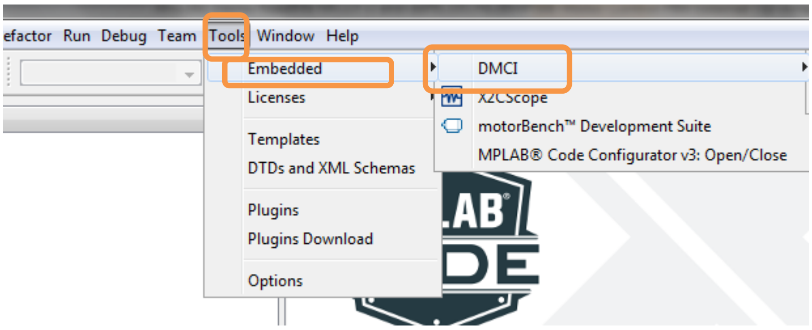
	Button Pressed N times	Operating Mode
	1	1/2 Step
	2	1/4 Step
	3	1/8 Step
	4	1/16 Step
	5	1/32 Step
	6	1/64 Step
	7	Stop
	8	Full Step (2-Phase ON)
	9	Full Step (Wave Driver)

11. If desired, the motor speed can be varied using the potentiometer (labeled “POT1”).

5.3. Data visualization through DMCI Plug-in of MPLABX

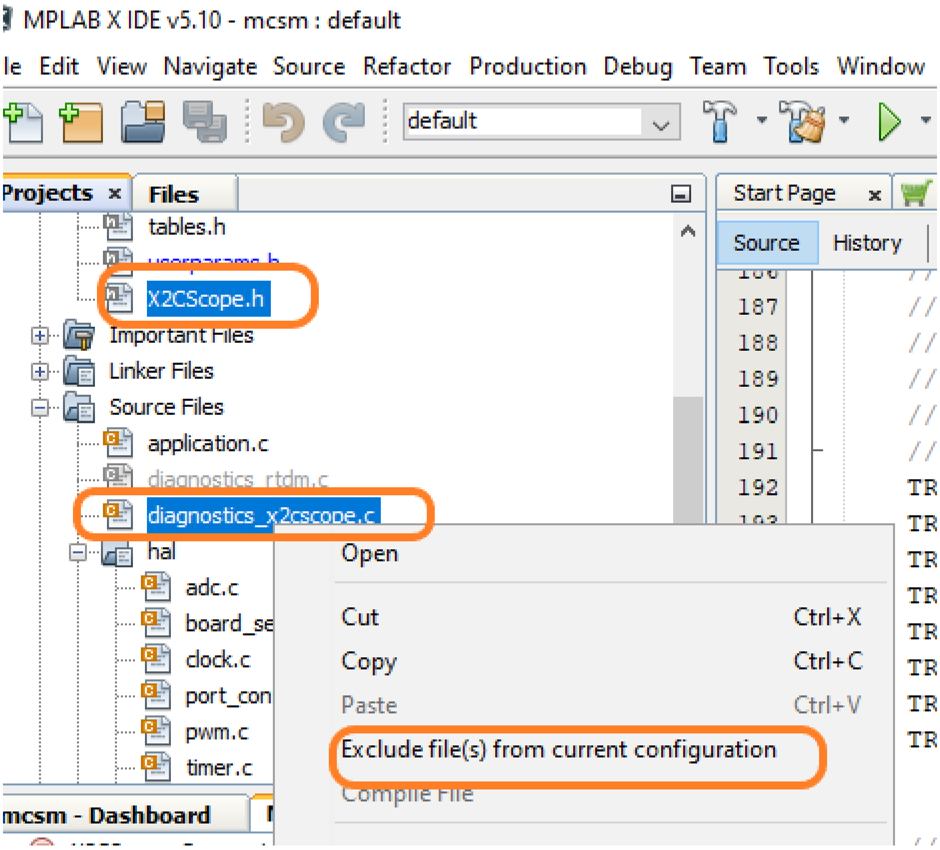
The project firmware comes with the software library for Real Time Data Monitoring (RTDM), needed to interface with DMCI Plug-in available in the MPLAB X IDE. RTDM, along with DMCI creates a communication link between a host PC and a target device for debugging applications in real-time. For additional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

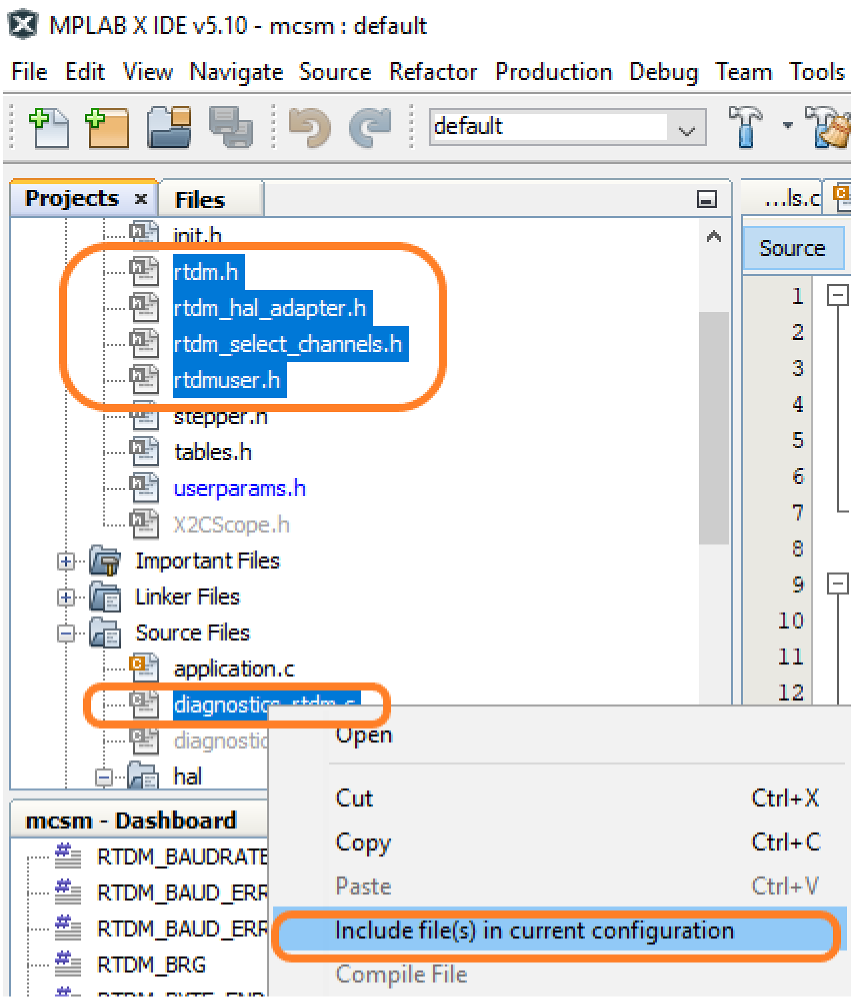


2. To utilize RTDM communication for this demonstration, a mini-USB connection is required between Host PC and dsPICDEM™ MCSM Development Board. Connect a mini-USB cable from your computer to the J4 connector of the dsPICDEM™ MCSM Development Board.

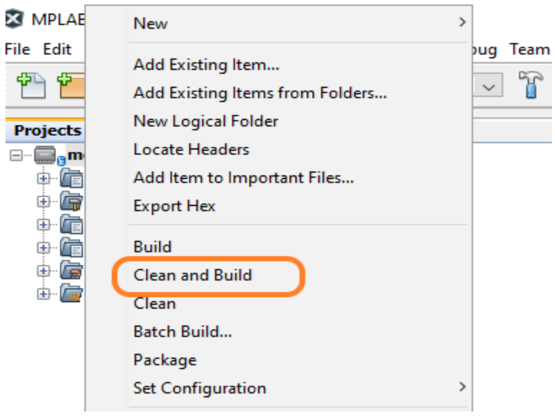
3. Ensure application is configured and running as described under Section Basic Demonstration by following steps 1 through 5.
4. Select files “diagnostics_x2cscope.c” and “**X2CScope.h**”, then right click and set **Exclude file(s) from current configuration**.



5. Select files **rtdm.h, rtdm_hal_adapter.h**, **rtdm_select_channels.h, rtdmuser.h and diagnostics_rtdm.c**, then right-click 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.

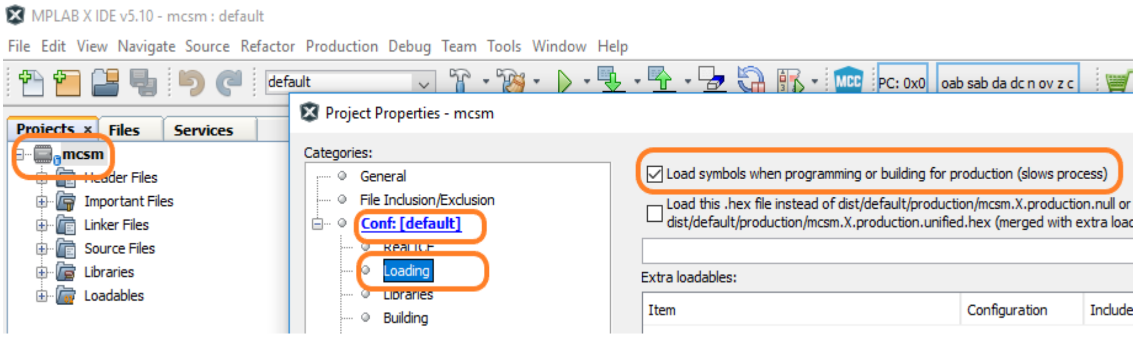


6. Build the project `mcsm.X`. To do that right click on the project `mcsm.X` and select “Clean and Build”.

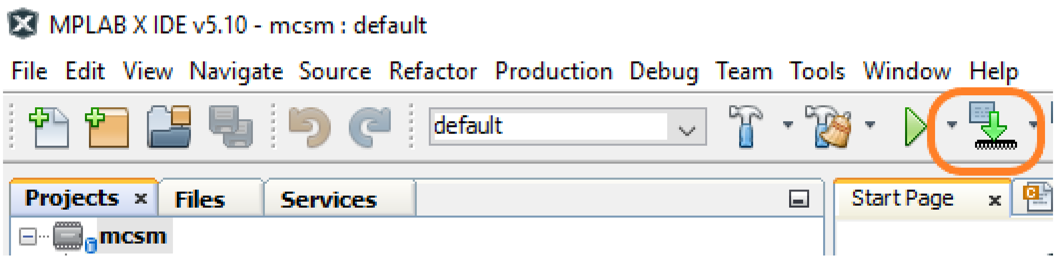


Note:

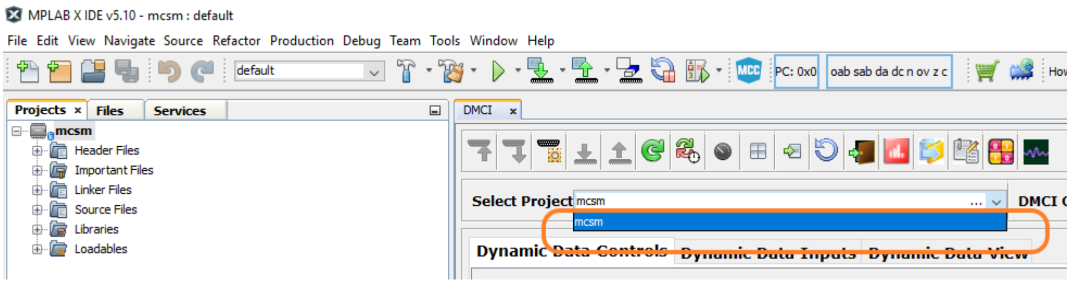
When using RTDM with DMCI you will need to check the checkbox “Load symbols when programming or building for production (slows process)” under the “Loading” category of the Project Property dialog.



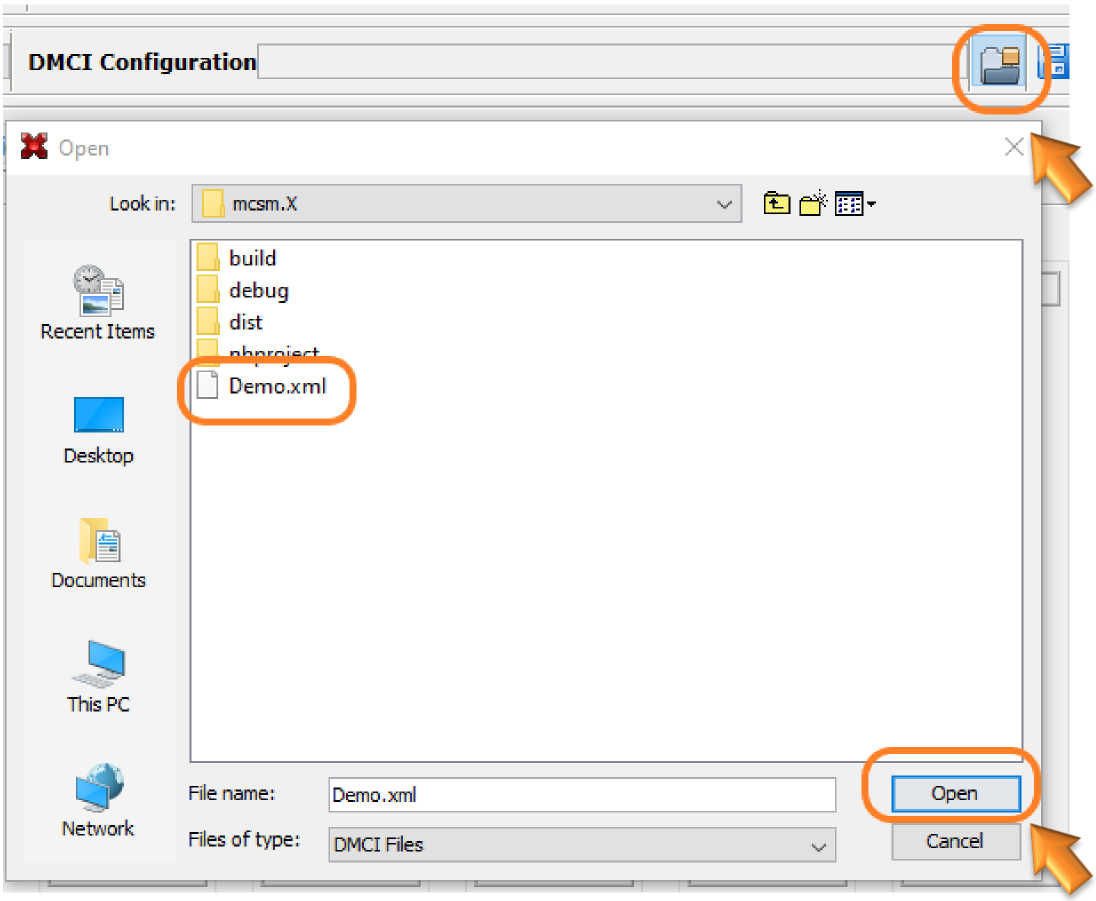
7. To program the device dsPIC33CK64MP105, click "**Make and Program Device Main project**" on the toolbar.



8. Open the DMCI window by selecting `Tools>Embedded>DMCI>DMCI Window`. From the Select Project drop down menu availble in the DMCI window , select project 'mcsm'



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

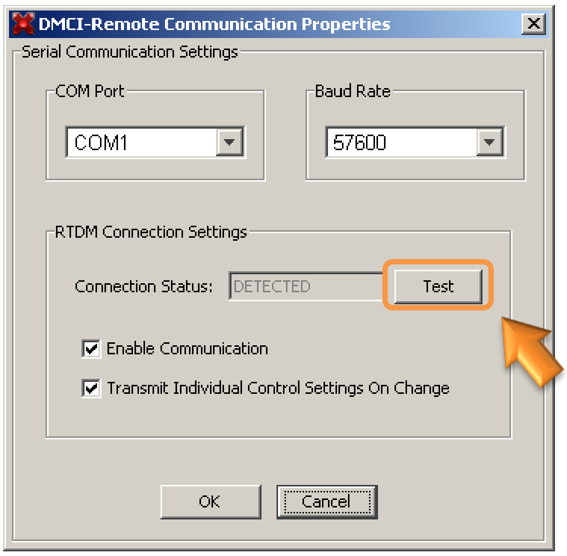


10. The DMCI window appears as follows:

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

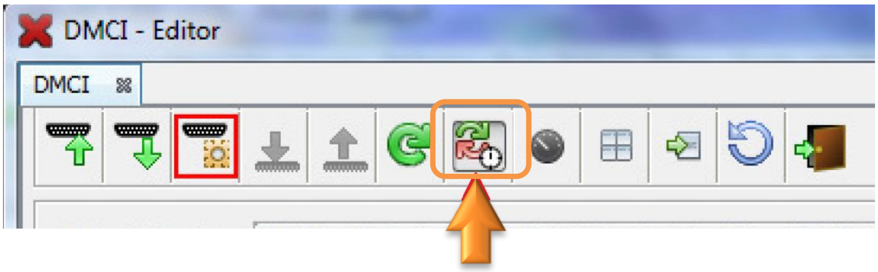
11. Click **Serial Settings** to connect RTDM with your computer.

12. 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. Once Baud Rate and COM Port are selected, then click on Test Button, then Conection Status turn into "DETECTED", if the link is established as programmed**.**

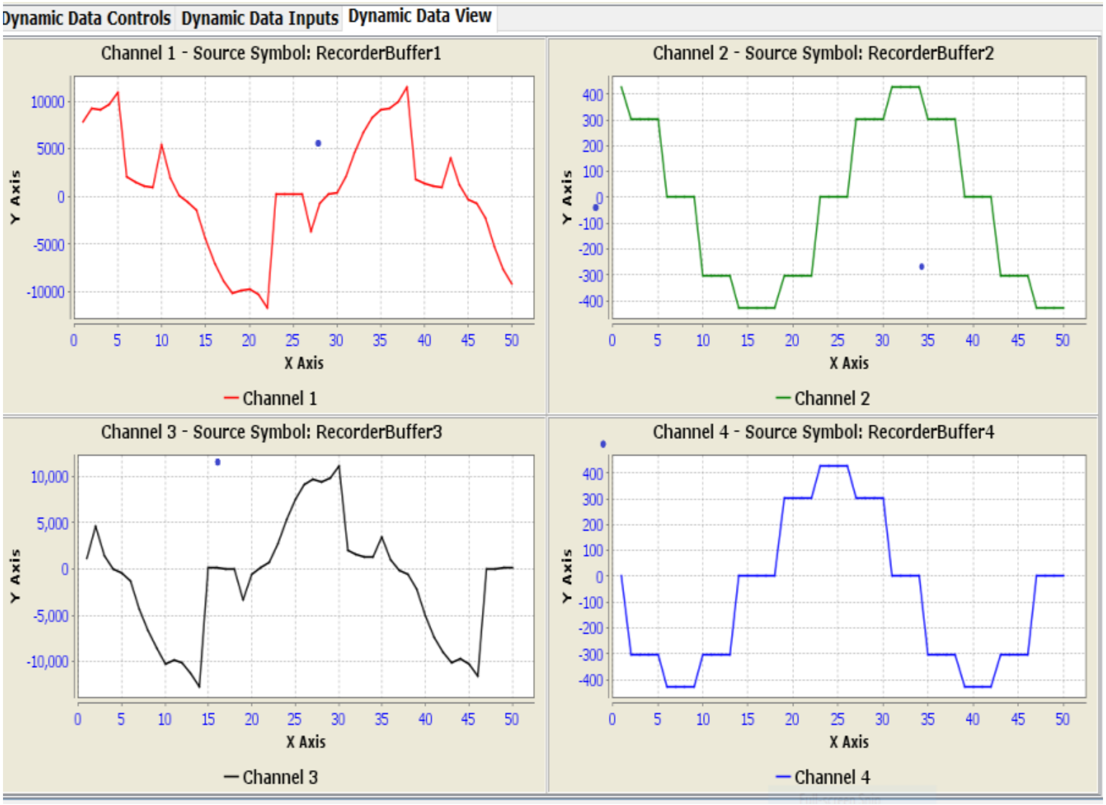


13. Once communication is detected, make sure the "Enable Communication" box is checked and click **OK**.

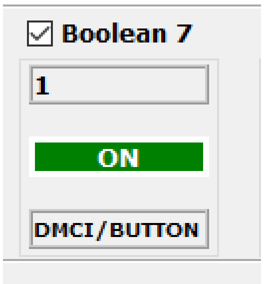
14. To plot variables in real time, enable "Automated Event Control" by clicking **Automatic Event Execution** icon found on the toolbar.



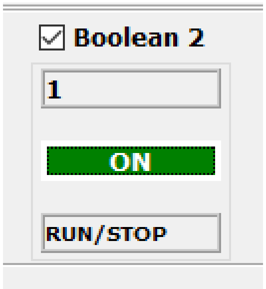
15. The DMCI window shows variables plotted in real time, which is updated automatically in Dynamic Data View Tab. The motor is running now in HalfStepMode with fixed voltage reference.



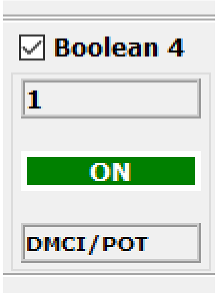
16. Press the "DMCI/BUTTON" to **ON** Position to accept START/STOP and StepChange Commands from DMCI.



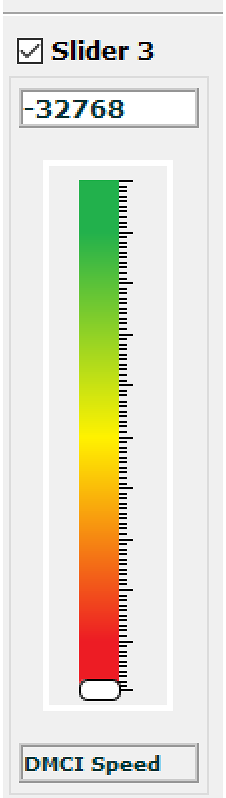
17. Press the "RUN/STOP" to **ON** from DMCI to start the motor at initial Speed.



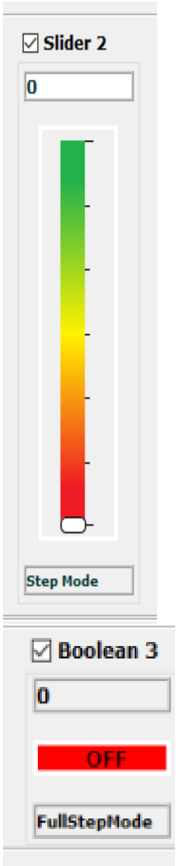
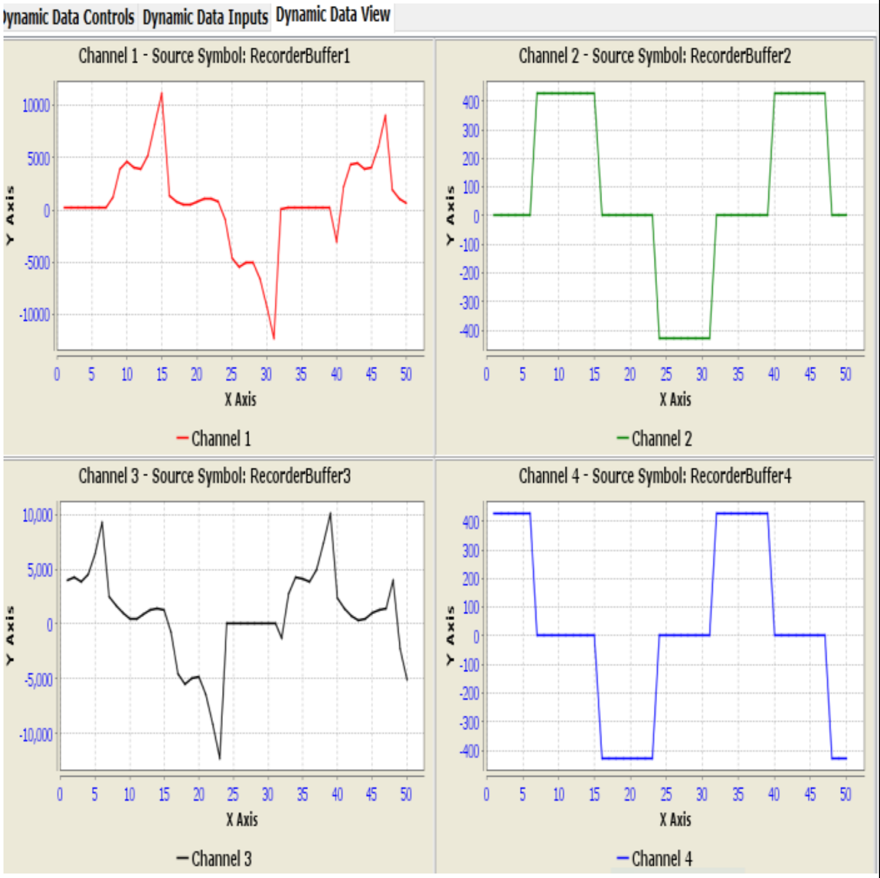
18. In order to change speed with slider, turn the “DMCI/POT” to the **ON** Position.



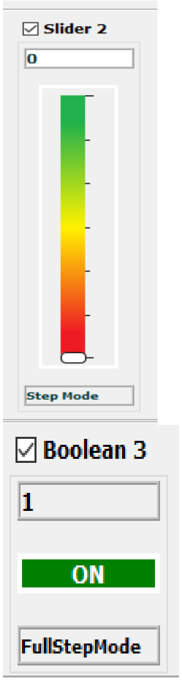
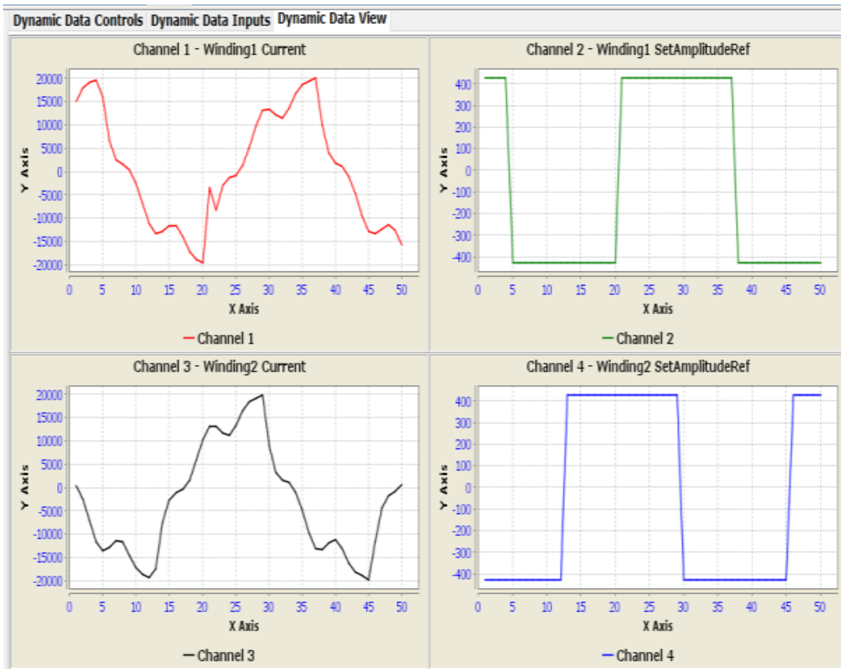
19. Vary the speed of the motor by setting the value of “DMCI Speed” Slider. +32768 corresponding to 100% in positive direction and -32768 Corresponding to 100% in reverse direction.



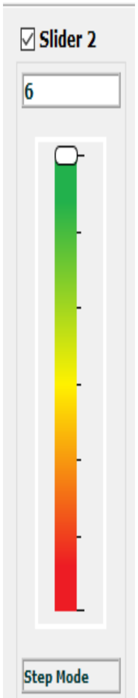
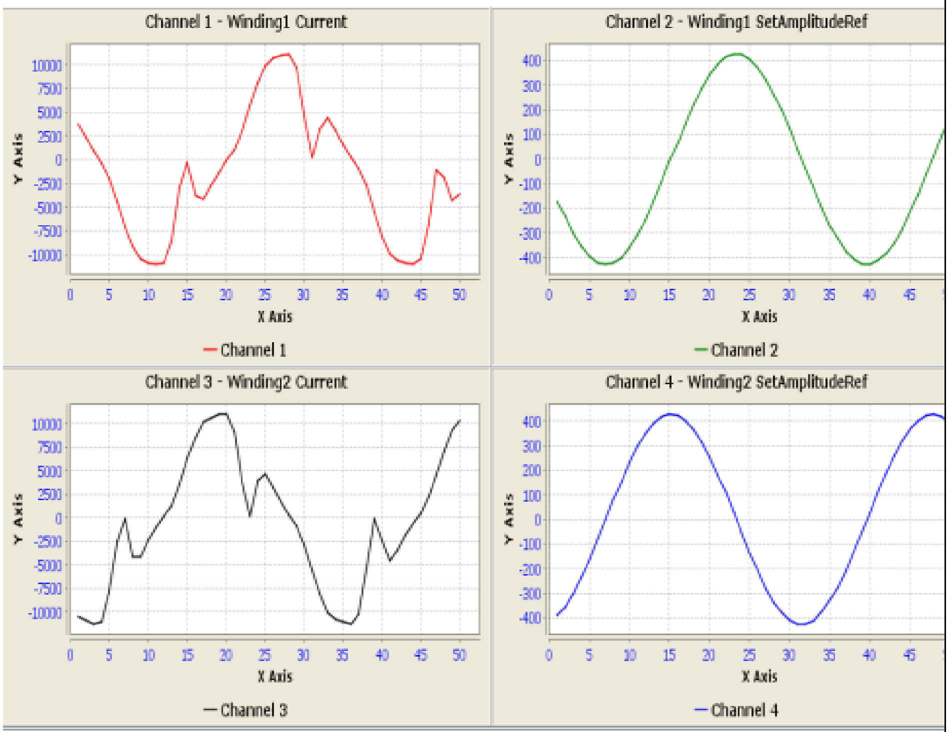
20. To run motor in Wave Mode with fixed voltage reference, the “Step Mode” slider has to be set to zero, indicating that a full step control is done, and also set the “FullStepMode” to the **OFF** Position. All of these controls are available in Dynamic Data Controls Tab.

Slider and Boolean	Resultant Graphs
	

21. To run motor in Full Step (2-Phase ON) Mode with fixed voltage reference, the “Step Mode” slider has to be set to zero, indicating that a full step control is done, and also set the “FullStepMode” to the **ON** Position.

Slider and Boolean	Resultant Graphs
	

22. To run the motor in 1/64 micro-stepping mode with the fixed voltage reference, Set the value of “Step Mode” slider to 6.

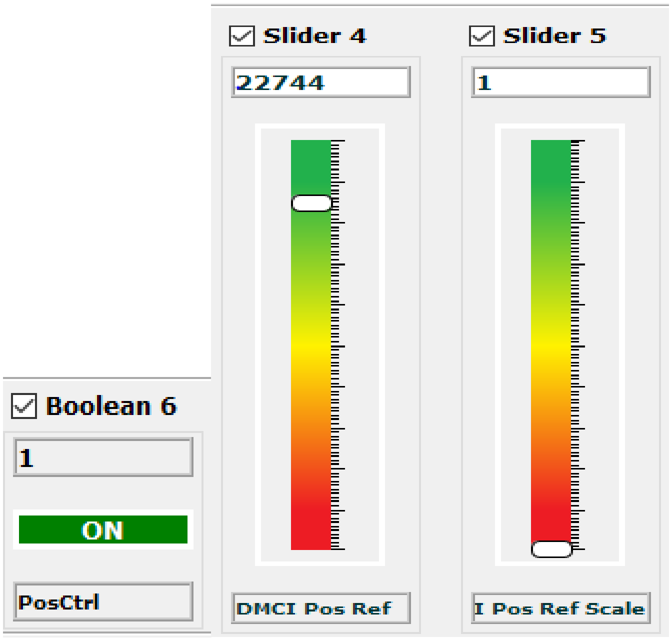
Slider	Resultant Graphs
	

The following table shows the relationship between the “Step Mode” slider values and the operating modes.

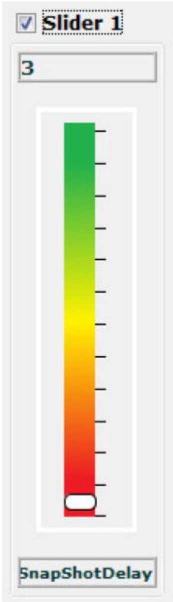
“Step Mode” Slider value	Mode
0	Full Step
1	1/2 Step
2	1/4 Step
3	1/8 Step
4	1/16 Step
5	1/32 Step
6	1/64 Step

23. To change to Alternative Decay modes, which uses both slow and fast decay modes, set the “AltDecay” Boolean to **ON** Position.

24. To enable position control mode, set the “PosCtrl” Boolean to **ON** Position, the “DMCI Pos Ref” and “DMCI Pos Ref Scale” slides decides the number of degrees rotor will rotate before stops.



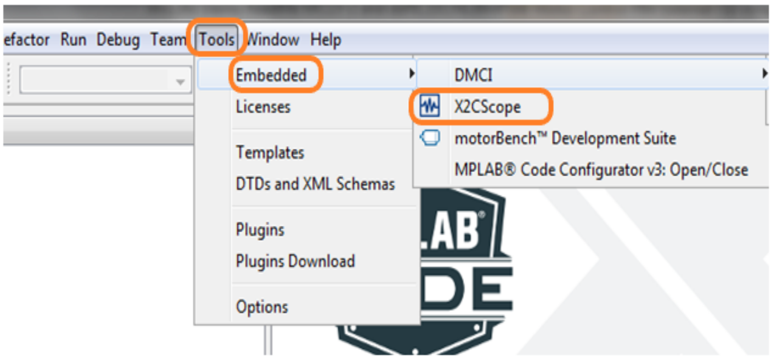
25. To change the time window to see data over larger time span, change the value of the `'SnapShotDelay'`, which controls how the buffers are being filled.



5.4. Data visualization through X2CScope Plug-in of MPLABX

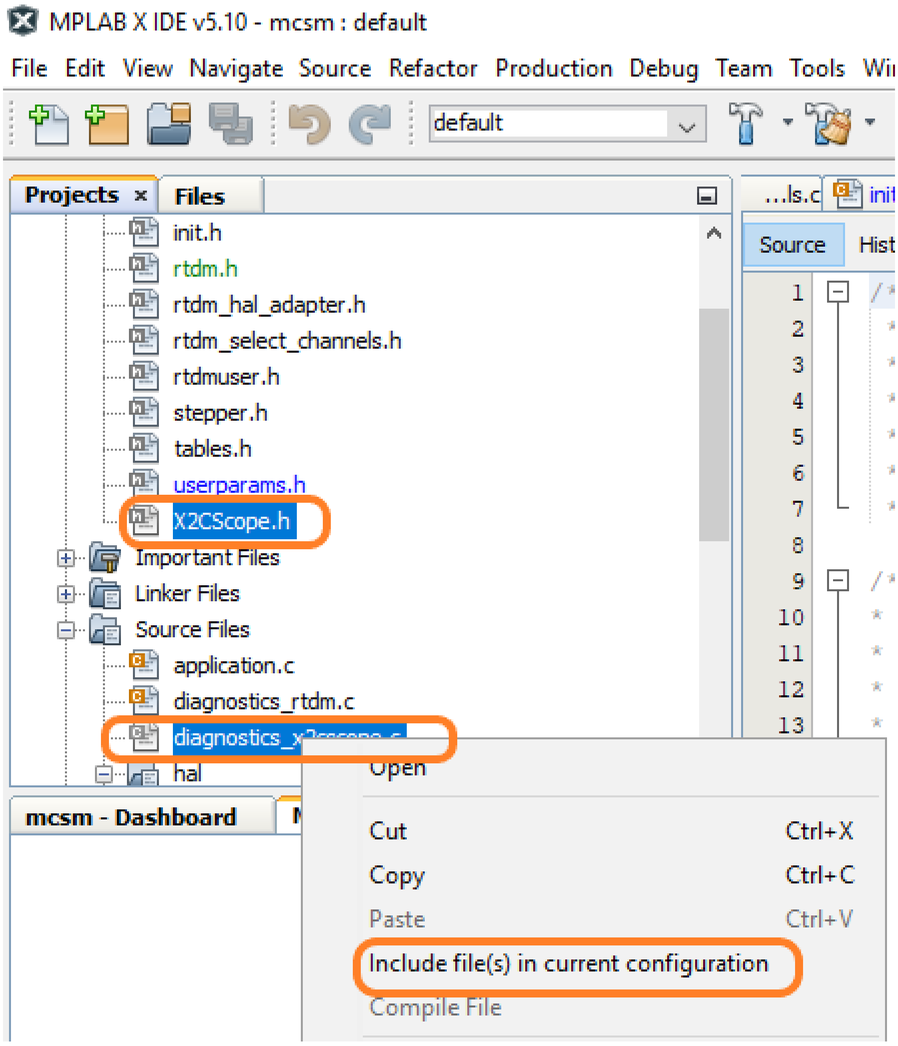
The application firmware comes with initialization required to interface Controller with Host PC to enable Data visualization through X2C Scope plug-in.X2C-Scope is a third-party plugin for MPLAB X which facilitates real-time diagnostics.

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

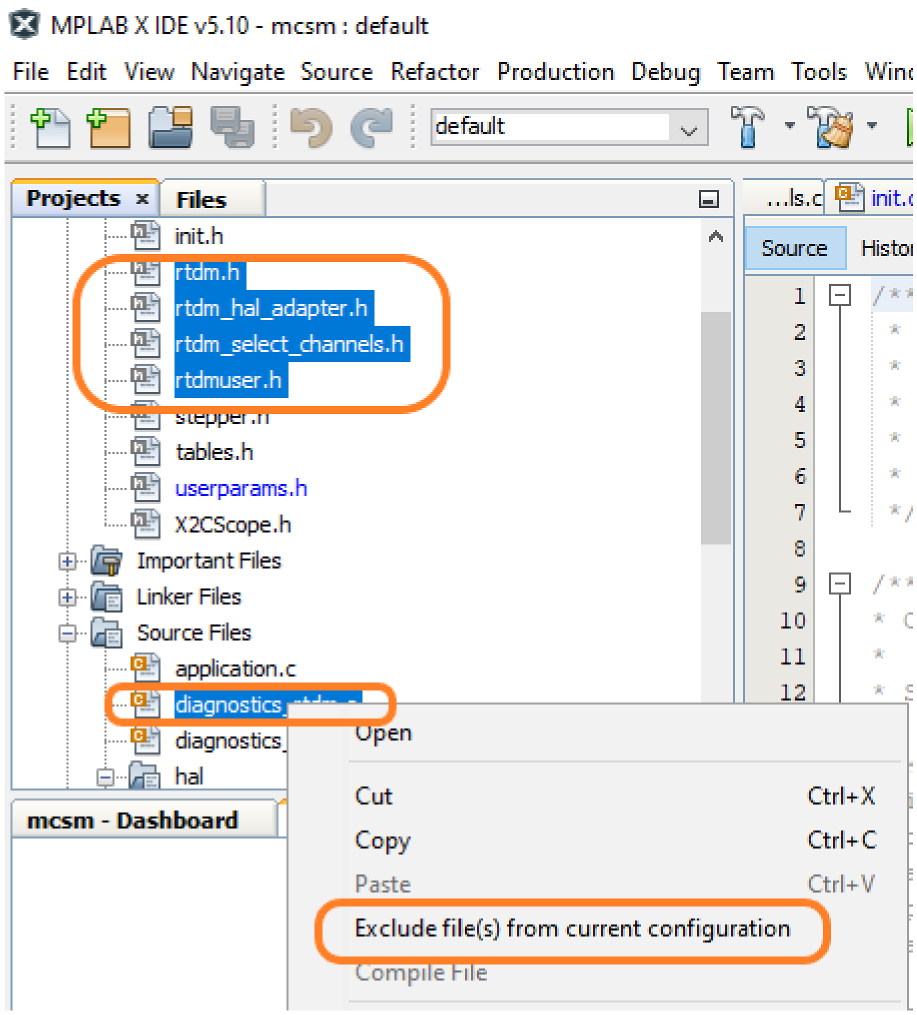


2. To utilize X2C communication for this demonstration, a mini-USB connection is required between Host PC and dsPICDEM™ MCSM Development Board. Connect a mini-USB cable from your computer to the J4 connector of the dsPICDEM™ MCSM Development Board.

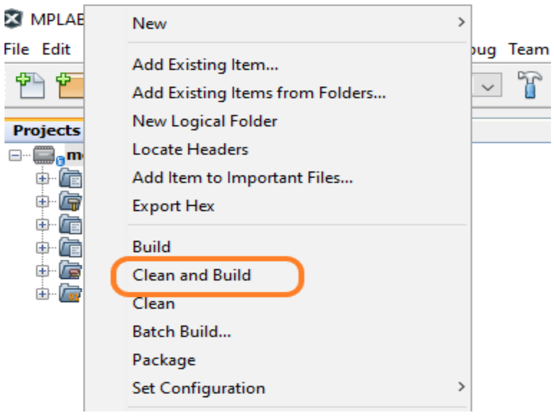
3. Ensure application is configured and running as described under Section Basic Demonstration by following steps 1 through 5.
4. Select files “diagnostics_x2cscope.c” and “**X2CScope.h**”, then right click and set **include file(s) from current configuration**.



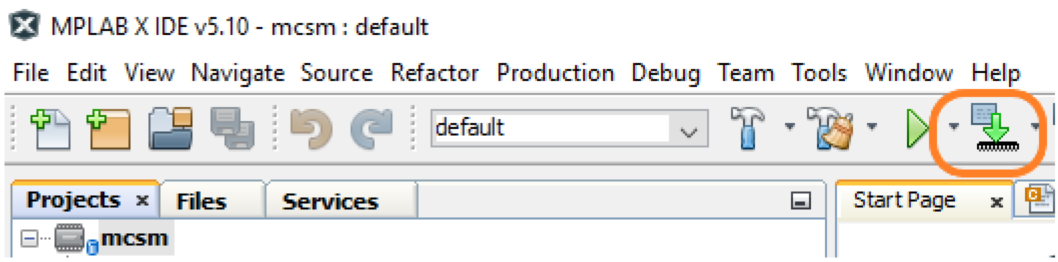
5. Select files **rtdm.h, rtdm_hal_adapter.h**, **rtdm_select_channels.h, rtdmuser.h and diagnostics_rtdm.c**, then right-click and set **Exclude file(s) from current configuration** to remove these files as part of current Project Configuration.



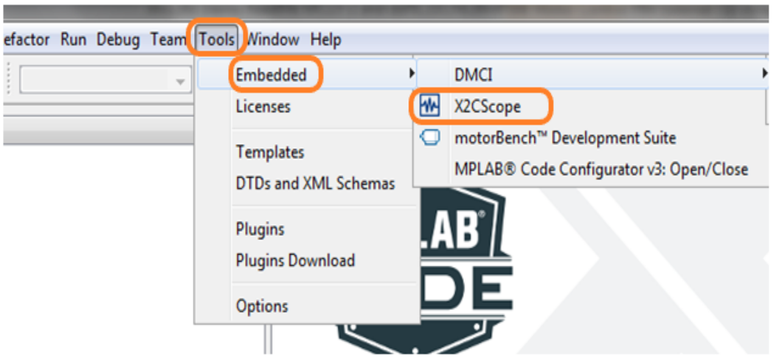
6. Build the project `mcsm.X`. To do that right click on the project `mcsm.X` and select “Clean and Build”.



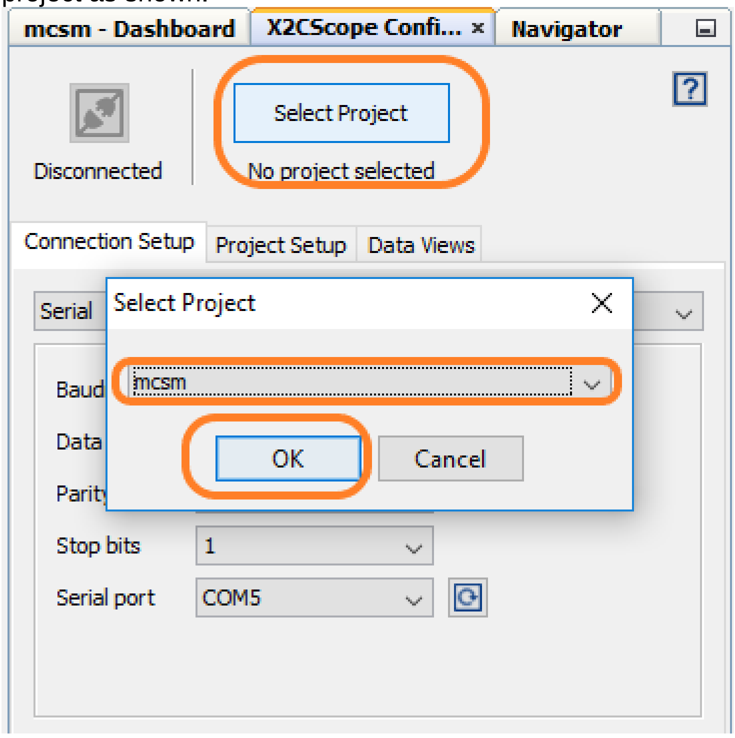
7. To program the device dsPIC33CK64MP105, click “**Make and Program Device Main project**” on the toolbar.



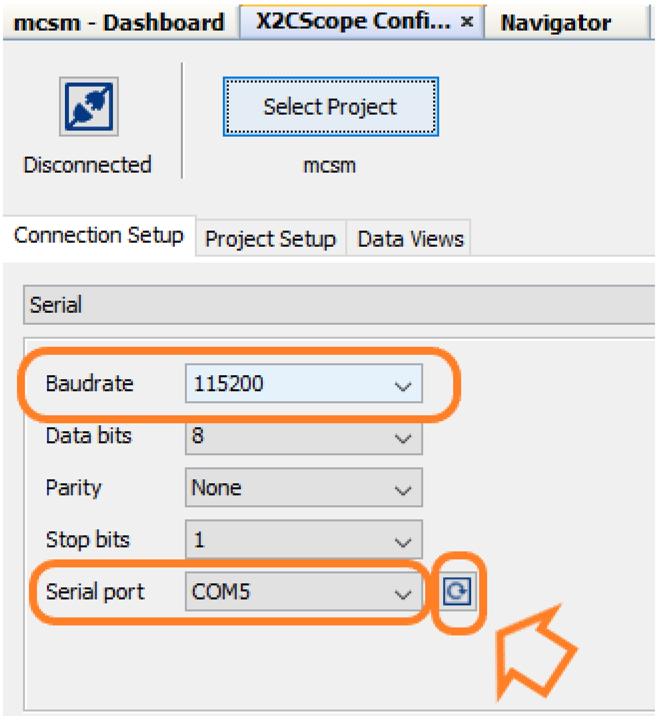
8. Open the X2C Scope window by selecting Tools>Embedded>X2CScope Window.



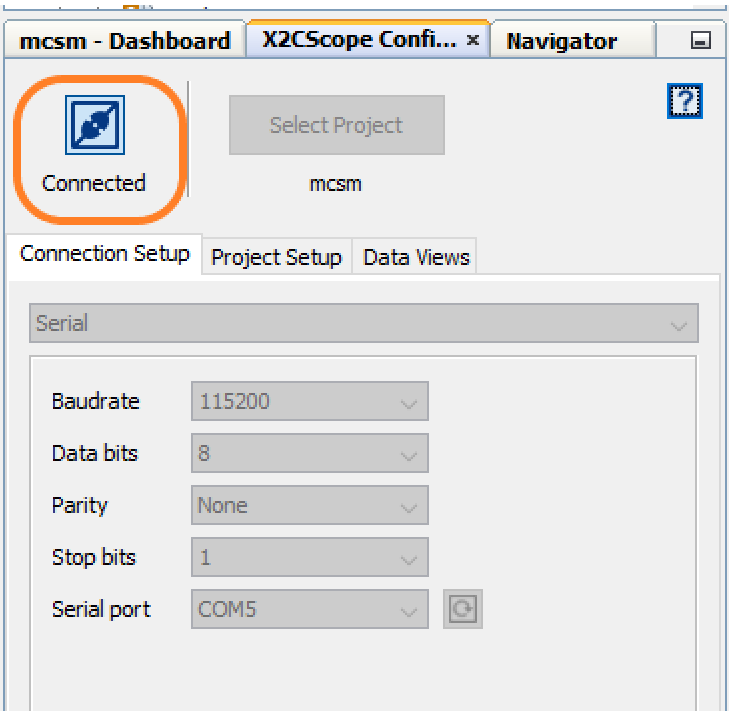
9. Open the X2CScope Configuration window and in “Select project” menu, select mcsm.X project as shown.



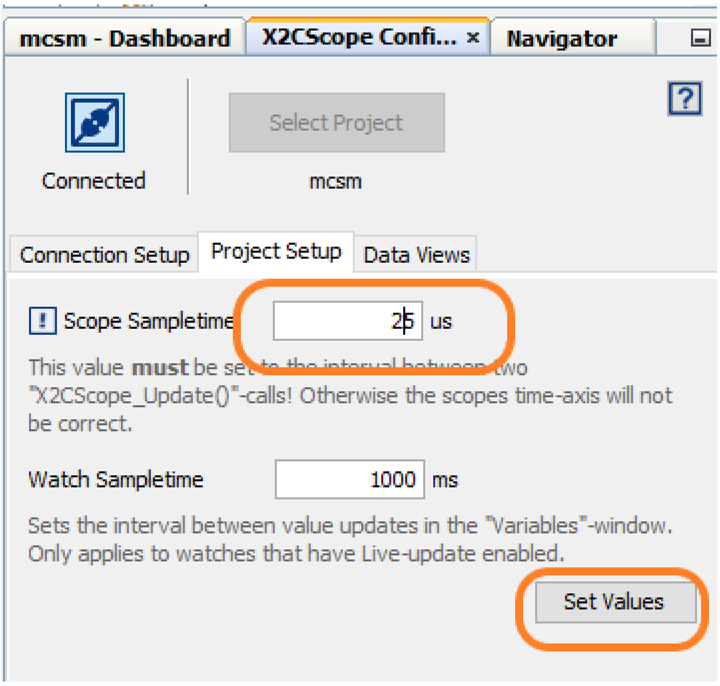
10. Remote Communication needs to be established, as indicated in the following figure (the communication baud rate should set to 115200, while COM port used depends on your settings. Click on Refresh button to detect COM Port automatically.



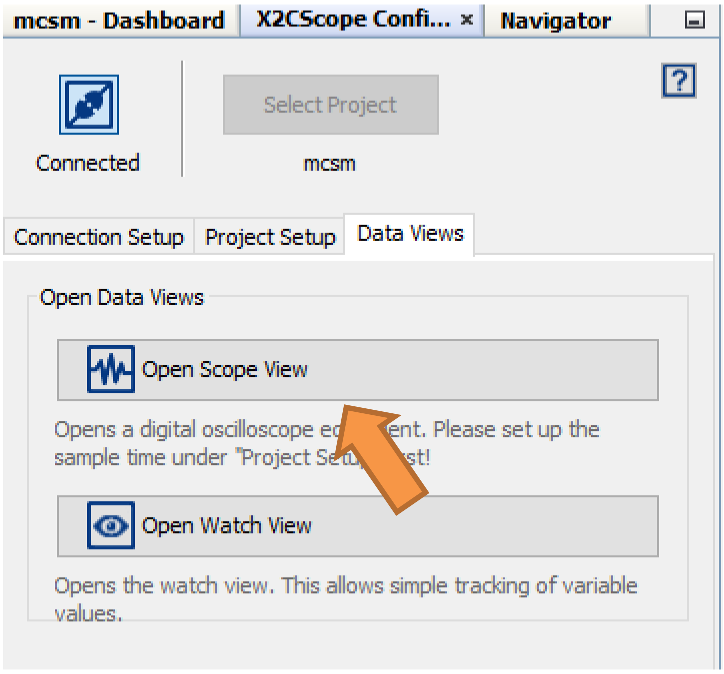
11. Once COM port detected, click on "**Disconnected**", and it will be turn into "**Connected**", if the link is established as programmed.



12. 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 40kHz (25Us).

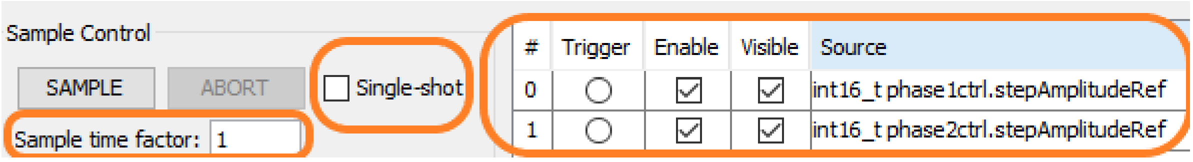


13. When the setup is established, click on open scope View (under sub window "Data Views"), this open Scope Window.

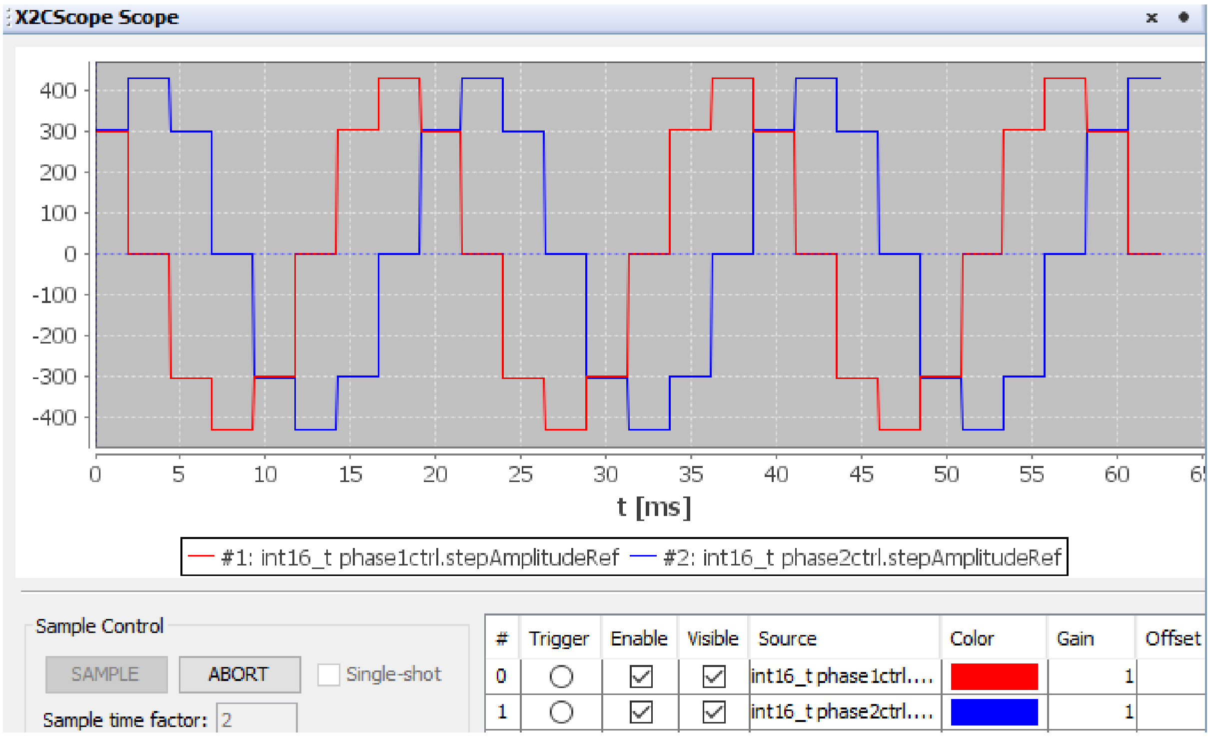


14. In the window select the variables user may want to watch. Click on the source, a window Select Variables opens upon the screen. From the select variables list, choose the variable that you want to view. Then ensure Enable, Visible check boxes are checked as shown.

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



15. X2C scope window shows variables in real time, which is updated automatically.



6. dsPIC® DSC RESOURCE USAGE SUMMARY

6.1. Device Pin Mapping and Its Functionality in the Firmware:

The following table summarizes device pins configured and used in the AN1307 motor control application firmware demonstrated using the Development Board and the dsPIC33CK64MP105 External Op Amp Motor Control PIM(MA330050-1). Refer "*dsPIC33CK64MP105 External Op Amp Motor Control Plug-in-Module (PIM) Information Sheet (DS50002848)"* 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**/**RB14**	PWM Output	Controls Hex Bridge MOSFET Q1:A
PWM1L	PIM:93	2	RP47/**PWM1L** /**RB15**	PWM Output	Controls Hex Bridge MOSFET Q1:B
PWM2H	PIM:99	47	TDI/RP44/**PWM2H/RB12**	PWM Output	Controls Hex Bridge MOSFET Q2:B
PWM2L	PIM:98	48	RP45**/PWM2L/RB13**	PWM Output	Controls Hex Bridge MOSFET Q2:A
PWM3H	PIM:03	45	TMS/RP42/**PWM3H/RB10**	PWM Output	Controls Hex Bridge MOSFET Q3:A
PWM3L	PIM:100	46	TCK/RP43/**PWM3L/RB11**	PWM Output	Controls Hex Bridge MOSFET Q3:B
PWM4H	PIM:18	30	RP72/SDO2/**PCI19/RD8**	PWM Output	Controls Hex Bridge MOSFET Q4:B
PWM4L	PIM:19	17	AN17/ANN1/**IBAS1/RP54/RC6**	PWM Output	Controls Hex Bridge MOSFET Q4:B Solder R87, remove R86 from the board
FAULT_MC	PIM:68	29	RP57/ASCL1/**SDI2/RC9**	PWM Input	Connected to Over Current Fault Output
Analog Inputs – Bus Current, Phase Currents, Speed Reference					
POT	PIM:20	11	DA-COUT/AN3/CMP1C/RA3	Analog Input	Speed Reference Connected to Potentiometer POT1
IMOTOR1	PIM:25	8	OA1OUT/AN0/CMP1A/IBIAS0/RA0	Analog Input	Connected to Motor Phase Current 1
IMOTOR2	PIM:24	9	OA1IN-/ANA1/RA1	Analog Input	Connected to Motor Phase Current 2
DCBUS VOLTAGE	PIM: 23	10	OA1IN+/AN9/RA2	Analog Input	Connected to Bus Voltage
Miscellaneous Signals					
BTN_1	PIM:70	24	AN16/ISRC2/RP55/RC7	Digital Input	Connected to Push Button S1
RX (UART)	PIM:49	28	**RP56**/ASDA1/SCK2/**RC8**	UART1 Input	Connected to UART-USB converter to establish serial communication between Host PC and the dsPIC® DSC as needed by DMCI-RTDM or X2C.

TX (UART)	PIM:50	44	**RP65/**PWM4H**/RD1**	UART1 Output	Connected to UART-USB converter to establish serial communication between Host PC and the dsPIC® DSC as needed by DMCI-RTDM or X2C.
MCLR	PIM:13	5	MCLR	Device Reset	Master Clear (Reset) input. This pin is an active-low Reset to the device.
PGC	PIM:26	34	PGC3/RP38/SCL2/RB6	Program-ing Clock	Clock input pin for Program-ming/Debugging Communication Channel 3.
PGD	PIM:27	33	PGD3/RP37/SDA2/RB5	Program-ming Data	Data I/O pin for Program-ming/Debugging Communication Channel 3.
OSCI	PIM:63	21	OS-CI/CLKI/AN5/RP32/RB0	Oscillator Input	Oscillator crystal input.
OSCO	PIM:64	22	OS-CO/CLKO/AN6/RP33/RB1	Oscillator Output	Oscillator crystal output.

7. REFERENCES:

For additional information, refer following documents or links.

1. AN1307 Application Note “*Stepper Motor Control with dsPIC® DSCs*”
2. dsPICDEM™ MCSM Development Board User’s Guide (DS70610)
3. dsPIC33CK64MP105 External Op Amp Motor Control Plug-in-Module (PIM) Information Sheet (DS50002848)
4. dsPIC33CK64MP105 Family datasheet (DS70005363).
5. Family Reference manuals (FRM) of dsPIC33CK64MP105 family
6. MPLAB® X IDE User’s Guide (DS50002027) or MPLAB® X IDE help
7. Real-Time Data Monitor User’s Guide (DS70567) or Real Time Data Monitoring Tool - RTDM
8. Data Monitor and Control Interface - Developer Help
9. MPLAB® X IDE installation
10. MPLAB® XC16 Compiler installation