AN957 Demonstration ReadMe for the dsPICDEM™ MCLV-2 Development Board with the dsPIC33CK256MP508 External Op-Amp Motor Control PIM (MPLAB® X IDE)

1. INTRODUCTION

This document describes the setup requirements for running the Sensored BLDC Motor Control Algorithm, which is referenced in AN957 "Sensored BLDC Motor Control" using a dsPICDEM™ MCLV-2 Development Board in the External Op Amp configuration.

The demonstration is configured to run on the dsPICDEM™ MCLV-2 Development Board (DM330021-2) in External Op-amp configuration with the dsPIC33CK256MP508 External Op-Amp Motor Control Plug-In Module (PIM) (MA330041-1).

2. SUGGESTED DEMONSTRATION REQUIREMENTS

2.1. Motor Control Application Firmware Required for the Demonstration

- `AN957_dsPIC33CK256MP508_EXT_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 v5.40
- MPLAB® XC16 Compiler v1.50
- MPLAB® X IDE Plugin: Latest version of X2C-Scope Plugin

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

- dsPICDEM™ MCLV-2 Development Board (DM330021-2)
- 24V Power Supply (AC002013)
- 24V 3-Phase Brushless DC Motor (AC300020)
- dsPIC33CK256MP508 External Op-Amp Motor Control Plug-in module (MA330041-1)
- Microchip Programmer tools like MPLAB® REAL ICE™ In-Circuit Emulator (DV244005) or MPLAB® ICD 3 (DV164035) etc.

Note:

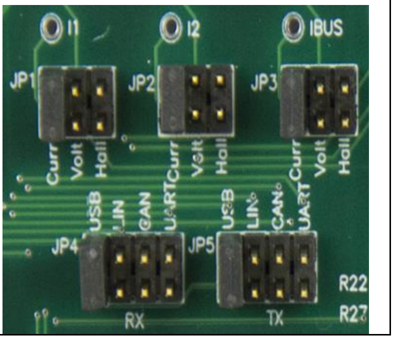
All items listed under the section 2.3. Hardware Tools Required for the Demonstration are available at microchip DIRECT.

3. HARDWARE SETUP

This section describes hardware setup required for the demonstration. Bus current feedback needed by the firmware is amplified by the operational amplifier provided on the Development Board. This is referred as ‘*external amplifier configuration*’.

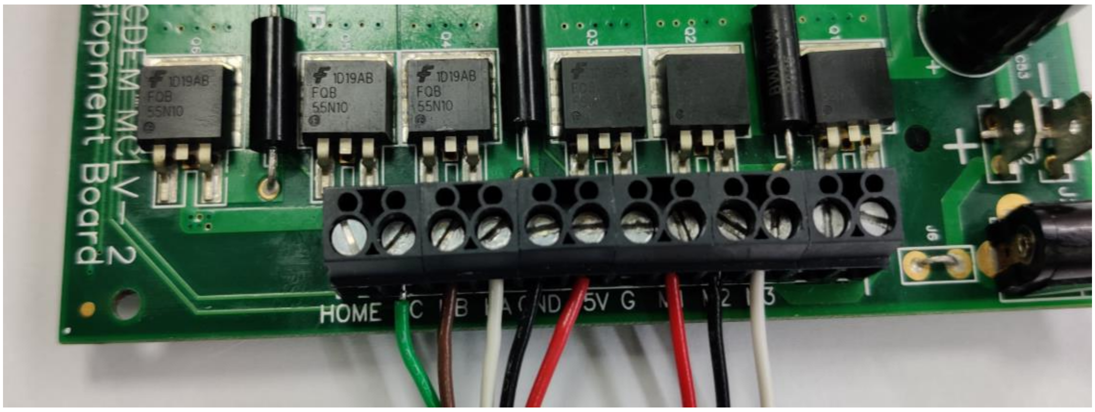
Refer *dsPICDEM™ MCLV-2 Development Board User’s Guide* for any clarification while setting up the hardware

1. Disconnect power to the dsPICDEM™ MCLV-2 Development Board and set up the following jumpers:

Jumper	Pins to Short	Board Reference
JP1	Don’t care	 I1 I2 IBUS JP1 JP2 JP3 Curr Volt Hall USB LIN CAN UART JP4 JP5 RX TX R22 R27
JP2	Don’t care	
JP3	1-2(**Curr**-Current)	
JP4	USB position	
JP5	USB position	

2. Connect the three phase wires from the motor to M1, M2, and M3 terminals of connector J7, provided on the Development Board as mentioned in the below table.

MCLV2 Board	Hurst075 Motor	
	Winding Terminals (Color as per image below)	Molex 39-01-2040 (Mating Connector)
M1	Red	1
M2	Black	2
M3	White	3



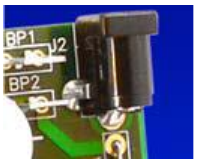
3. Connect the hall sensors from the motor to HA, HB and HC terminals of connector J7, provided on the Development Board as mentioned in the below table.

MCLV2 Board	Hurst075 Motor	
	Hall Terminals (Color as per image above)	Molex 50-57-9408 (Mating Connector)
5V	Red	1
GND	Black	2
HA	White	4
HB	Brown	3
HC	Green	5

4. Connect the ‘External Op Amp Configuration Matrix board’ to matrix board header J14. Ensure the matrix board is correctly oriented before proceeding.

5. Insert the dsPIC33CK256MP508 External Op-Amp Motor Control PIM into the PIM Socket U9 provided on the dsPICDEM™ MCLV-2 Development Board. Make sure the PIM is correctly placed and oriented before proceeding.

6. Plug in the 24V power supply to connector J2 provided on the dsPICDEM™ MCLV-2 Development Board.



7. Connect the Microchip programmer/debugger tools like MPLAB REAL ICE™ or MPLAB ICD-3™ to the Connector J11 of the dsPICDEM™ MCLV-2 Development Board and to the Host PC used for programming the device.

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 X2C plug-in used for testing the firmware are mentioned in the section Motor Control Application Firmware Required for the Demonstration. To get help on

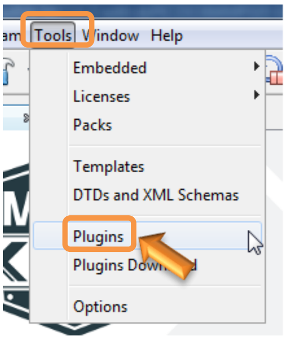
- MPLAB X IDE installation, refer link
- MPLAB XC16 Compiler installation steps, refer link

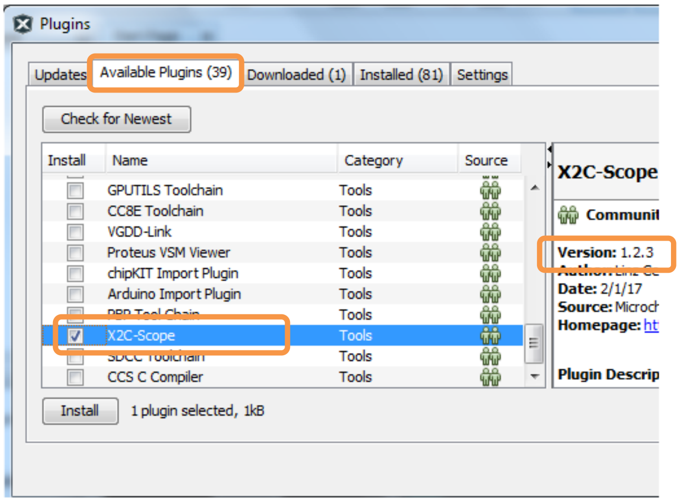
If MPLAB IDE v8 or earlier is already installed on your computer, then run the MPLAB driver switcher (It is installed when MPLAB®X IDE is installed) to switch from MPLAB IDE v8 drivers to MPLAB X IDE drivers. If you have Windows 7 or 8, you must run MPLAB driver switcher in ‘Administrator Mode’. To run 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: 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`.





Install	Name	Category	Source
□	GPUTILS Toolchain	Tools	
□	CC8E Toolchain	Tools	
□	VGDD-Link	Tools	
□	Proteus VSM Viewer	Tools	
□	chipKIT Import Plugin	Tools	
□	Arduino Import Plugin	Tools	
□	PRP Tool Chain	Tools	
☑	X2C-Scope	Tools	
□	SDCC Toolchain	Tools	
□	CCS C Compiler	Tools	

1 plugin selected, 1kB

X2C-Scope
Communit
Version: 1.2.3
Date: 2/1/17
Source: Microch
Homepage: ht
Plugin Descrip

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.

For more details refer Microchip Application note AN957 "Sensored BLDC Motor Control Using dsPIC30F2010" 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:

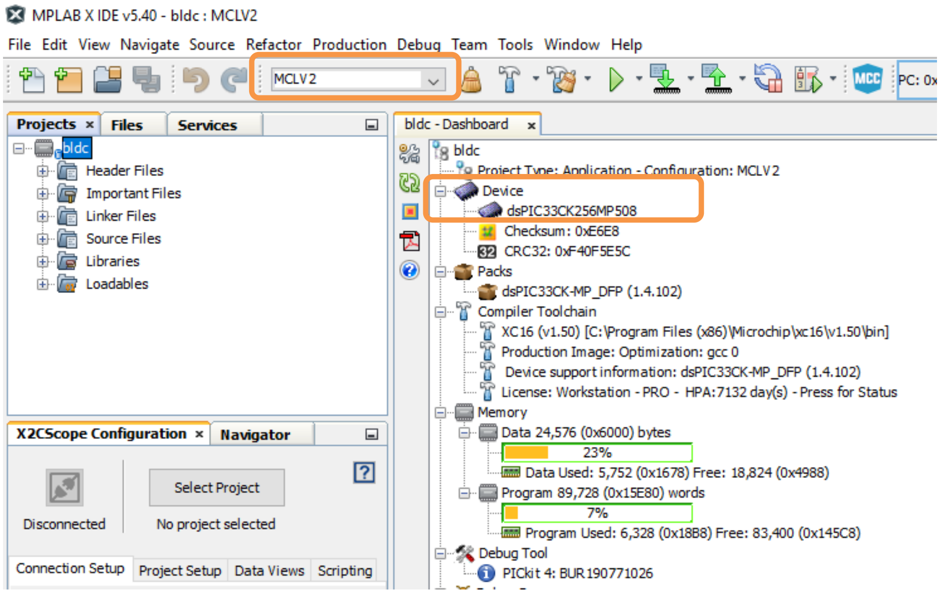
- Shorten the name of the directory containing the firmware used in this demonstration. In this case, rename directory `AN957_dsPIC33CK256MP508_EXT_OPAMP_MCLV2_MCHV2_MCHV3` to more appropriate shorter name. In case you renamed the directory, consider the new name while reading instructions provided in 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

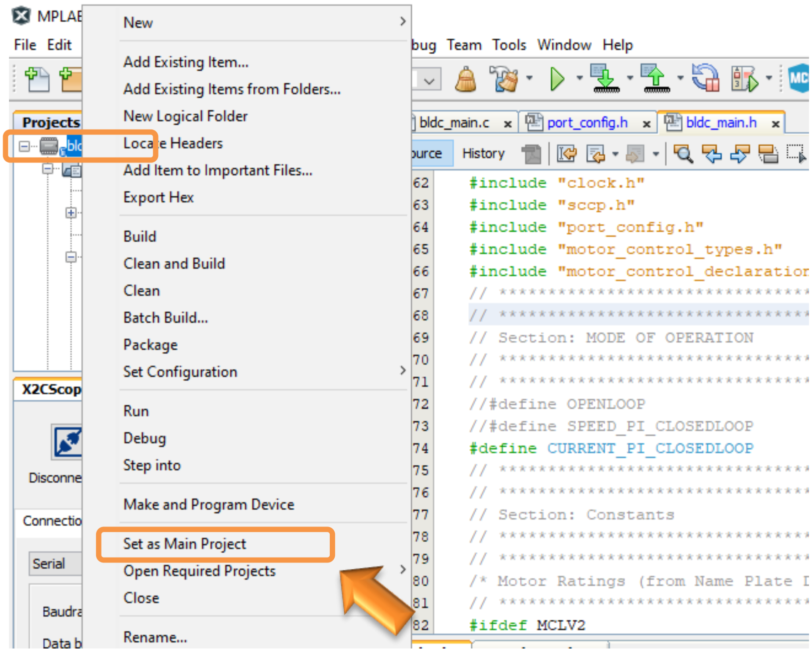
Follow below instructions step by step to setup and run the motor control demo application:

1. Start MPLAB X IDE and open `(File>Open Project)` the project `bldc.X` (`..\AN957_dsPIC33CK256MP508_EXT_OPAMP_MCLV2_MCHV2_MCHV3\bldc.X`) with device selection *dsPIC33CK256MP508*.



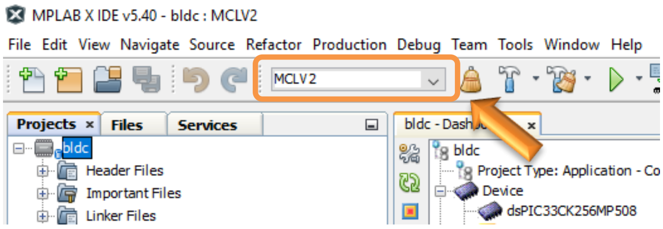
bldc
Project Type: Application - Configuration: MCLV2
Device
dsPIC33CK256MP508
Checksum: 0xE6E8
CRC32: 0xF40F5E5C
Packs
dsPIC33CK-MP_DFP (1.4.102)
Compiler Toolchain
XC16 (v1.50) [C:\Program Files (x86)\Microchip\xc16\v1.50\bin]
Production Image: Optimization: gcc 0
Device support information: dsPIC33CK-MP_DFP (1.4.102)
License: Workstation - PRO - HPA:7132 day(s) - Press for Status
Memory
Data 24,576 (0x6000) bytes
23%
Data Used: 5,752 (0x1678) Free: 18,824 (0x4988)
Program 89,728 (0x15E80) words
7%
Program Used: 6,328 (0x18B8) Free: 83,400 (0x145C8)
Debug Tool
PICkit 4: BUR190771026

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



```
#include "clock.h"
#include "sccp.h"
#include "port_config.h"
#include "motor_control_types.h"
#include "motor_control_declaratior
// ********************************
// ********************************
// Section: MODE OF OPERATION
// ********************************
// ********************************
//#define OPENLOOP
//#define SPEED_PI_CLOSEDLOOP
#define CURRENT_PI_CLOSEDLOOP
// ********************************
// ********************************
// Section: Constants
// ********************************
// ********************************
/* Motor Ratings (from Name Plate I
// ********************************
#ifdef MCLV2
```

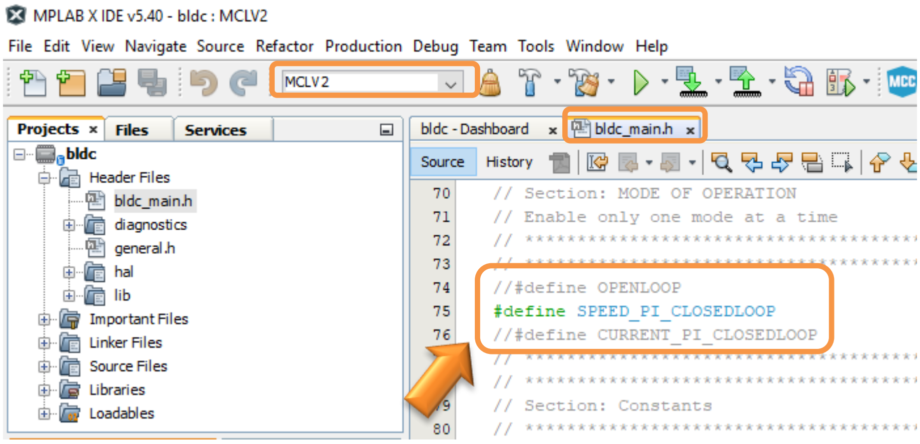
3. Select project configuration as "`MCLV2`" from the Project Configuration drop down box on the toolbar as shown:


MPLAB X IDE v5.40 - bldc : MCLV2
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
MCLV2
Projects
Files
Services
bldc
Header Files
Important Files
Linker Files
bldc - Dashb

bldc
Project Type: Application - Co
Device
dsPIC33CK256MP508

There may be multiple project configurations available for *bldc.X*.

4. Open `bldc_main.h` (under `bldc.X -> headerfiles`) in the project `bldc.X` and ensure any one of the modes of operation is defined. The user may choose any one of the available modes i.e. `OPENLOOP`, `SPEED_PI_CLOSEDLOOP`, and `CURRENT_PI_CLOSEDLOOP`. (ensure only one is selected at a time)

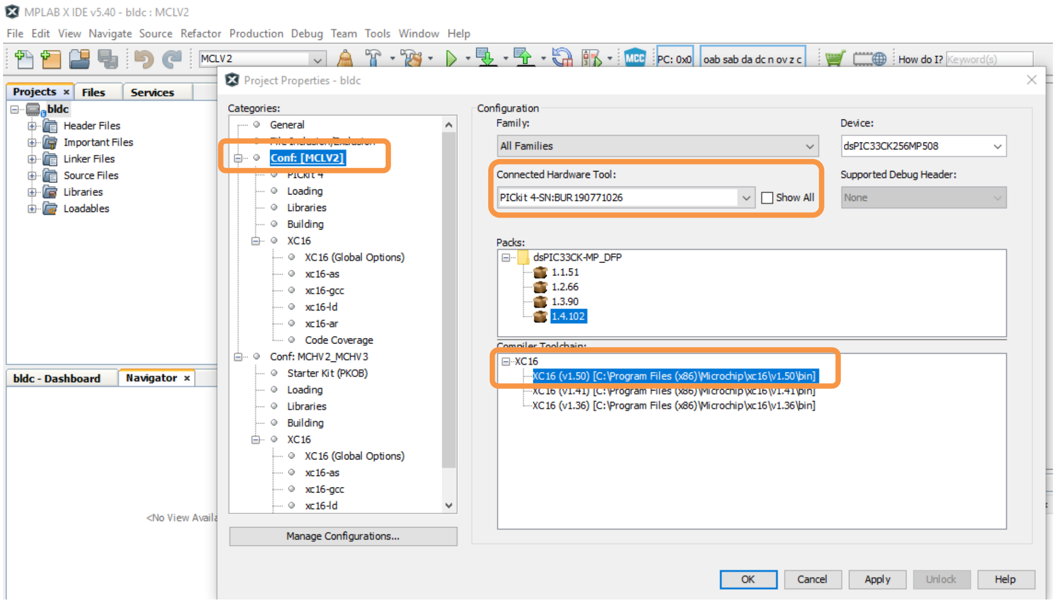


```
// Section: MODE OF OPERATION
// Enable only one mode at a time
// ***********************************
// ***********************************
//#define OPENLOOP
#define SPEED_PI_CLOSEDLOOP
//#define CURRENT_PI_CLOSEDLOOP
// ***********************************
// ***********************************
// Section: Constants
// ***********************************
```

5. Right click on the project *bldc.X* and select “Properties” to open its Project Properties Dialog. Click the “Conf: [MCLV2]” category to reveal the general project configuration information.

   In the ***‘Conf-MCLV2’*** 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(v1.50)” is selected. The compiler used for testing the firmware is listed in the section 2.2 Software Tools Used for Testing the firmware.
   - 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**



Family: All Families

Device: dsPIC33CK256MP508

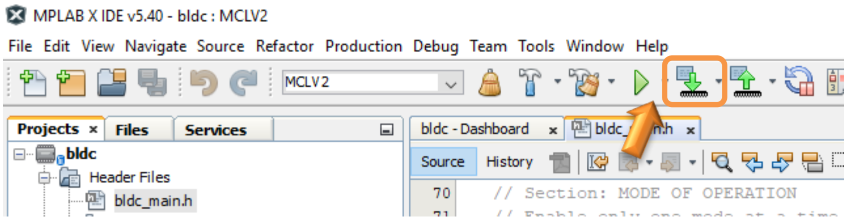
Connected Hardware Tool: PICkit 4-SN:BUR190771026 ☐ Show All

Supported Debug Header: None

Packs:
dsPIC33CK-MP_DFP
1.1.51
1.2.66
1.3.90
1.4.102

Compiler Toolchain:
XC16
XC16 (v1.50) [C:\Program Files (x86)\Microchip\xc16\v1.50\bin]
XC16 (v1.41) [C:\Program Files (x86)\Microchip\xc16\v1.41\bin]
XC16 (v1.36) [C:\Program Files (x86)\Microchip\xc16\v1.36\bin]

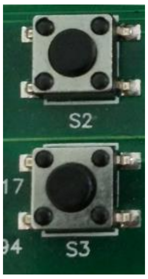
6. To build the project (in this case `bldc.X`) and program the device dsPIC33CK256MP508, click “**Make and Program Device Main project**” on the toolbar.


MPLAB X IDE v5.40 - bldc : MCLV2
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
MCLV2
Projects
Files
Services
bldc
Header Files
bldc_main.h
bldc - Dashboard
bldc_main.h
Source
History

```
70    // Section: MODE OF OPERATION
```

7. If the device is successfully programmed, **LED D17** will be turned ON, indicating that the dsPIC® DSC is enabled.

8. Run or Stop the motor by pressing the push button **S2**. The function of the pushbutton **S2** (Run/Stop of the motor) is indicated by turning ON or OFF the **LED D2**.



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

10. To reverse the direction of rotation, press the push button **S3**.
11. Press the push button **S2** to stop the motor.

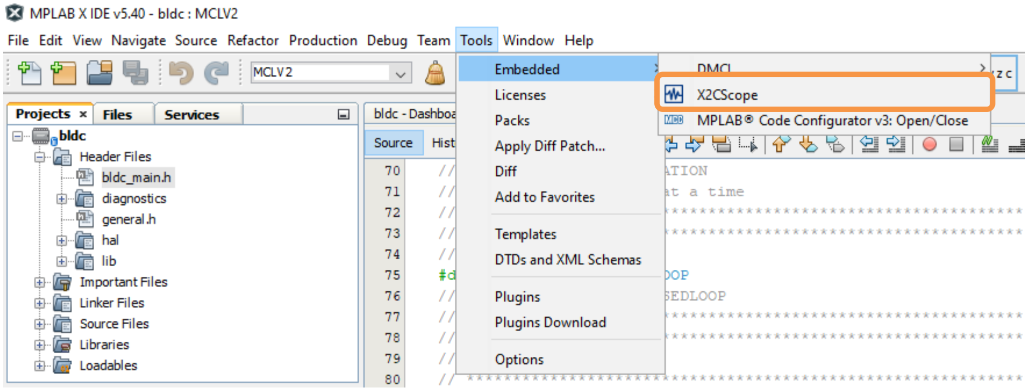
Note:

The macro definitions `MAX_MOTORSPEED, MAX_MOTORCURRENT, POLEPAIRS, SECTOR and MAX_BOARDCURRENT` are specified in `bldc_main.h` file included in the project `bldc.X`. The definitions `MAX_MOTORSPEED and MAX_MOTORCURRENT` are defined as per the specification provided by the Motor manufacturer. *Exceeding manufacturer specification may lead to damage to the motor or (and) the board.*

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.

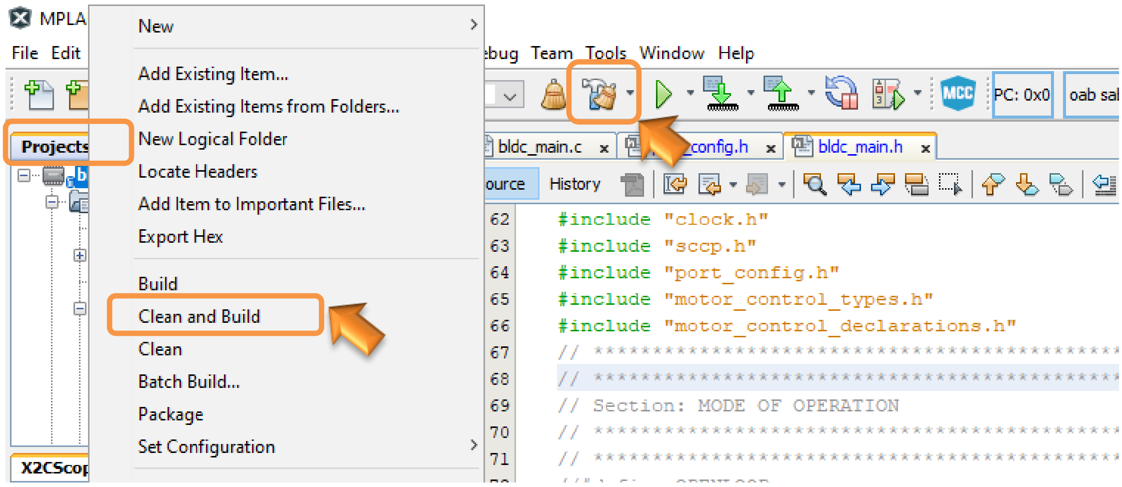
1. Ensure X2C Scope Plug-in is installed. For additional information on how to set up a plug-in refer to https://microchipdeveloper.com/mplabx:tools-plugins-available



2. To utilize X2C 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 and install USB drivers if necessary.

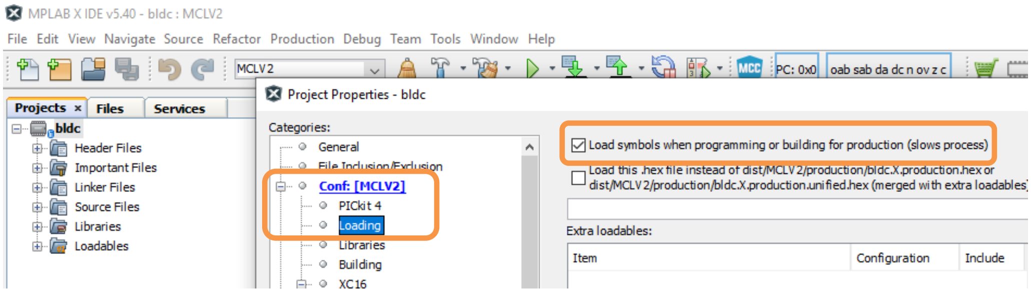
3. Ensure application is configured and running as described under Section 5.2 Basic Demonstration by following steps 1 through 11.

4. Build the project *bldc.X*. To do that right click on the project *bldc.X* and select “Clean and Build” or click the icon on the toolbar as shown.



```
#include "clock.h"
#include "sccp.h"
#include "port_config.h"
#include "motor_control_types.h"
#include "motor_control_declarations.h"
// ****************************************************
// ****************************************************
// Section: MODE OF OPERATION
// ****************************************************
// ****************************************************
```

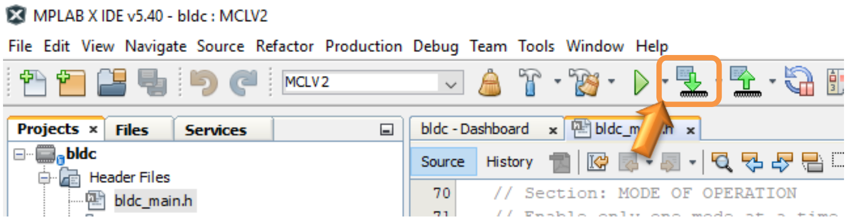
5. Please ensure that the checkbox “Load symbols when programming or building for production (slows process)” is checked, which is under the “Loading” category of the Project Property dialog.



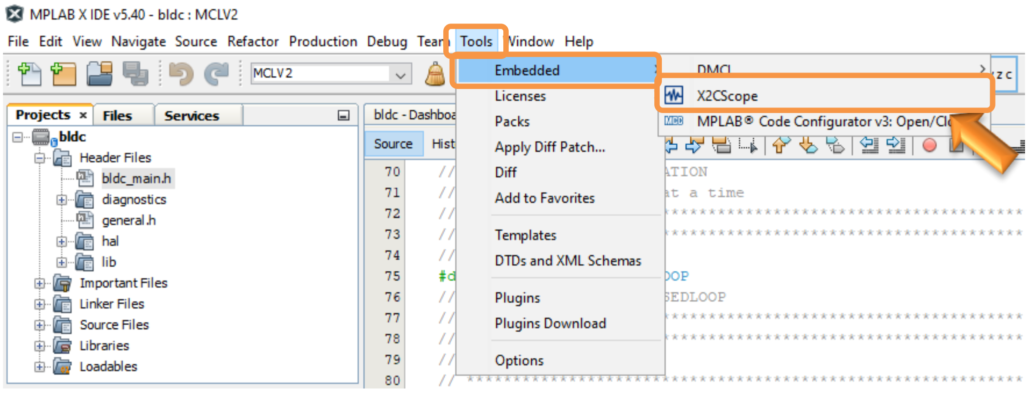
☑ Load symbols when programming or building for production (slows process)

□ Load this .hex file instead of dist/MCLV2/production/bldc.X.production.hex or dist/MCLV2/production/bldc.X.production.unified.hex (merged with extra loadables)

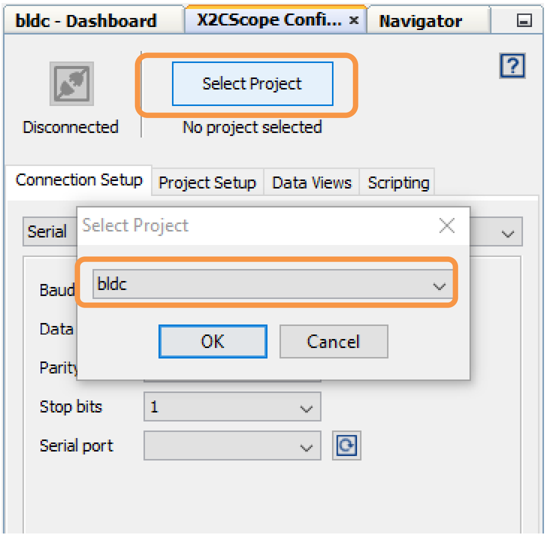
6. To build the project (in this case `bldc.X`) and program the device dsPIC33CK256MP508, click “**Make and Program Device Main project**” on the toolbar.


MPLAB X IDE v5.40 - bldc : MCLV2
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
MCLV2
Projects
Files
Services
bldc
Header Files
bldc_main.h
bldc - Dashboard
bldc_main.h
Source
History
70 // Section: MODE OF OPERATION
71 // Enable only one mode at a time

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



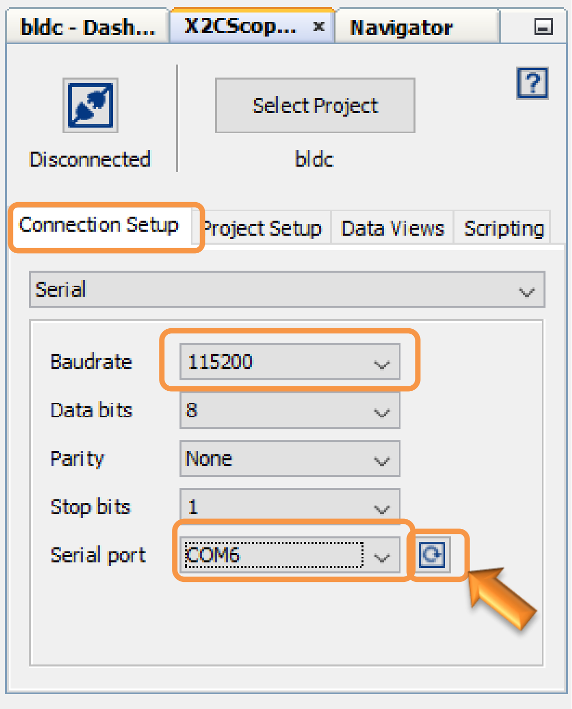
8. Open the X2CScope Configuration window and in “Select project” menu, select bldc project as shown.



Select Project: bldc

Stop bits: 1

9. Remote Communication needs to be established, as indicated in the following figure. Ensure the communication baud rate is set to 115200 as the same is set in the application firmware, while COM port used depends on the system settings. Refresh button lists the available COM Ports. Select the COM Port as per the connection.



Serial

Baudrate 115200

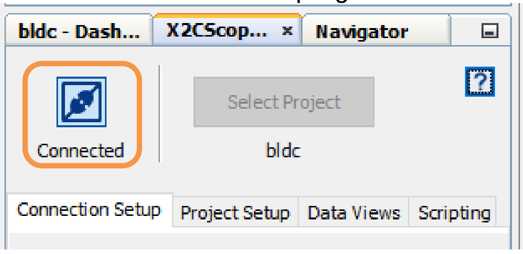
Data bits 8

Parity None

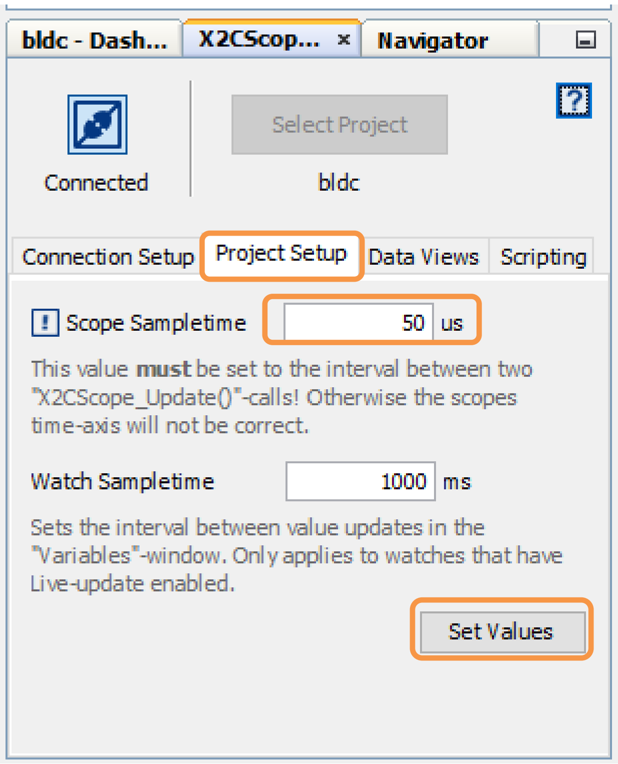
Stop bits 1

Serial port COM6

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



11. Set the “Project Setup” as shown below and click “Set Values”. Set Scope sample time as interval at which X2CScopeUpdate() is called. In this application it is every 20kHz (50µs).



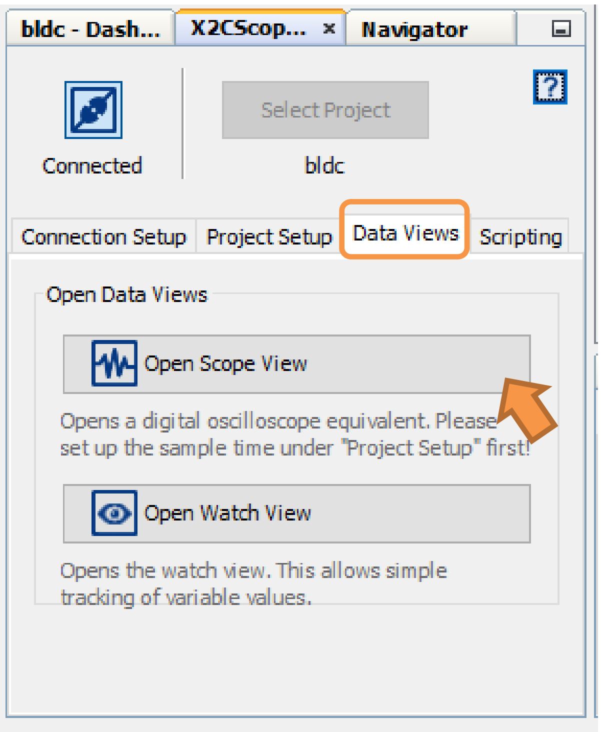
Scope Sampletime 50 us

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.

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

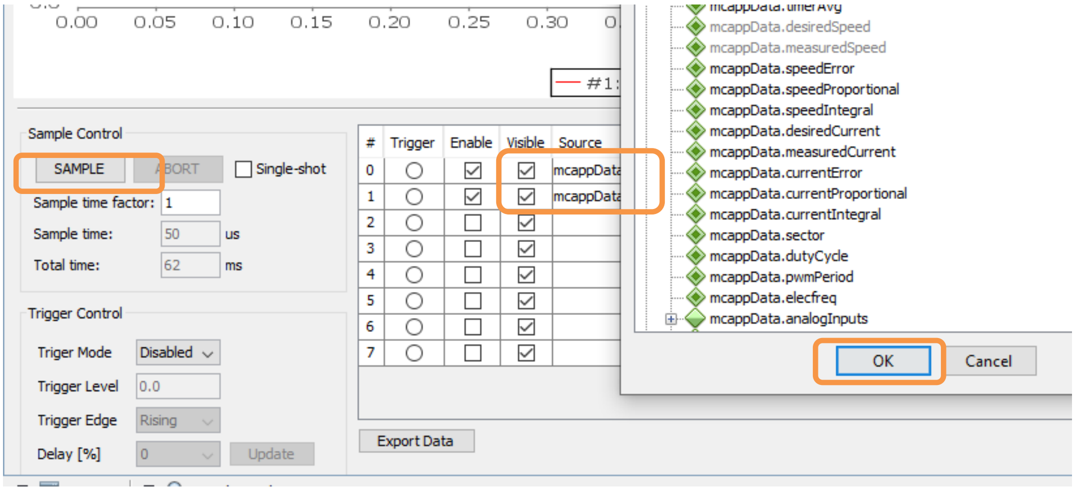


Opens a digital oscilloscope equivalent. Please set up the sample time under "Project Setup" first!

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

13. In this window, select the variables that needs to be monitored. To do this, click on the source against each channel, a window Select Variables opens upon the screen. From the available list, the required variable can be chosen. Ensure check boxes Enable & Visible are checked for the variables to be plotted

    To view data plots continuously, uncheck Single-shot. When Single-shot is checked it captures the data once and stops. The Sample time factor value multiplied with Sample time determines 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: 62 ms

Trigger Control

Triger Mode: Disabled

Trigger Level: 0.0

Trigger Edge: Rising

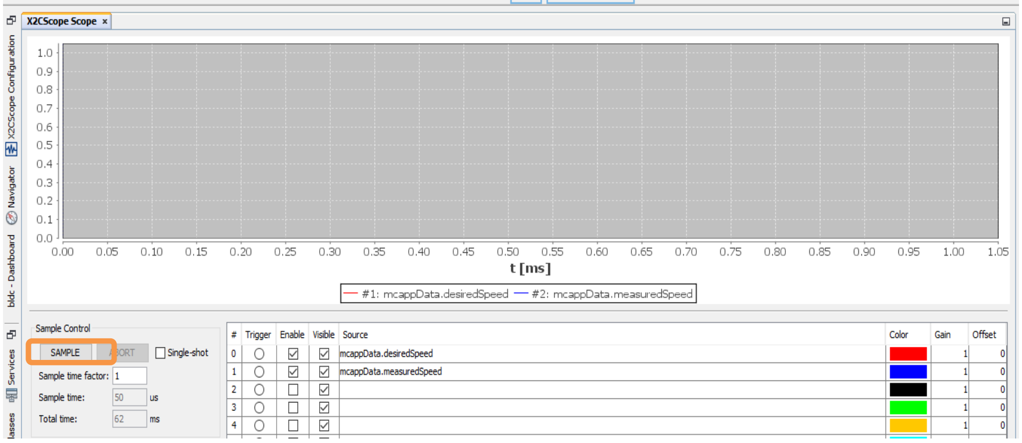
Delay [%]: 0

Update

#	Trigger	Enable	Visible	Source
0	○	☑	☑	mcappData
1	○	☑	☑	mcappData
2	○	□	☑	
3	○	□	☑	
4	○	□	☑	
5	○	□	☑	
6	○	□	☑	
7	○	□	☑	

mcappData.timerAvg
mcappData.desiredSpeed
mcappData.measuredSpeed
mcappData.speedError
mcappData.speedProportional
mcappData.speedIntegral
mcappData.desiredCurrent
mcappData.measuredCurrent
mcappData.currentError
mcappData.currentProportional
mcappData.currentIntegral
mcappData.sector
mcappData.dutyCycle
mcappData.pwmPeriod
mcappData.elecfreq
mcappData.analogInputs

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



Sample Control

SAMPLE ABORT □ Single-shot

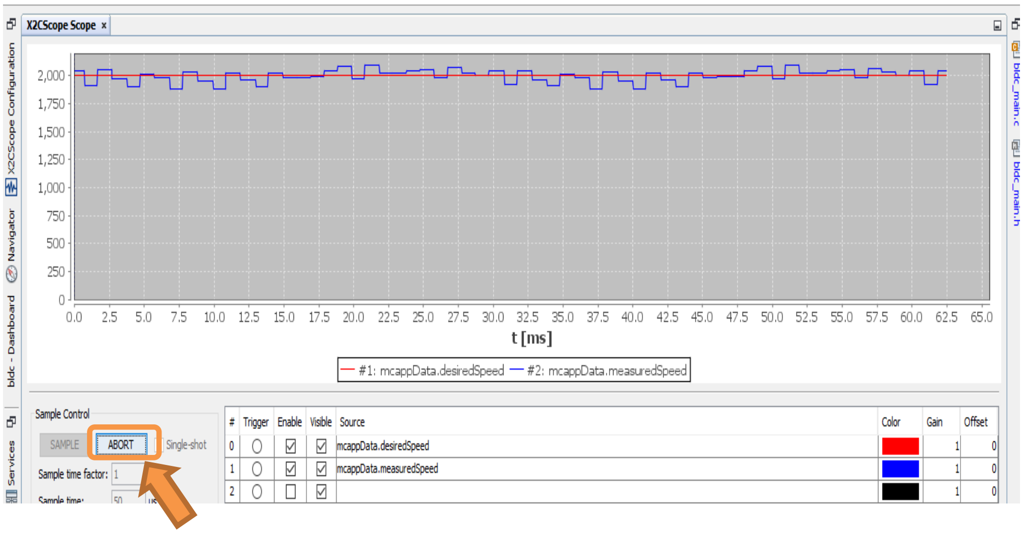
Sample time factor: 1

Sample time: 50 us

Total time: 62 ms

#	Trigger	Enable	Visible	Source	Color	Gain	Offset
0	○	☑	☑	mcappData.desiredSpeed	🟥	1	0
1	○	☑	☑	mcappData.measuredSpeed	🟦	1	0
2	○	□	☑		⬛	1	0
3	○	□	☑		🟩	1	0
4	○	□	☑		🟨	1	0

15. Click on ABORT to stop.



Sample Control

SAMPLE ABORT □ Single-shot

Sample time factor: 1

Sample time: 50 us

#	Trigger	Enable	Visible	Source	Color	Gain	Offset
0	○	☑	☑	mcappData.desiredSpeed	🟥	1	0
1	○	☑	☑	mcappData.measuredSpeed	🟦	1	0
2	○	□	☑		⬛	1	0

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 AN957 motor control application firmware demonstrated using the Development Board and the dsPIC33CK256MP508 External Op-Amp Motor Control PIM(MA330041-1). Refer "*dsPIC33CK256MP508 External Op-Amp Motor Control Plug-in-Module (PIM) Information Sheet (DS50002757a)"* 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
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:43	30	**AN17**/ANN1/IBIAS1/RP54/PMD12/PMA12/**RC6**	Analog Input	Connected to bus current through External Op-Amp Matrix Board and Jumper J13
Hall Sensor Connections					
HALL A	PIM:80	73	RP65/PWM4H/**RD1**	Digital Input	Connected to PIM (5V tolerant)
HALL B	PIM:47	72	RP66/**RD2**	Digital Input	Connected to PIM (5V tolerant)
HALL C	PIM:48	69	RP67/ASCL3/**RD3**	Digital Input	Connected to PIM (5V tolerant)
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 X2C-Scope.
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 X2C-Scope.

8. REFERENCES:

For additional information, refer following documents or links.

1. AN957 Application Note “Sensored BLDC Motor Control Using dsPIC30F2010”
2. dsPICDEM™ MCLV-2 Development Board User’s Guide (DS52080)
3. dsPIC33CK256MP508 External Op-Amp Motor Control Plug-in-Module (PIM) Information Sheet (DS50002757)
4. dsPIC33CK256MP508 Family datasheet (DS70005349).
5. Family Reference manuals (FRM) of dsPIC33CK256MP508 family
6. MPLAB® X IDE User’s Guide (DS50002027) or MPLAB® X IDE help
7. MPLAB® X IDE installation
8. MPLAB® XC16 Compiler installation