AN1292 Demonstration ReadMe for the dsPICDEM™ MCLV-2 Development Board with the dsPIC33CH512MP508 Motor Control PIM (MPLAB® X IDE)

1. INTRODUCTION

This document describes the setup requirements for running the Sensor-less FOC algorithm with a PLL Estimator, which is referenced in AN1292 "*Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)"* using 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 in External Op-amp configuration with the dsPIC33CH512MP508 Motor Control Plug-In Module(PIM).

2. SUGGESTED DEMONSTRATION REQUIREMENTS

2.1. Motor Control Application Firmware Required for the Demonstration

- `AN1292_dsPIC33CH512MP508_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 v5.10
- MPLAB® XC16 Compiler v1.36b
- MPLAB X IDE Plugin: Data Monitor and Control Interface (DMCI) v2.71

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)
- dsPIC33CH512MP508 Motor Control Plug-in module (MA330045)
- Microchip Programmer tools like MPLAB® REAL ICE™ In-Circuit Emulator (DV244005) or MPLAB® ICD 3 (DV164035) etc.

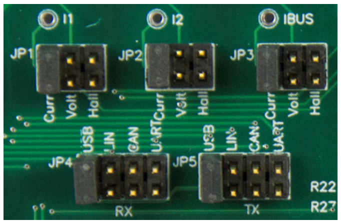
Note:

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

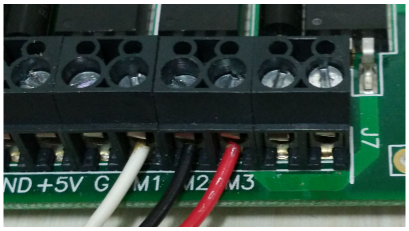
3. HARDWARE SETUP

This section describes hardware setup required for the demonstration. Motor phase current feedbacks needed by the firmware are amplified by the operational amplifiers provided on the dsPICDEM™ MCLV-2 Development Board. This is referred as ‘*external amplifier configuration*’.

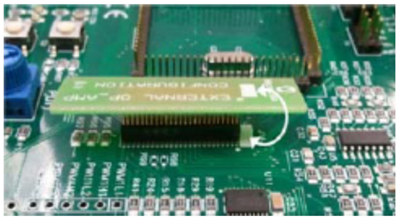
1. Disconnect power to the dsPICDEM™ MCLV-2 Development Board and set up the following jumpers:

Jumper	Pins to Short	Board Reference
JP1	1-2(**Curr**-Current)	
JP2	1-2(**Curr**-Current)	
JP3	1-2(**Curr**-Current)	
JP4	USB position	
JP5	USB position	
J5	Don’t care	
JP11	Don’t care	

2. Connect the three phase wires from the motor to M1, M2, and M3 terminals of connector J7(there is no specific order), provided on the dsPICDEM™ MCLV-2 Development Board.

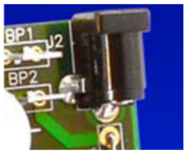


3. Connect the ‘External Op Amp Configuration Matrix board’ to matrix board header J14. Ensure the matrix board is correctly oriented before proceeding.



4. Insert the dsPIC33CH512MP508 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.

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



6. 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 USB port of 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 dsPIC33CH512MP508 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

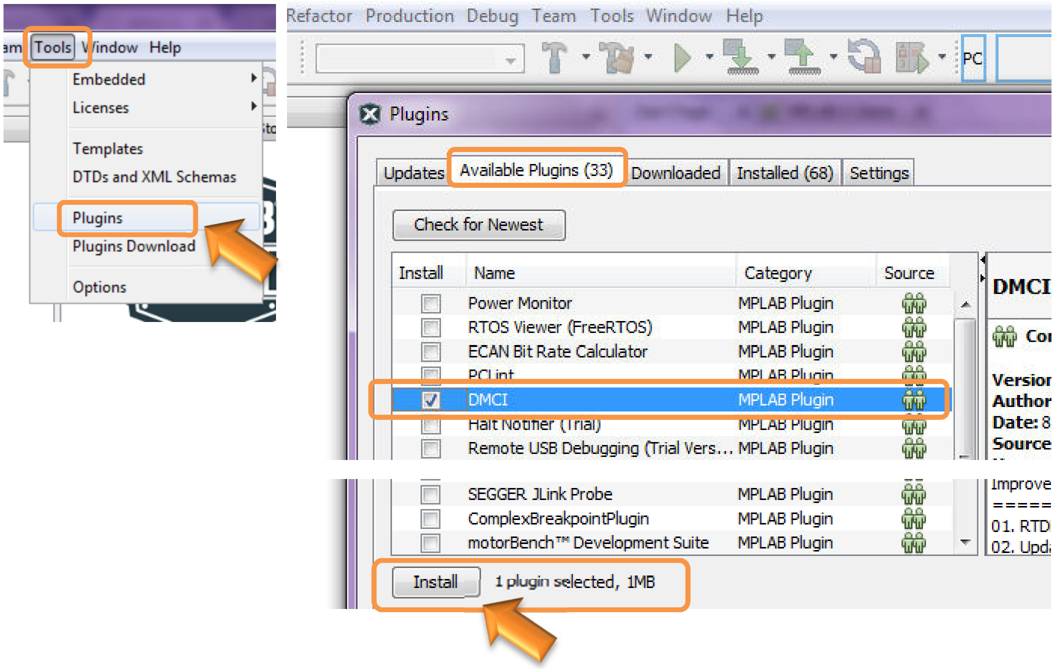
- MPLAB X IDE installation, refer link
- MPLAB XC16 Compiler installation steps, refer link

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.



Install	Name	Category	Source
□	Power Monitor	MPLAB Plugin	
□	RTOS Viewer (FreeRTOS)	MPLAB Plugin	
□	ECAN Bit Rate Calculator	MPLAB Plugin	
□	PCLint	MPLAB Plugin	
☑	DMCI	MPLAB Plugin	
□	Halt Notifier (Trial)	MPLAB Plugin	
□	Remote USB Debugging (Trial Vers...	MPLAB Plugin	
□	SEGGER JLink Probe	MPLAB Plugin	
□	ComplexBreakpointPlugin	MPLAB Plugin	
□	motorBench™ Development Suite	MPLAB Plugin	

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.

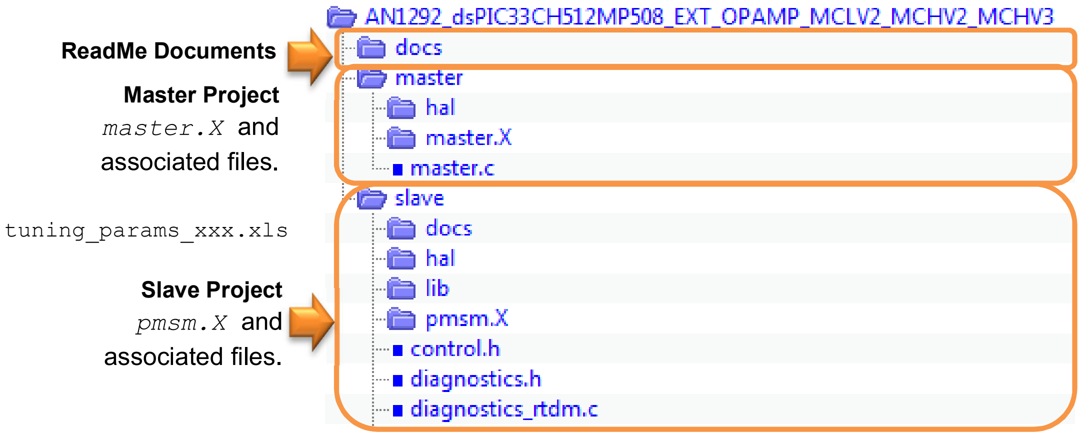
This firmware is implemented to work on Microchip's dual core 16-bit Digital signal controller (dsPIC® DSC) dsPIC33CH512MP508.There are two independent dsPIC DSC Cores called 'Master Core' and 'Slave Core' in the device. For more information, see the *dsPIC33CH512MP508 Family datasheet*.

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

- Device selection in Master Project (code for Master Core) is dsPIC33CH512MP508
- Device selection in Slave Project (code for Slave Core) is dsPIC33CH512MP508S1

Hence the firmware used in this demonstration consists of two MPLAB X projects `master.X` (referred as Master Project) and `pmsm.X` (referred as Slave Project).

The firmware directory structure is shown below:



```
AN1292_dsPIC33CH512MP508_EXT_OPAMP_MCLV2_MCHV2_MCHV3
├── docs
├── master
│   ├── hal
│   ├── master.X
│   └── master.c
└── slave
    ├── docs
    ├── hal
    ├── lib
    ├── pmsm.X
    ├── control.h
    ├── diagnostics.h
    └── diagnostics_rtdm.c
```

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 AN1292_dsPIC33CH512MP508_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".*

Function of the Master Core (as defined in the Master Project `master.X`) is:

- To set device Configuration bits applicable for both Master and Slave cores (As Configuration bits for Master and Slave cores are available in Master core registers). Note that the I/O port ownership between Master Core and Slave Core is decided by the Configuration bits.
- Configure Master Core Oscillator Subsystem to generate clocks needed for the operation of Core and its peripherals. In the firmware Master is configured to operate at 90MHz.
- To program and enable the Slave core by invoking XC16 library (libpic30.h) routines `_program_slave ()` and `_start_slave ()`.

Function of the Slave Core (as defined in the Slave Project `pmsm.X`) is:

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

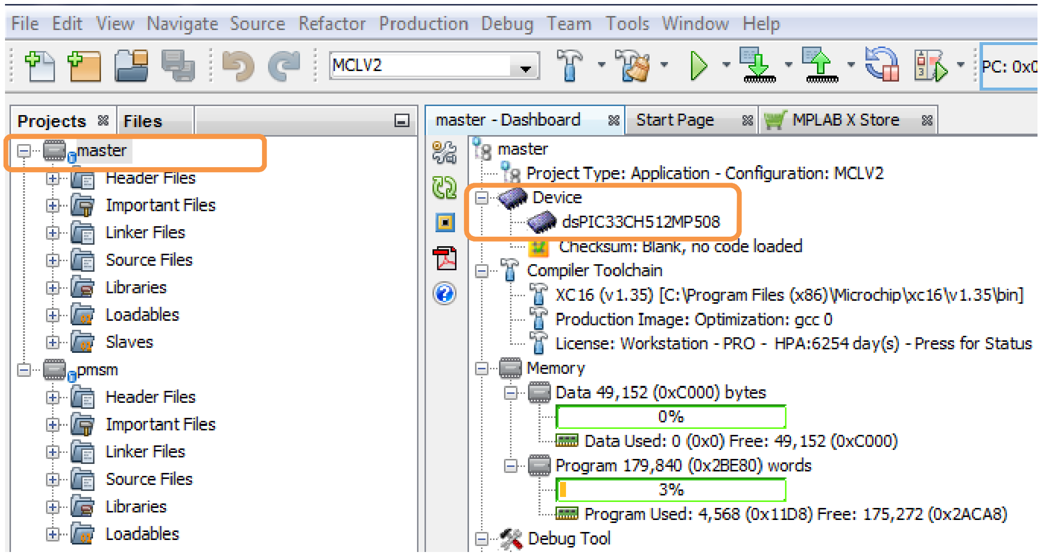
Once Master Core programs and enables the Slave Core, it can autonomously run the Motor Control Demo application residing in its PRAM. The Motor Control Demo application uses push button to start or stop the motor and potentiometer to vary speed of the motor.

For more details refer Microchip Application note 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

5.2. Basic Demonstration

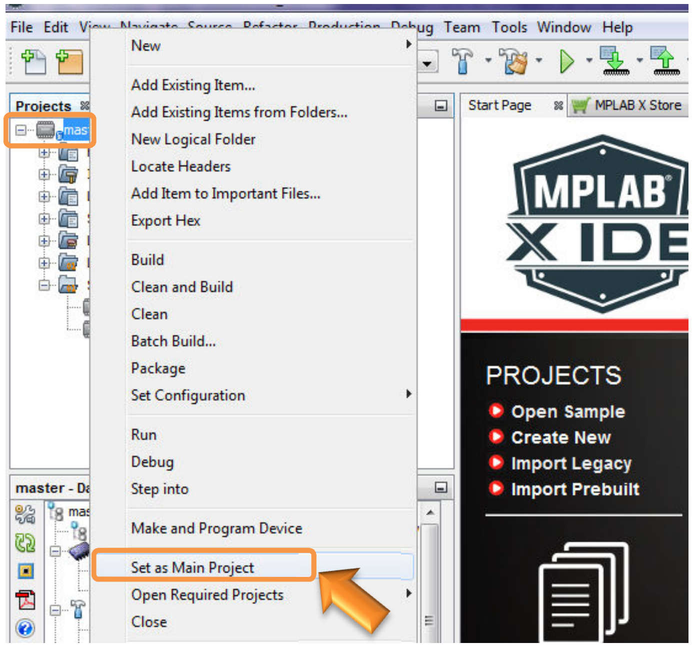
Follow below instructions step by step to setup and run the motor control demo application:

1. Start MPLAB X IDE and open (*File>Open Project*) the Master Project `master.X` (..`\AN1292_dsPIC33CH512MP508_EXT_OPAMP_MCLV2_MCHV2_MCHV3\master\master.X`) with device selection *dsPIC33CH512MP508* (Master Core).

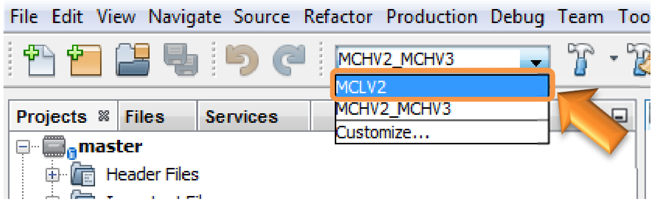


master
Project Type: Application - Configuration: MCLV2
Device
dsPIC33CH512MP508
Checksum: Blank, no code loaded
Compiler Toolchain
XC16 (v1.35) [C:\Program Files (x86)\Microchip\xc16\v1.35\bin]
Production Image: Optimization: gcc 0
License: Workstation - PRO - HPA:6254 day(s) - Press for Status
Memory
Data 49,152 (0xC000) bytes
0%
Data Used: 0 (0x0) Free: 49,152 (0xC000)
Program 179,840 (0x2BE80) words
3%
Program Used: 4,568 (0x11D8) Free: 175,272 (0x2ACA8)
Debug Tool

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

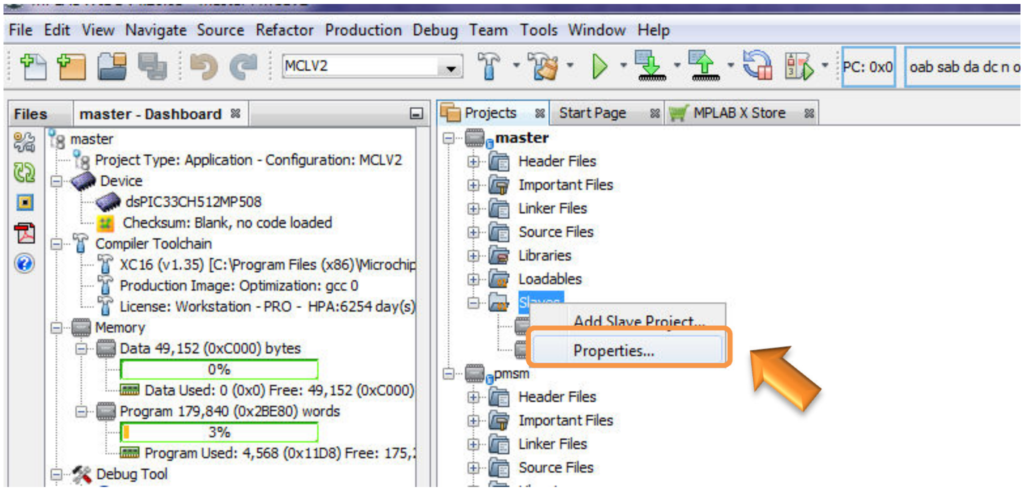


3. Select project configuration as “MCLV2” from the Project Configuration drop down box on the toolbar as shown:



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

4. In the Projects window, right click on the Slaves folder of the project tree (of Master project `master.X)` and select “Properties”. This will open the “Slaves” category of Project Properties dialog.



master
Project Type: Application - Configuration: MCLV2
Device
dsPIC33CH512MP508
Checksum: Blank, no code loaded
Compiler Toolchain
XC16 (v1.35) [C:\Program Files (x86)\Microchip
Production Image: Optimization: gcc 0
License: Workstation - PRO - HPA:6254 day(s)
Memory
Data 49,152 (0xC000) bytes
0%
Data Used: 0 (0x0) Free: 49,152 (0xC000)
Program 179,840 (0x2BE80) words
3%
Program Used: 4,568 (0x11D8) Free: 175,2
Debug Tool

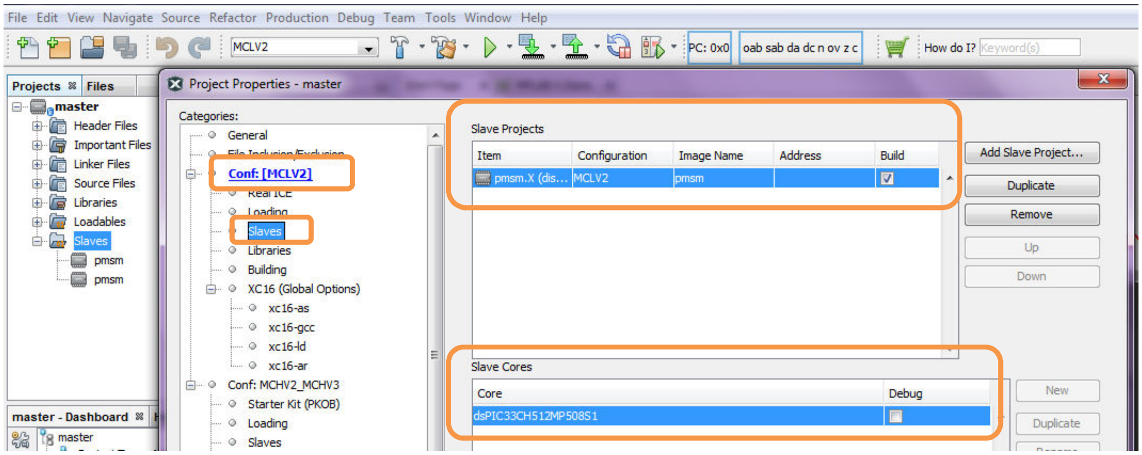
Verify “Slaves” category of Project Properties dialog, and ensure details are as follows (see figure):

- Item is `pmsm.X`
- Configuration is *“MCLV2”*
- Image name is *“pmsm”* and
- Check Box *“Build’* is checked
- Check Box *“Debug”* is unchecked

Note:

You may encounter build error,

- If any of the values are not as mentioned above (or as shown figure below) or
- If the slave project `pmsm.X` is moved or deleted from the firmware directory (in this case ‘A1292_dsPIC33CH512MP508_EXT_OPAMP_MCLV2_MCHV2_MCHV3’



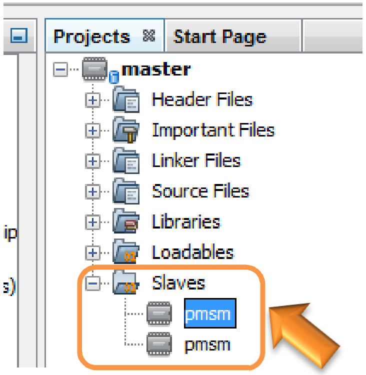
Slave Projects

Item	Configuration	Image Name	Address	Build
pmsm.X (dis...	MCLV2	pmsm		☑

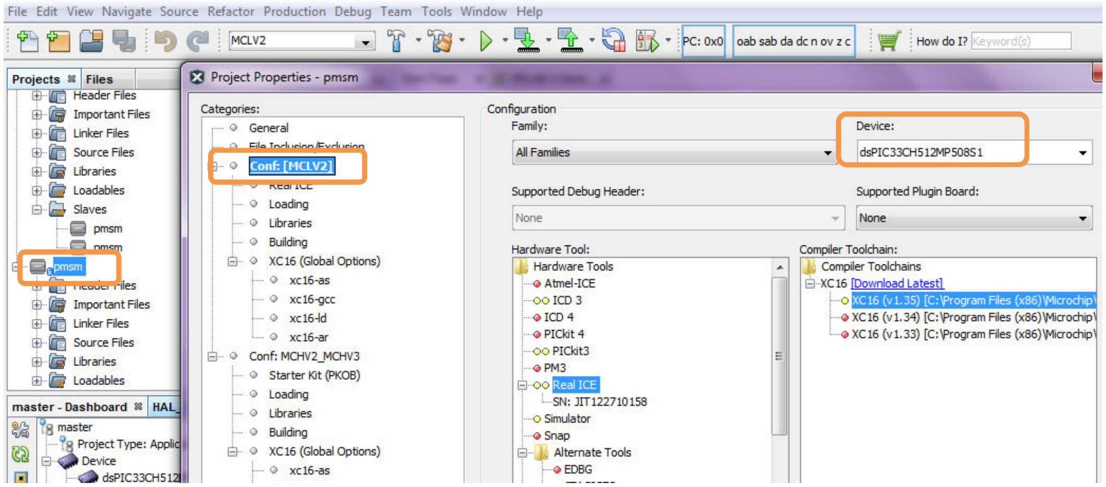
Slave Cores

Core	Debug
dsPIC33CH512MP508S1	□

5. In the Projects window, right click on the Slaves folder of the project tree (of Master project `master.X)` and select any one of the project "pmsm". This will open the Slave project `pmsm.X` in MPLAB X IDE project window. Alternatively, you can open (*File>Open Project*) the Slave project from its current location like any other MPLAB X project.



The selected device in Slave project `pmsm.X` can be viewed by opening its Project Property Dialog. As can be seen from the below figure, in this firmware it is set as dsPIC33CH512MP508S1(as shown in figure below), which represents Slave core of the dsPIC33CH512MP508.


Project Properties - pmsm

Categories: Conf: [MCLV2]

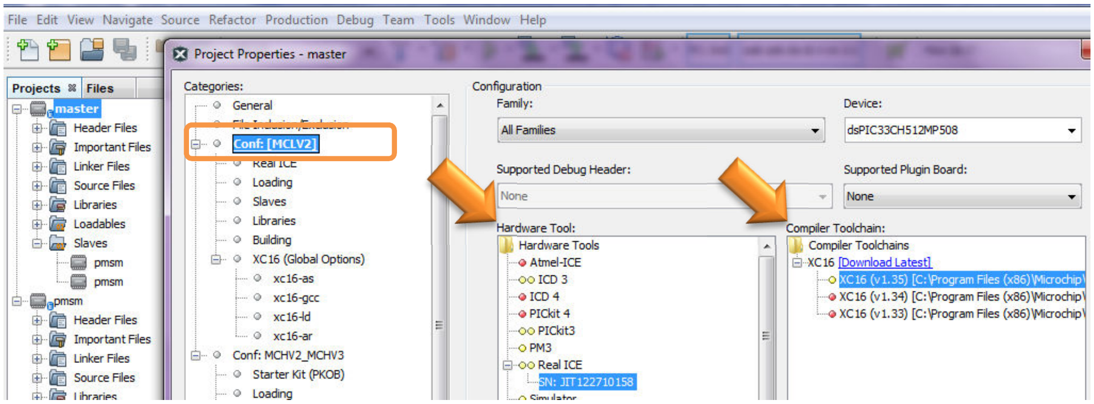
Configuration
Family: All Families
Device: dsPIC33CH512MP508S1
Supported Debug Header: None
Supported Plugin Board: None
Hardware Tool: Real ICE, SN: JIT122710158
Compiler Toolchain: XC16 (v1.35) [C:\Program Files (x86)\Microchip\

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

7. Right click on the Master Project `master.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 dsPIC33CH512MP508.In this case “XC16(v1.35)” 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: dsPIC33CH512MP508

Supported Debug Header: None

Supported Plugin Board: None

Hardware Tool: Real ICE SN: JIT122710158

Compiler Toolchain: XC16 (v1.35) [C:\Program Files (x86)\Microchip\

Note:

- There may be multiple project configurations like “MCLV2”, “MCHV2-MCHV3”, etc. available in both Master project *`master.X`* and Slave project *`pmsm.X`*.
- The project configuration “MCLV2” in the Master Project `master.X` is configured to associate with configuration “MCLV2” of Slave project *`pmsm.X`*.
- If multiple project configuration associations are defined between Master Project and Slave Project you will see as many slave projects listed under Slaves folder of the Master Project tree. In the above figure, you can see two “pmsm” projects, as two project configuration associations were defined between Master project *`master.X`* and Slave project *`pmsm.X`*

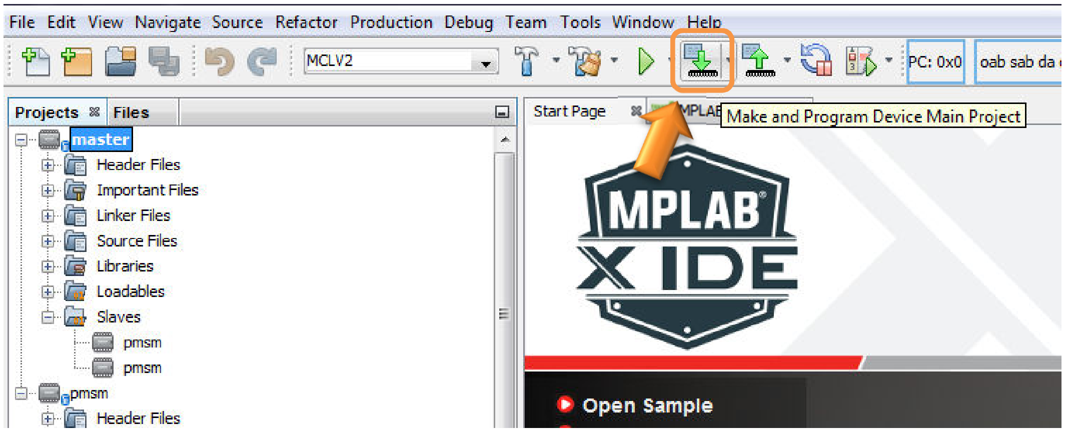
8. Right click on the associated Slave Project `pmsm.X` and select “Properties” to open its Project Properties Dialog. Click the “Conf: [MCLV2]” category to reveal the general project configuration information.
   In the ‘***Conf-MCLV2’*** category window:
   - Select the specific Compiler Toolchain from the available list of compilers. Please ensure MPLAB XC16 Compiler supports the device dsPIC33CH512MP508S1.
   - After selecting Compiler Toolchain, click button **Apply**

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

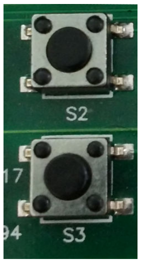
9. To build the Master Project (in this case `master.X`) and program the device dsPIC33CH512MP508, click “**Make and Program Device Main project**” on the toolbar. Upon this, MPLAB X IDE begin executing following activities in order:
   - builds Slave Project `pmsm.X` (linked to Master Project `master.X`)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

Note:

In this firmware configuration, the Master Core programs the Slave Core. When device is programmed, the Slave core image is placed in the Master flash. When the Master Core is powered on and begins execution of code, it transfers the Slave image from the Master flash to the Slave PRAM.



10. If the device is successfully programmed, the **LED D2** will be turned ON. This indicates the Slave Core is programmed and enabled by the Master Core.
11. 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 D17**.



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

13. To enter the extended speed range (`NOMINAL_SPEED_RPM` to `MAXIMUM_SPEED_RPM`) press the push button **S3**. Press the push button **S3** again to revert the speed of the motor to its normal speed (`END_SPEED_RPM` to `NOMINAL_SPEED_RPM`) range.

14. Press the push button **S2** to stop the motor.

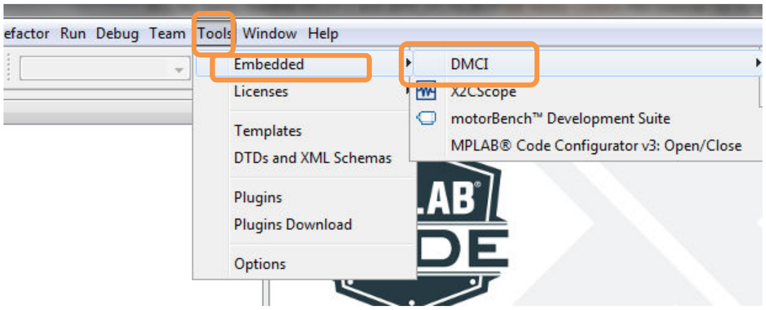
Note:

The macro definitions `END_SPEED_RPM, NOMINAL_SPEED_RPM`, and `MAXIMUM_SPEED_RPM` are specified in `userparms.h` file included in the Slave 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.*

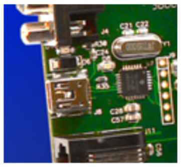
5.3. Data visualization through DMCI Plug-in of MPLABX

The Slave Project of the 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.

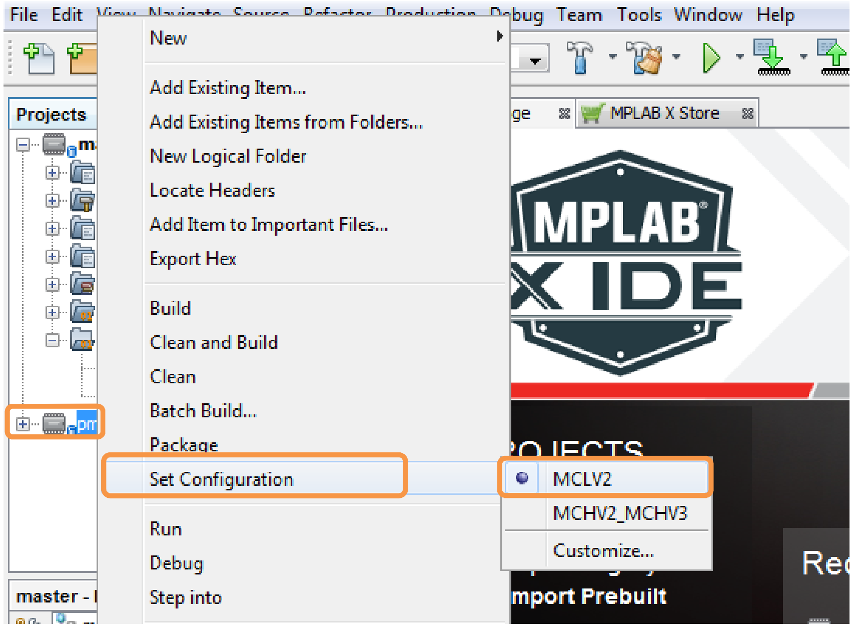

efactor Run Debug Team Tools Window Help
Embedded
Licenses
Templates
DTDs and XML Schemas
Plugins
Plugins Download
Options
DMCI
X2CScope
motorBench™ Development Suite
MPLAB® Code Configurator v3: Open/Close

2. To utilize RTDM 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.

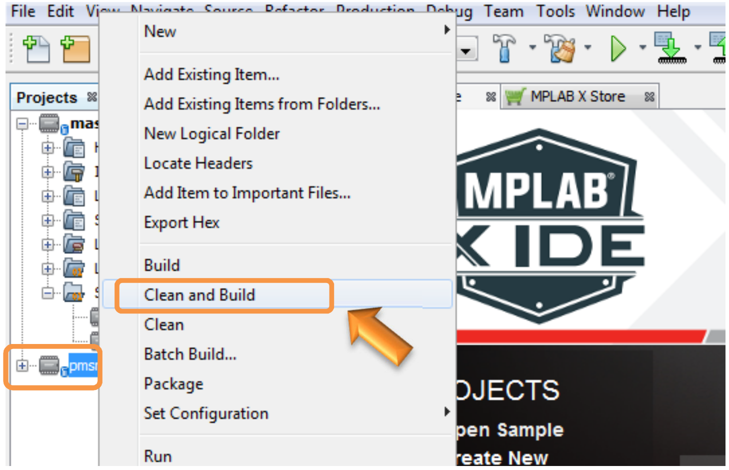


3. Ensure application is configured and running as described under Section Basic Demonstration by following steps 1 through 14.

4. Set Slave project configuration. To do that right click on the Slave Project `pmsm.X` and set project configuration (*File>Set Configuration>MCLV2)* as shown

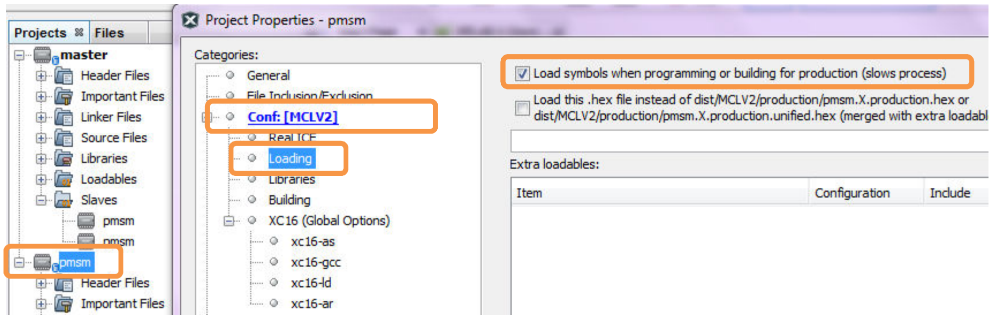


5. Build the Slave Project *pmsm.X*. To do that right click on the Slave project *pmsm.X* and select “Clean and Build”.



Note:

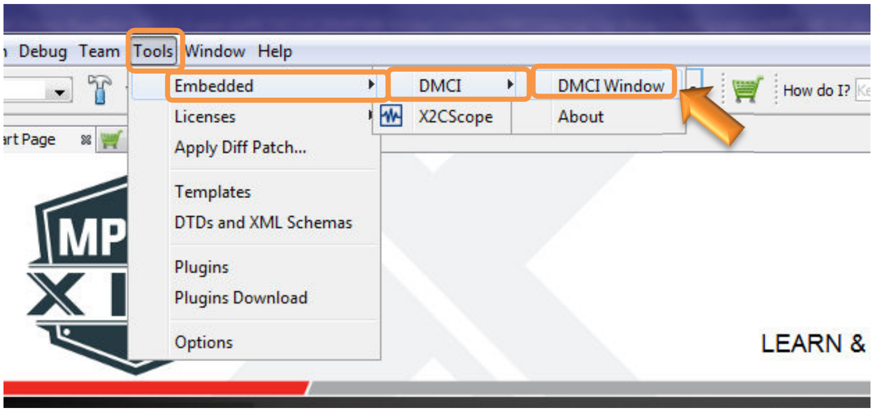
When using RTDM with DMCI you will need to check the checkbox “Load symbols when programming or building for production (slows process)” under the “Loading” category of the Project Property dialog.



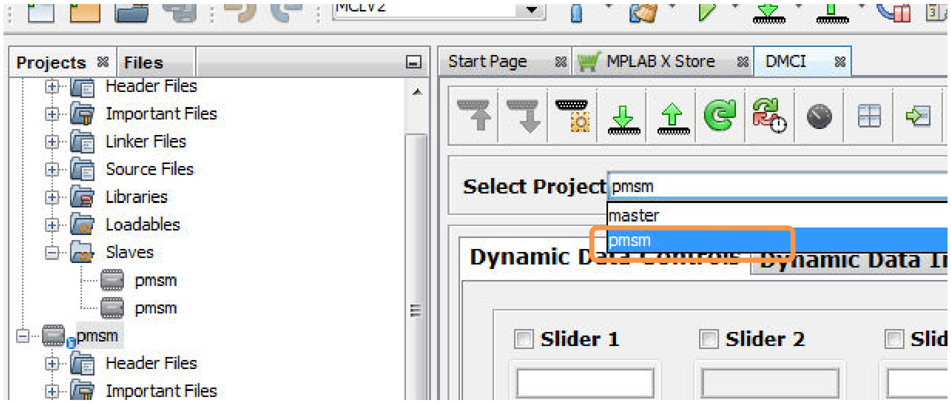
☑ Load symbols when programming or building for production (slows process)

□ Load this .hex file instead of dist/MCLV2/production/pmsm.X.production.hex or dist/MCLV2/production/pmsm.X.production.unified.hex (merged with extra loadabl

6. Open the DMCI window by selecting *Tools>Embedded>DMCI>DMCI Window.*

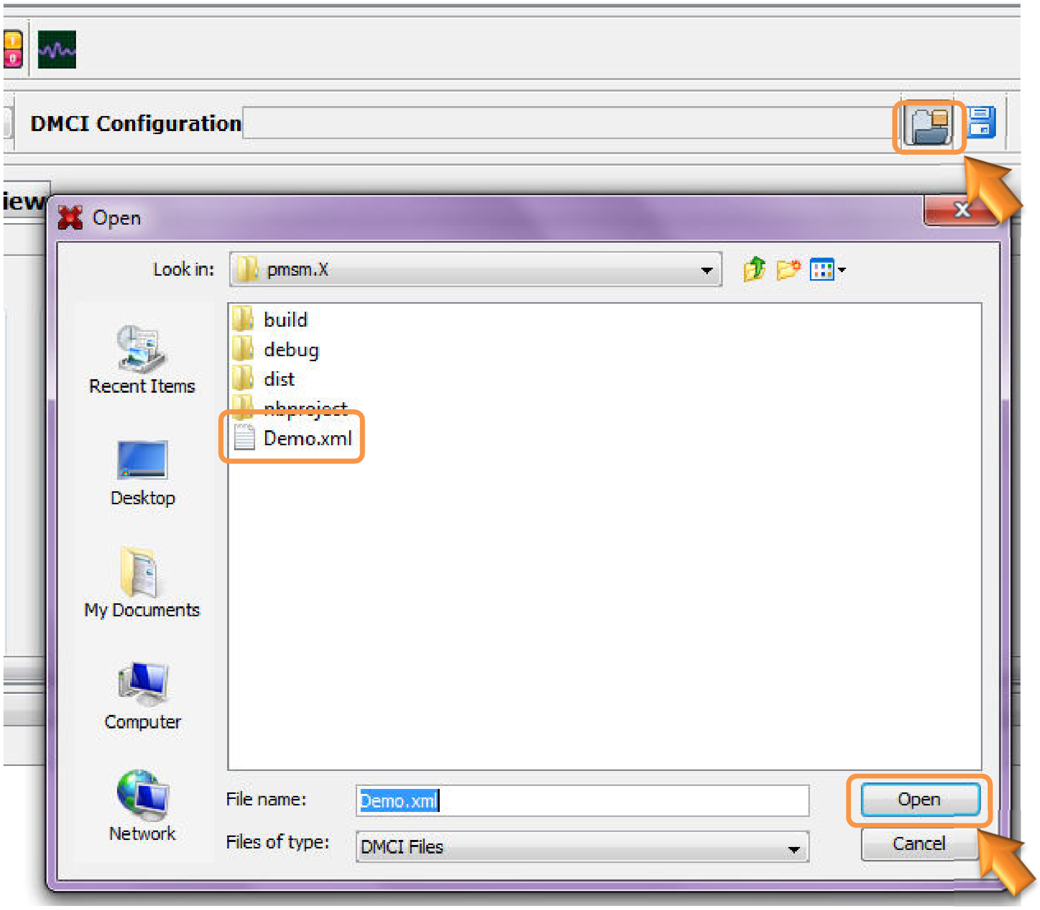


7. From the Select Project drop down menu availble in the DMCI window , select Slave Project '*`pmsm`*'



Select Project: pmsm

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

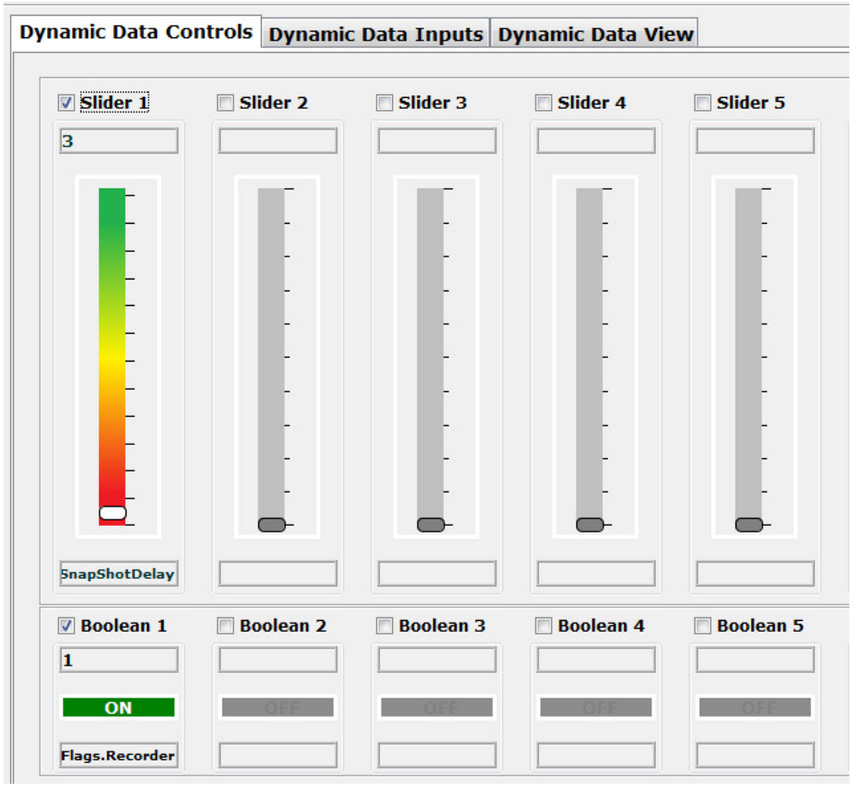


Look in: pmsm.X

File name: Demo.xml

Files of type: DMCI Files

9. The DMCI window appears as follows:

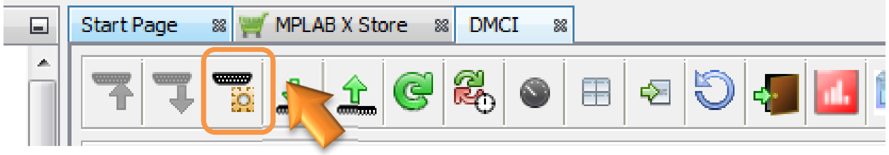


☑ Slider 1: 3 — SnapShotDelay

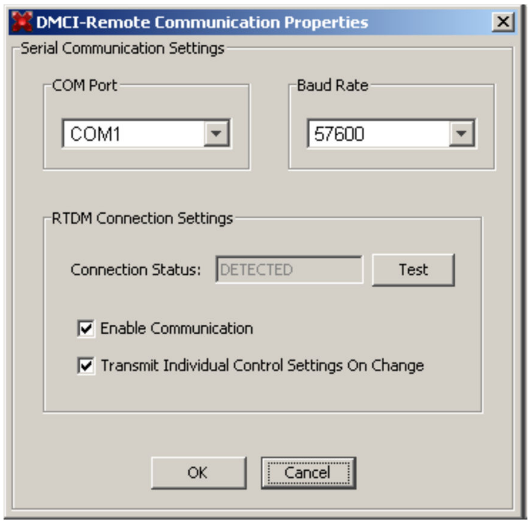
☑ Boolean 1: 1 — ON — Flags.Recorder

Please consult the *“Real-Time Data Monitor User’s Guide”* (DS70567) for additional settings needed for a RDTM connection.

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



11. 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.


DMCI-Remote Communication Properties
Serial Communication Settings
RTDM Connection Settings
Test
OK
Cancel

COM Port: COM1

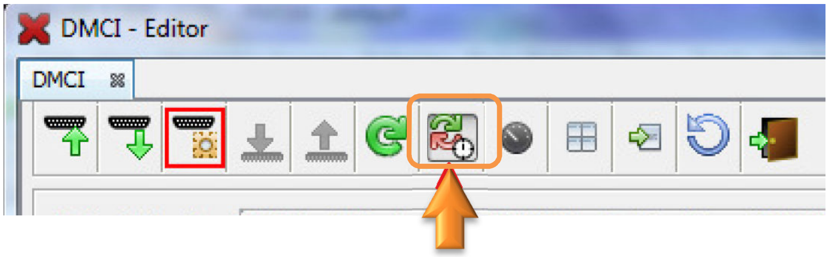
Baud Rate: 57600

Connection Status: DETECTED

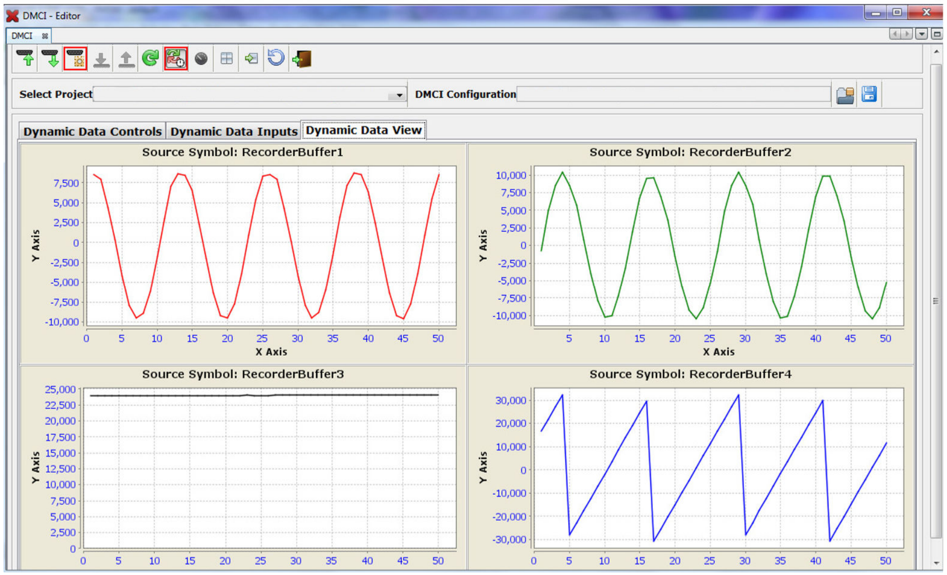
☑ Enable Communication

☑ Transmit Individual Control Settings On Change

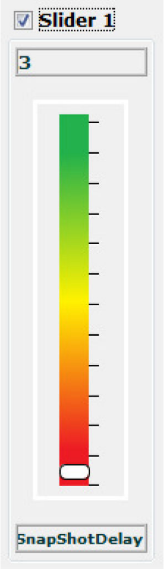
12. Once communication is detected, make sure the “Enable Communication” box is checked and click **OK**.
13. To plot variables in real time, enable “Automated Event Control” by clicking **Automatic Event Execution** icon found on the toolbar.



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


DMCI - Editor
DMCI
Select Project
DMCI Configuration
Dynamic Data Controls
Dynamic Data Inputs
Dynamic Data View
Source Symbol: RecorderBuffer1
Source Symbol: RecorderBuffer2
Source Symbol: RecorderBuffer3
Source Symbol: RecorderBuffer4
Y Axis
X Axis

15. 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.



☑ Slider 1

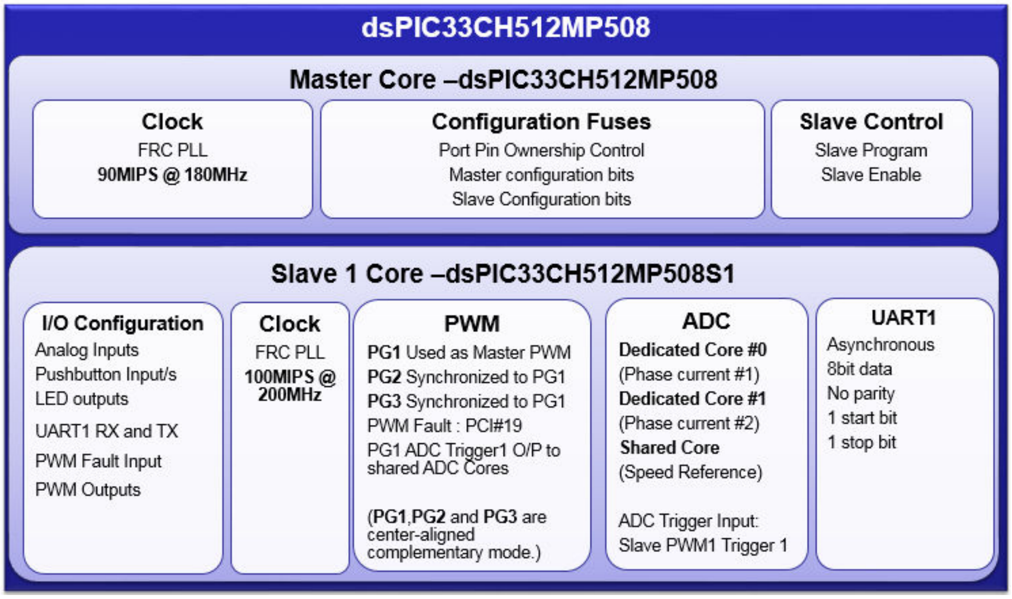
3

SnapShotDelay

6. dsPIC® DSC RESOURCE USAGE SUMMARY

6.1. Device Peripherals Configured and Used in the Firmware:

The following block diagram shows the various peripherals used in the application. The dsPIC33CH512MP508 has two cores namely Master and Slave. The AN1292 motor control application firmware is implemented in the Slave core.



dsPIC33CH512MP508

Master Core –dsPIC33CH512MP508

Clock
FRC PLL
90MIPS @ 180MHz

Configuration Fuses
Port Pin Ownership Control
Master configuration bits
Slave Configuration bits

Slave Control
Slave Program
Slave Enable

Slave 1 Core –dsPIC33CH512MP508S1

I/O Configuration
Analog Inputs
Pushbutton Input/s
LED outputs
UART1 RX and TX
PWM Fault Input
PWM Outputs

Clock
FRC PLL
100MIPS @ 200MHz

PWM
PG1 Used as Master PWM
PG2 Synchronized to PG1
PG3 Synchronized to PG1
PWM Fault : PCI#19
PG1 ADC Trigger1 O/P to shared ADC Cores
(**PG1**,**PG2** and PG3 are center-aligned complementary mode.)

ADC
Dedicated Core #0
(Phase current #1)
Dedicated Core #1
(Phase current #2)
Shared Core
(Speed Reference)
ADC Trigger Input:
Slave PWM1 Trigger 1

UART1
Asynchronous
8bit data
No parity
1 start bit
1 stop bit

Following configurations are done in Master:

- Sets device Configuration bits applicable for the Master and the Slave. Note that the I/O port ownership between the Master and the Slave are decided by the Configuration bits.
- Configure Master Core Oscillator Subsystem to generate clocks needed for Master and its peripherals. In the firmware Master is configured to operate at 90MIPS.
- To program and enable the Slave

Following Configuration are done in Slave:

- Configure Slave Core Oscillator Subsystem to generate clocks needed for operation of the Core and its peripherals. In the firmware, Slave core is configured to operate at 100MIPS.
- Configure I/O ports such as analog inputs (motor phase currents and speed reference), Push button inputs (S2 and S3), LED outputs (D2 and D17), PWM Outputs Fault Input(PCI#19) and UART1 Remappable input/output (U1TX and U1RX)
- Configure PWM Generators PG1, PG2, and PG3 in Centre Aligned (PWM Mode – Complementary (Output mode) configuration as required for three phase motor control. The PWM Generators PG1, PG2, and PG3 are synchronized to each other. The PCI#19 is configured as fault source of PG1, PG2, and PG3. The PG1 ADC Trigger 1 output is enabled to trigger Slave ADC cores for sampling analog input signals at the start of PWM cycle (i.e. middle of S1PWM1H/ S1PWM2H/ S1PWM3H off time).
- Configure all Slave ADC Cores – Dedicated Core #0, Dedicated Core#1 and Shared Core. The Dedicated Core#0 is configured to sense Motor Phase Current #1. The Dedicated Core#1 is configured to sense Motor Phase Current #2. And the shared core is configured to sense speed reference set through potentiometer POT1.ADC trigger source “Slave PWM1 Trigger1” is used as trigger source of all the ADC cores to allow simultaneous sampling of analog inputs. ADC interrupt `_ADCAN17Interrupt ()` is enabled to execute motor control algorithm every PWM cycle.
- UART1 is configured for asynchronous 8bit Data transfer (with No parity ,1 Start and Stop bit). This will be used for data transfer between Host PC and Slave Core if DMCI-RTDM interface is enabled to plot data.

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

The following table summarizes device pins configured and used in the AN1292 motor control application firmware demonstrated using the dsPICDEM™ MCLV-2 Development Board and the dsPIC33CH512MP508 Motor Control Plug-in Module in External Op-amp configuration. Refer “*dsPIC33CH512MP508 Motor Control Plug-in-Module (PIM) Information Sheet(DS50002781)”* for more information.

Functional Description	PIM PIN Number	Device PIN Number	Device Pin Name	Signal Type	Remarks
Motor Control PWMs and Fault Input					
PWM1H1	PIM:94	66	S1RP58/**S1PWM1H/S1RC10**	PWM Output	Controls Hex Bridge MOSFET Q5
PWM1L1	PIM:93	67	S1RP59/**S1PWM1L/S1RC11**	PWM Output	Controls Hex Bridge MOSFET Q6
PWM1H2	PIM:99	63	S1RP52/**S1PWM2H/S1RC4**	PWM Output	Controls Hex Bridge MOSFET Q3
PWM1L2	PIM:98	65	S1RP53/**S1PWM2L/S1RC5**	PWM Output	Controls Hex Bridge MOSFET Q4
PWM1H3	PIM:03	68	S1RP68/**S1PWM3H/S1RD4**	PWM Output	Controls Hex Bridge MOSFET Q1
PWM1L3	PIM:100	69	S1RP67/**S1PWM3L/S1RD3**	PWM Output	Controls Hex Bridge MOSFET Q2
FAULT_MC	PIM:18	49	S1SDO1/**S1PCI19**/**S1RD8**	PWM Input	Connected to Over Current Fault Output
Analog Inputs – Phase Currents, Speed Reference					
POT	PIM:32	36	**S1AN17**/S1PGA1P2/**S1RD11**	Analog Input	Speed Reference Connected to Potentiometer POT1
IMOTOR1	PIM:22	23	S1MCLR3/**S1AN1**/S1CMP2A/S1PGA2P1/S1PGA3P2/**S1RA4**	Analog Input	Connected to Motor Phase Current 1 through External Op-Amp Matrix Board and Jumper JP1
IMOTOR2	PIM:21	21	**S1AN0**/S1CMP1A/S1PGA1P1/**S1RA3**	Analog Input	Connected to Motor Phase Current 2 through External Op-Amp Matrix Board and Jumper JP2
Miscellaneous Signals					
BTN_1	PIM:83	15	S1AN10/S1RP48/**S1RC0**	Digital Input	Connected to Push Button S2
BTN_2	PIM:84	77	**S1RE14**	Digital Input	Connected to Push Button S3
Debug LED1	PIM:60	59	**S1RE11**	Digital Output	Connected to LED D17
Debug LED2	PIM:01	2	**S1RE0**	Digital Output	Connected to LED D2
RX (UART)	PIM:49	47	**S1RP57**/S1ASCL1/S1SDI1/**S1RC9**	UART1 Input	Connected to UART-USB converter to establish serial communication interface between Host PC and Slave Core as needed by DMCI-RTDM.
TX (UART)	PIM:50	46	**S1RP56**/S1ASDA1/S1SCK1/**S1RC8**	UART1 Output	Connected to UART-USB converter to establish serial communication interface between Host PC and Slave Core as needed by DMCI-RTDM.

7. REFERENCES:

For additional information, refer following documents or links.

1. AN1292 Application Note “Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)”
2. dsPICDEM™ MCLV-2 Development Board User’s Guide(DS52080)
3. dsPIC33CH512MP508 Motor Control Plug-in-Module (PIM) Information Sheet(DS50002781)
4. dsPIC33CH512MP508 Family datasheet.
5. Family Reference manuals (FRM) of dsPIC33CH512MP508 family
6. MPLAB® X IDE User’s Guide (DS50002027) or MPLAB® X IDE help
7. 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