AN1292 Demonstration ReadMe for the dsPIC33CDVC256MP506 Motor Control Development Board (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)*" and AN1299"Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM" using a dsPIC33CDVC256MP506 Motor Control Development Board.

The demonstration is configured to run on the dsPIC33CDVC256MP506 Motor Control Development Board in both Internal and External Op Amp configuration with the dsPIC33CDVC256MP506.

2. SUGGESTED DEMONSTRATION REQUIREMENTS

2.1. Motor Control Application Firmware Required for the Demonstration

- AN1292_dsPIC33CDVC256MP506_MC_DEV_BOARD.`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 v5.50 or later
- MPLAB® XC16 Compiler v1.70
- DFP: dsPIC33CK-MP_DFP v1.6.176
- MPLAB® X IDE Plugin: X2C-Scope v1.3.0 or later

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

- dsPIC33CDVC256MP506 Motor Control Development Board (P/N: EV32J63A)
- 24V Power Supply (P/N: AC002013)
- 24V 3-Phase Brushless DC Motor (P/N: AC300020)

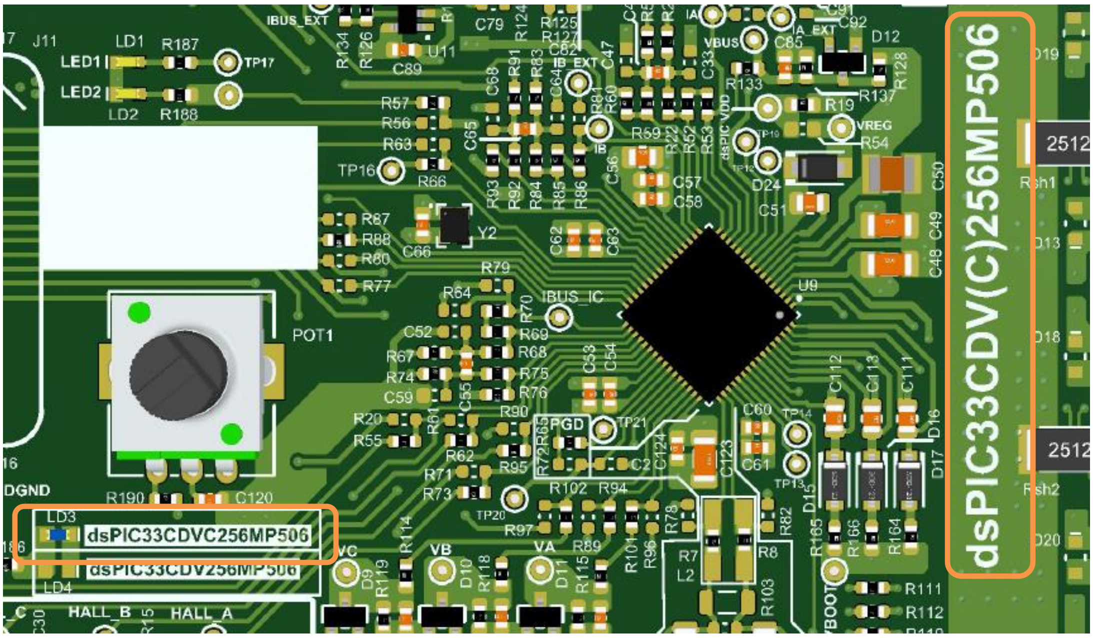
Note:

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

3. HARDWARE SETUP

This section describes the hardware setup required for the demonstration. Motor phase current feedbacks needed by the firmware are amplified by the operational amplifiers.

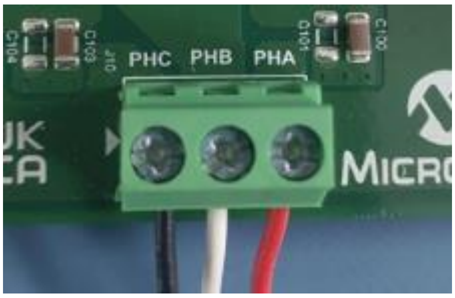
1. The blue color power-on LED (LD3) indicates the device dsPIC33CDVC256MP506 is populated on the development board



2. Motor currents are amplified using external amplifiers ('external op amp configuration'), and amplifiers internal to the dsPIC33CDVC256MP506('internal op amp configuration'). By default, the firmware and Development Board are configured to sample and convert internal amplifier outputs. Below table summarizes the resistors to be populated and removed to convert the Development Board from ‘internal op amp configuration’ to ‘external op amp configuration’ or vice versa.

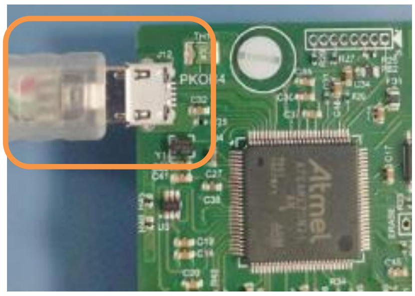
Current Signal	Internal Amplifier Output		External Amplifier Output		Remarks
	Jumper Resistor Settings				
	Populate	Remove	Populate	Remove	
Amplified Phase A Currents **IA or IA_EXT**	R53	R18	R18	R53	In internal amplifier configuration, configure and enable Op Amp 1 (OA1). In external amplifier configuration, ensure internal amplifier Op Amp 1 (OA1) is disabled.
Amplified Phase B Currents **IB or IB_EXT**	R86	R81	R81	R86	In internal amplifier configuration, configure and enable Op Amp 3 (OA3). In external amplifier configuration, ensure internal amplifier Op Amp 3 (OA3) is disabled.
Amplified Phase C Currents **IC_EXT**	Not Implemented		R79	R64 and R70	In external amplifier configuration, ensure internal amplifier Op Amp 2 (OA2) is disabled.
Amplified Bus Currents **IBUS or IBUS_EXT**	R70	R64 and R79	R64	R70 and R79	In internal amplifier configuration, configure and enable Op Amp 2 (OA2). In external amplifier configuration, ensure internal amplifier Op Amp 2 (OA2) is disabled.

3. Connect the three-phase wires from the motor to PHA, PHB, and PHC terminals of connector J10(there is no specific order), provided on the dsPIC33CDVC256MP506 Motor Control Development Board.

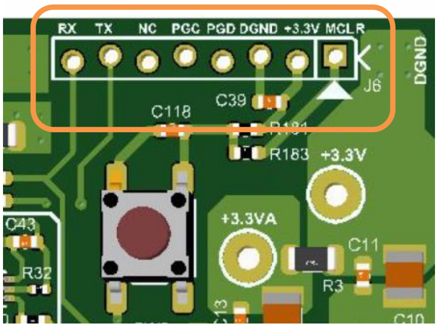


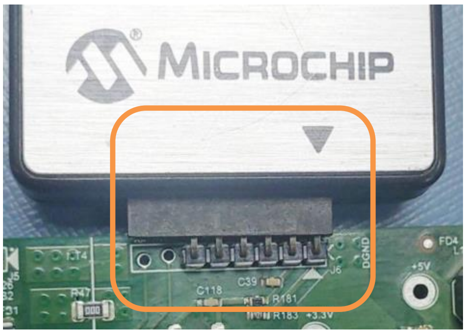
4. Plug in the 24V power supply to connector J1 or J2 provided on the dsPIC33CDVC256MP506 Motor Control Development Board.

5. The board has an onboard programmer ‘PICKIT™ On Board (PKOBv4)”, which can be used for programming or debugging dsPIC33CDVC256MP506 device to control the motor. To use an on-board programmer, connect a micro-USB cable between Host PC and Connector J12 provided on the dsPIC33CDVC256MP506 Motor Control Development Board.



6. Alternatively, the device can also be programmed using the programmer/debugger (MPLAB® PICkit™ 4 In-Circuit Debugger - PG164140) by interfacing it through connector J6 of the dsPIC33CDVC256MP506 Motor Control Development Board as shown below. Ensure that the programmer is oriented correctly before proceeding.





4. SOFTWARE SETUP AND RUN

4.1. Setup: MPLAB X IDE and MPLAB XC16 Compiler

Install MPLAB X IDE and MPLAB XC16 Compiler versions that support the device dsPIC33CDVC256MP506 and PKOBv4. The MPLAB X IDE, MPLAB XC16 Compiler, and X2C-Scope plug-in used for testing the firmware are mentioned in the Motor Control Application Firmware Required for the Demonstration section.

To get help on

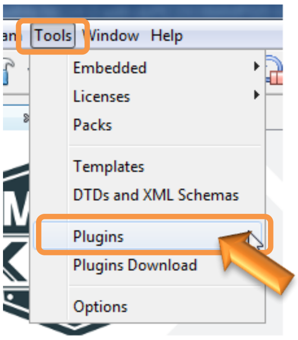
- MPLAB X IDE installation, refer to link
- MPLAB XC16 Compiler installation steps, refer to link

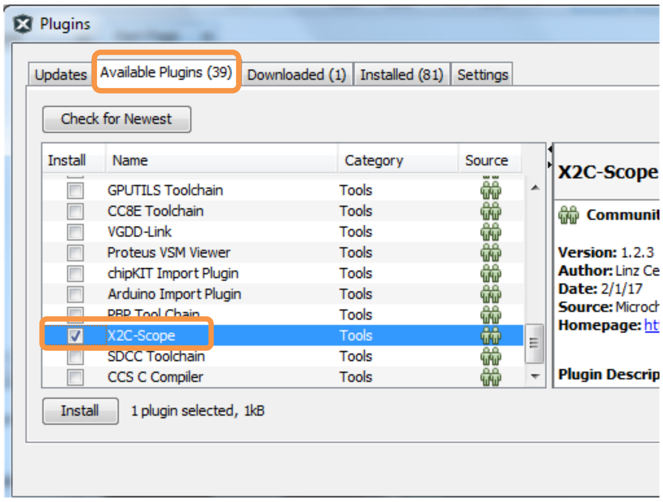
If MPLAB IDE v8 or earlier is already installed on your computer, then run the MPLAB driver switcher (It is installed when MPLAB®X IDE is installed) to switch from MPLAB IDE v8 drivers to MPLAB X IDE drivers. If you have Windows 7 or 8, you must run MPLAB driver switcher in 'Administrator Mode.' To run 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 to MPLAB X IDE help topic *"Before You Begin: Install the USB Device Drivers (For Hardware Tools): USB Driver Installation for Windows Operating Systems."*

4.2. Setup: X2C-Scope

X2C-Scope is an MPLAB X IDE plugin that allows a developer to interact with an application while the application program is running. X2C-Scope enables you to read, write, and plot global variables 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 clicking **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	
□	PBP Tool Chain	Tools	
☑	X2C-Scope	Tools	
□	SDCC Toolchain	Tools	
□	CCS C Compiler	Tools	

X2C-Scope

Communit

Version: 1.2.3
Author: Linz Ce
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 Motor Control Application Firmware Required for the Demonstration section.

This firmware is implemented to work on Microchip's 16-bit Digital signal controller (dsPIC® DSC) dsPIC33CDVC256MP506**.** For more information, see the *dsPIC33CDVC256MP506 Family datasheet (DS70005484)*.

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

This Motor Control Demo Application configures and uses peripherals like PWM, ADC, UART, etc., 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 to Microchip Application note AN1292 “Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor(PMSM) Using a PLL Estimator and Field Weakening(FW)” available on the website.

Note:

The project may not build correctly in Windows OS if Maximum path length of any source file in the project is more than 260 characters. In case absolute path is exceeding or nearing maximum length, do any (or both) of the following:

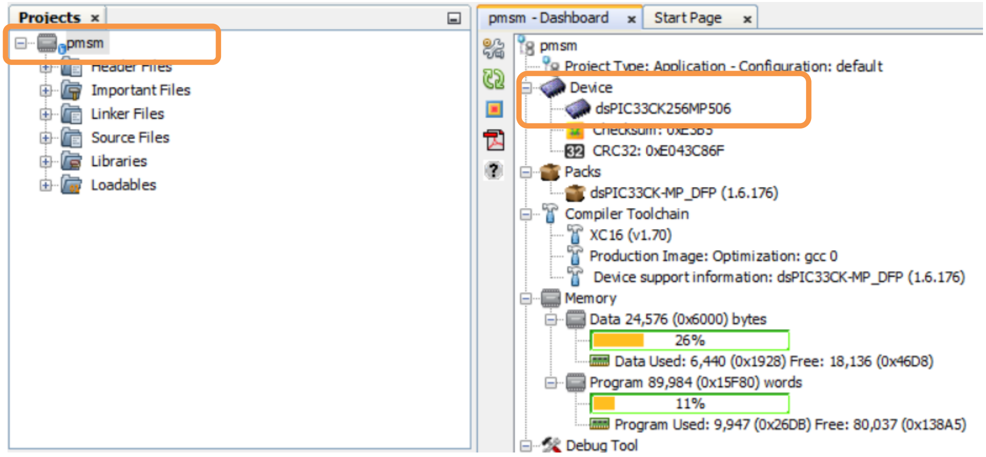
- Shorten the name of the directory containing the firmware used in this demonstration. In this case, rename directory `AN1292_dsPIC33CDVC256MP506_MC_DEV_BOARD` to more appropriate shorter name. In case you renamed the directory, consider the new name while reading instructions provided in the upcoming 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

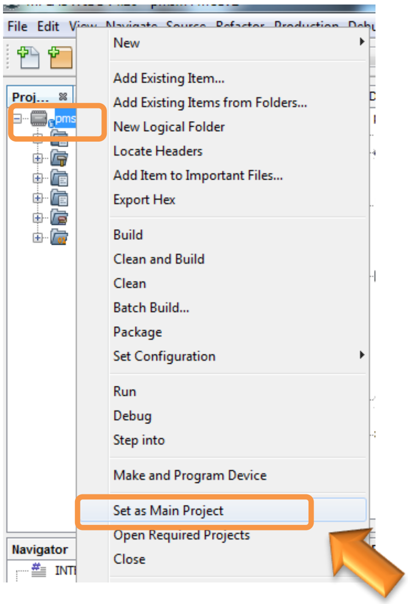
Follow the below instructions step by step to set up and run the motor control demo application:

1. Start MPLAB X IDE and open `(File>Open Project)` the project *`pmsm.X`* `(..\AN1292_dsPIC33CDVC256MP506_MC_DEV_BOARD\pmsm.X)` with device selection *dsPIC33CK256MP506*.



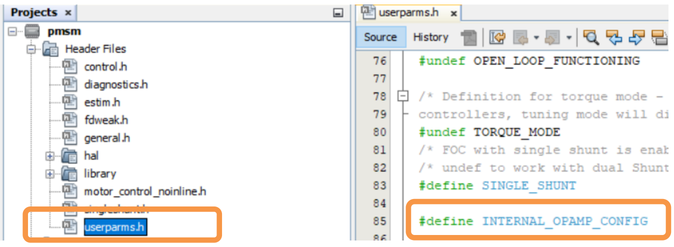
pmsm
Project Type: Application - Configuration: default
Device
dsPIC33CK256MP506
Checksum: 0xE3B5
CRC32: 0xE043C86F
Packs
dsPIC33CK-MP_DFP (1.6.176)
Compiler Toolchain
XC16 (v1.70)
Production Image: Optimization: gcc 0
Device support information: dsPIC33CK-MP_DFP (1.6.176)
Memory
Data 24,576 (0x6000) bytes
26%
Data Used: 6,440 (0x1928) Free: 18,136 (0x46D8)
Program 89,984 (0x15F80) words
11%
Program Used: 9,947 (0x26DB) Free: 80,037 (0x138A5)
Debug Tool

2. Set the project *`pmsm.X`* as the 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**.



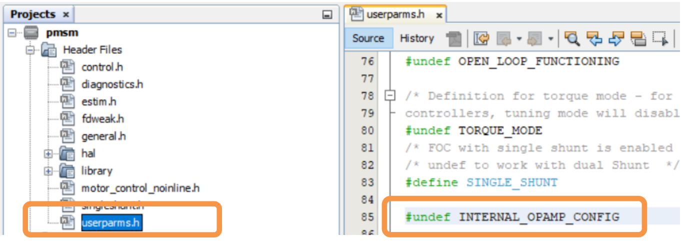
3. Open `userparams.h` (under `pmsm.X -> headerfiles`) in the project `pmsm.X`

   - Ensure that `TUNING`, `OPEN_LOOP_FUNCTIONING`, and `TORQUE_MODE` are not defined.
   - When internal amplifiers are used for current amplification (referred to as 'internal op amp configuration'), then define `INTERNAL_OPAMP_CONFIG`.



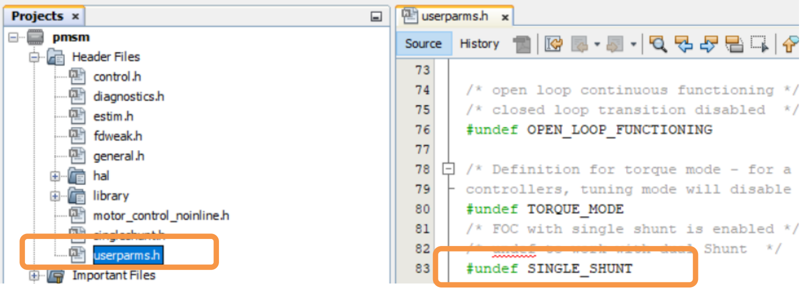
```
#undef OPEN_LOOP_FUNCTIONING

/* Definition for torque mode -
controllers, tuning mode will di
#undef TORQUE_MODE
/* FOC with single shunt is enak
/* undef to work with dual Shunt
#define SINGLE_SHUNT

#define INTERNAL_OPAMP_CONFIG
```

Otherwise, if external amplifiers are used for current amplification (referred to as 'external op amp configuration'), then undefine the macro `INTERNAL_OPAMP_CONFIG` in `userparams.h.`



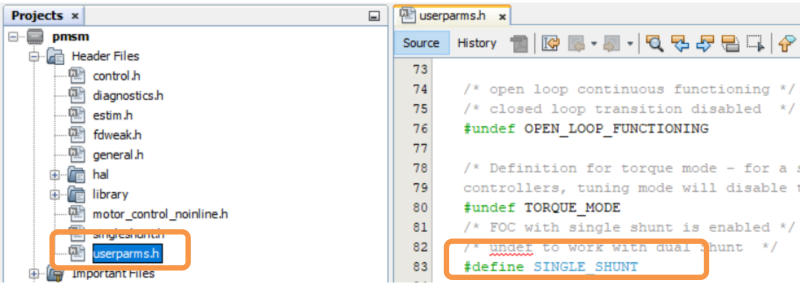
```
#undef OPEN_LOOP_FUNCTIONING

/* Definition for torque mode - for
controllers, tuning mode will disabl
#undef TORQUE_MODE
/* FOC with single shunt is enabled
/* undef to work with dual Shunt  */
#define SINGLE_SHUNT

#undef INTERNAL_OPAMP_CONFIG
```

4. Open `userparams.h` (under `pmsm.X -> headerfiles`) in the project pmsm.X and ensure `SINGLE_SHUNT` is undefined if dual shunt configuration is used for phase current measurement and demonstration. By default, firmware is enabled to operate in a dual shunt configuration.



```

/* open loop continuous functioning */
/* closed loop transition disabled  */
#undef OPEN_LOOP_FUNCTIONING

/* Definition for torque mode - for a
controllers, tuning mode will disable
#undef TORQUE_MODE
/* FOC with single shunt is enabled */
/* undef to work with dual Shunt  */
#undef SINGLE_SHUNT
```

5. For a demonstration of FOC using a single shunt reconstruction algorithm (i.e., reconstruction of phase currents from DC Bus Current), enable it by defining the macro `SINGLE_SHUNT` in `userparams.h` (under `pmsm.X -> headerfiles`) part of the project `pmsm.X.`

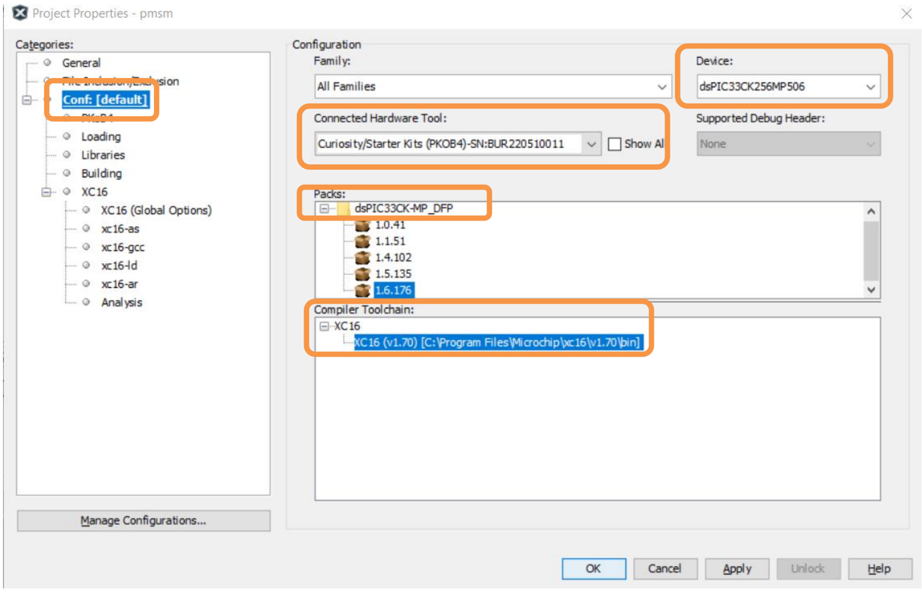


```

/* open loop continuous functioning */
/* closed loop transition disabled  */
#undef OPEN_LOOP_FUNCTIONING

/* Definition for torque mode - for a
controllers, tuning mode will disable
#undef TORQUE_MODE
/* FOC with single shunt is enabled */
/* undef to work with dual Shunt  */
#define SINGLE_SHUNT
```

6. Right-click on the project *pmsm.X* and select “Properties” to open its Project Properties Dialog. Click the “**Conf: [default]”** category to reveal the general project configuration information. The development tools used for testing the firmware are listed in section 2.2. Software Tools Used for Testing the firmware.

   In the *‘**Conf-default**’* category window:

   - Select the specific Compiler Toolchain from the available list of compilers. In this case, “XC16(v1.70)” is selected.
   - Select the Hardware Tool to be used for programming and debugging.
   - Select the specific Device Family Pack (DFP) from the available list of Packs. In this case, “dsPIC33CK-MP_DFP v1.6.176” is selected.
   - After selecting Hardware Tool and Compiler Toolchain, click the button **Apply**



Family: All Families

Device: dsPIC33CK256MP506

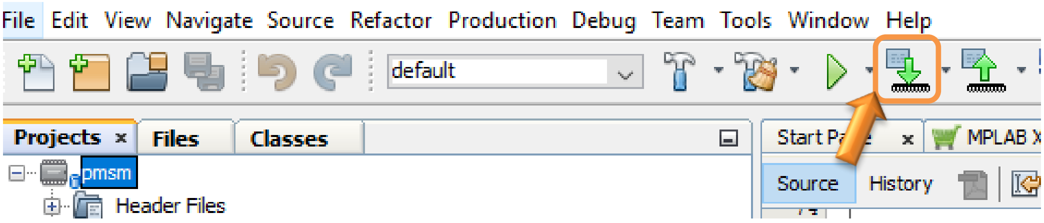
Connected Hardware Tool: Curiosity/Starter Kits (PKOB4)-SN:BUR220510011 □ Show All

Supported Debug Header: None

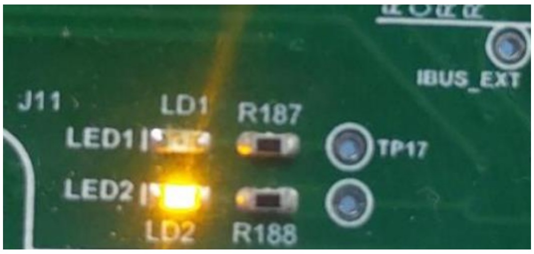
Packs:
dsPIC33CK-MP_DFP
1.0.41
1.1.51
1.4.102
1.5.135
1.6.176

Compiler Toolchain:
XC16
XC16 (v1.70) [C:\Program Files\Microchip\xc16\v1.70\bin]

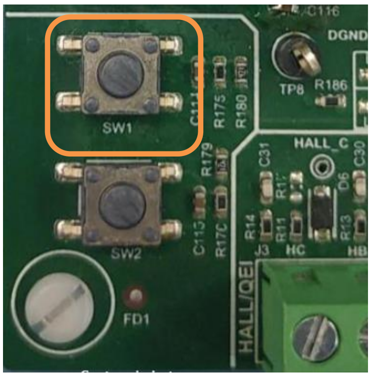
7. To build the project (in this case, *pmsm. X*) and program the device dsPIC33CDVC256MP506, click “**Make and Program Device Main project**” on the toolbar.



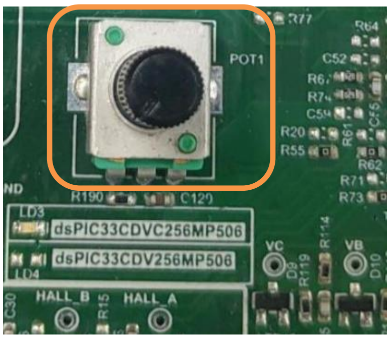
8. If the device is successfully programmed, **LD2 (‘LED2)** will be turned ON, indicating that the dsPIC® DSC is enabled.



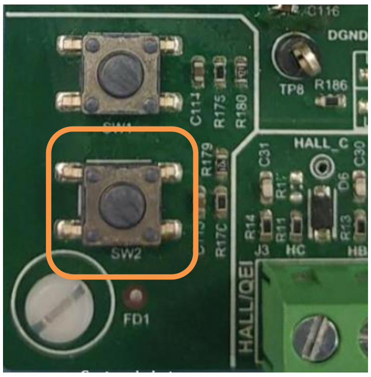
9. Run or Stop the motor by pressing the push button **SW1**.



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



11. To enter the extended speed range (`NOMINAL_SPEED_RPM` to `MAXIMUM_SPEED_RPM`), press the push button **SW2**. Press the push button **SW2** again to revert the speed of the motor to its normal speed (`END_SPEED_RPM` to `NOMINAL_SPEED_RPM`) range.



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

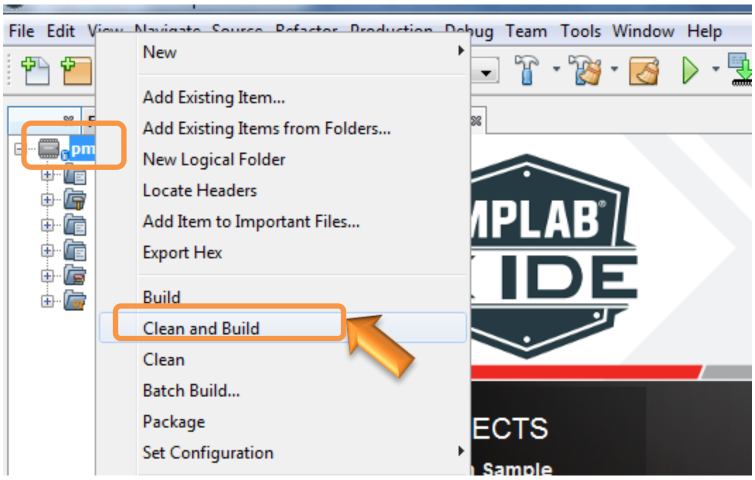
Note:

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.*

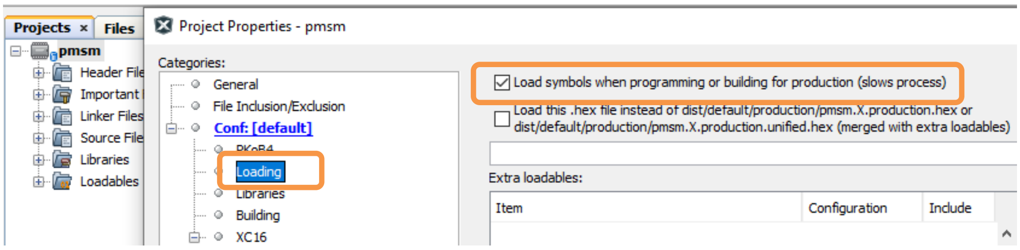
5.3. Data visualization through X2C-Scope Plug-in of MPLABX IDE

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

1. Ensure X2C-Scope Plug-in is installed. For more 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, connect a micro-USB cable between the Host PC and the micro-USB connector provided on the dsPIC33CDVC256MP506 Motor Control Development Board. This interface is used for programming as well.
3. Ensure the application is configured and running as described under Section Basic Demonstration by following steps 1 through 10.
4. Build the project pmsm.X. To do that, right-click on the project pmsm.X and select “Clean and Build.”



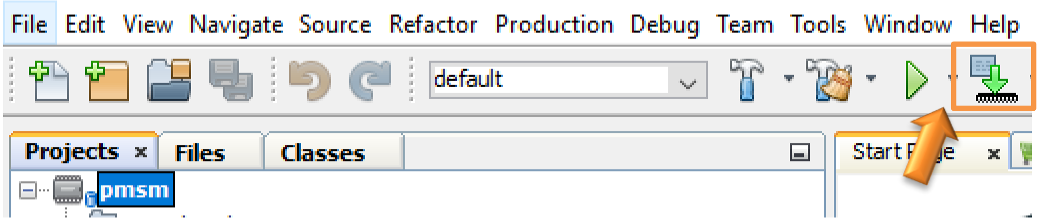
5. Please ensure that the checkbox “Load symbols when programming or building for production (slows process)” is checked, which is under the “Loading” category of the Project Properties window.



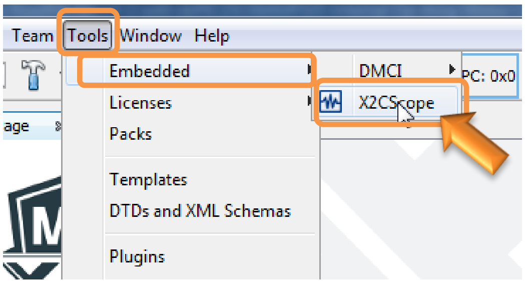
☑ Load symbols when programming or building for production (slows process)

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

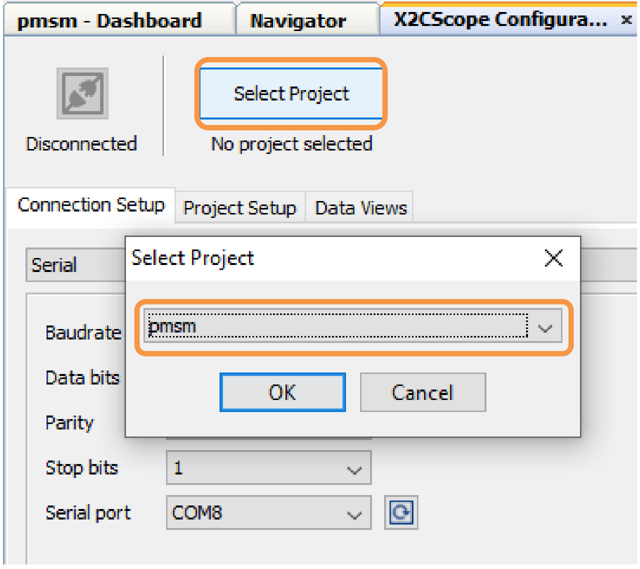
6. To build the project (in this case, *pmsm.X*) and program the device dsPIC33CDVC256MP506, click “**Make and Program Device Main project**” on the toolbar.


File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
default
Projects × Files Classes
Start Page ×
pmsm

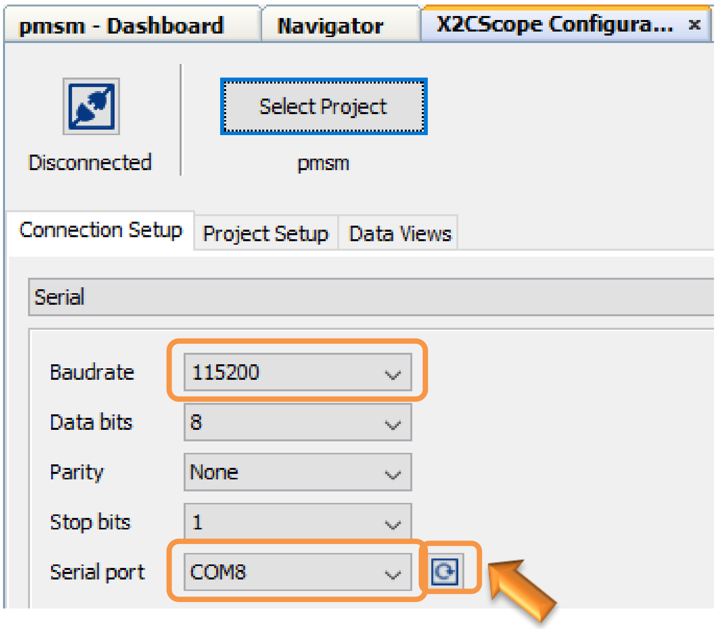
7. Open the X2C-Scope window by selecting `Tools>Embedded>X2CScope`.



8. In the X2C-Scope Configuration window, using the "Select Project" menu, select the **'pmsm'** project as shown.

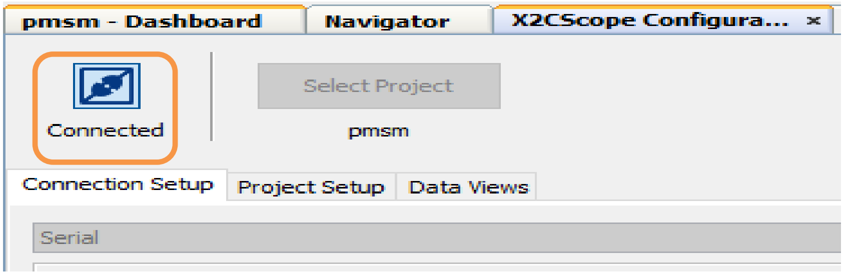


9. Remote Communication needs to be set up, as shown in the following figure. Ensure the communication baud rate is set to 115200 as configured in the application firmware. The COM port used depends on the system settings. `The refresh button` lists the available COM Ports. Select the COM Port as per the connection.

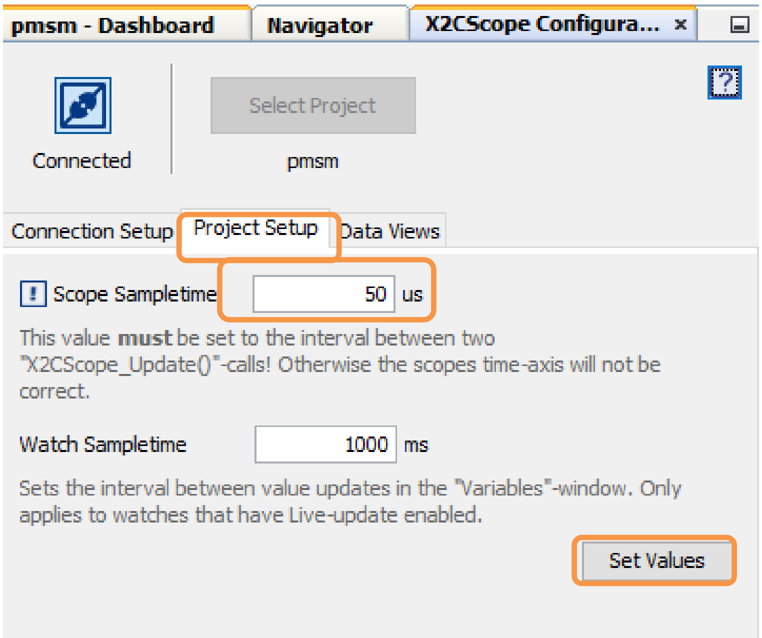


Baudrate 115200
Data bits 8
Parity None
Stop bits 1
Serial port COM8

10. Once the COM port is detected, click on “Disconnected” and turn to “Connected” to establish serial communication between the Host PC and the board.



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



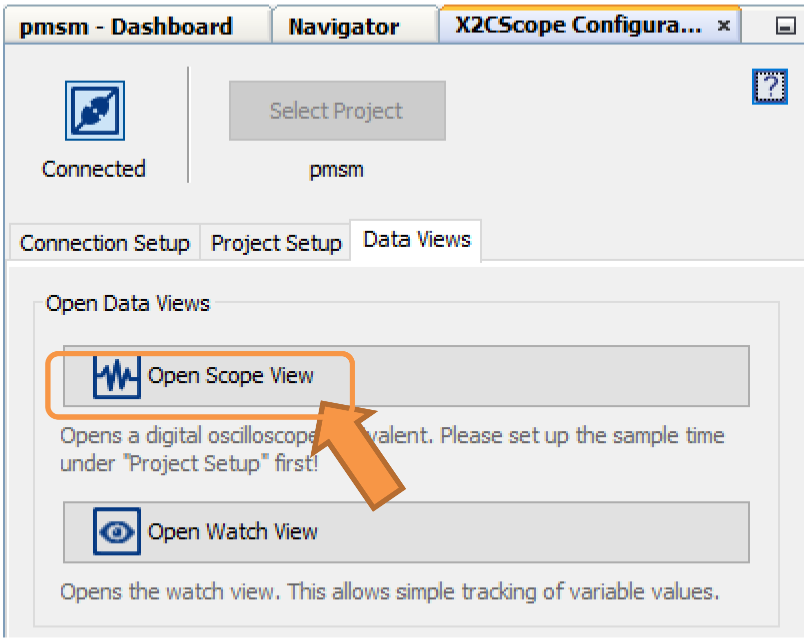
Scope Sampletime 50 us

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. Click on ‘Open Scope View’ (under sub-window “Data Views”); this opens ‘Scope Window.’

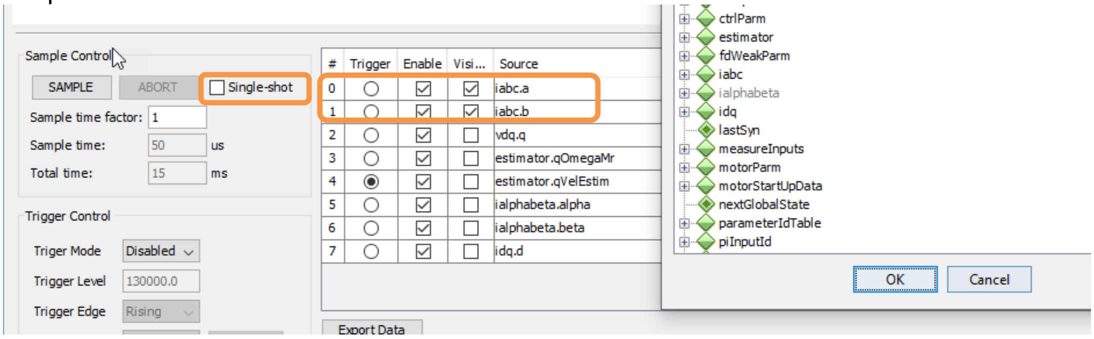


Opens a digital oscilloscope valent. Please set up the sample time under "Project Setup" first!

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

13. In this window, select the variables that need to be watched. To do this, click on the source against each channel, a window Select Variables opens on the screen. From the available list, the required variable can be chosen. Ensure checkboxes 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 decides the time difference between any two consecutive data points on the plot.



Sample Control

SAMPLE ABORT ☐ Single-shot

Sample time factor: 1

Sample time: 50 us

Total time: 15 ms

Trigger Control

Triger Mode: Disabled

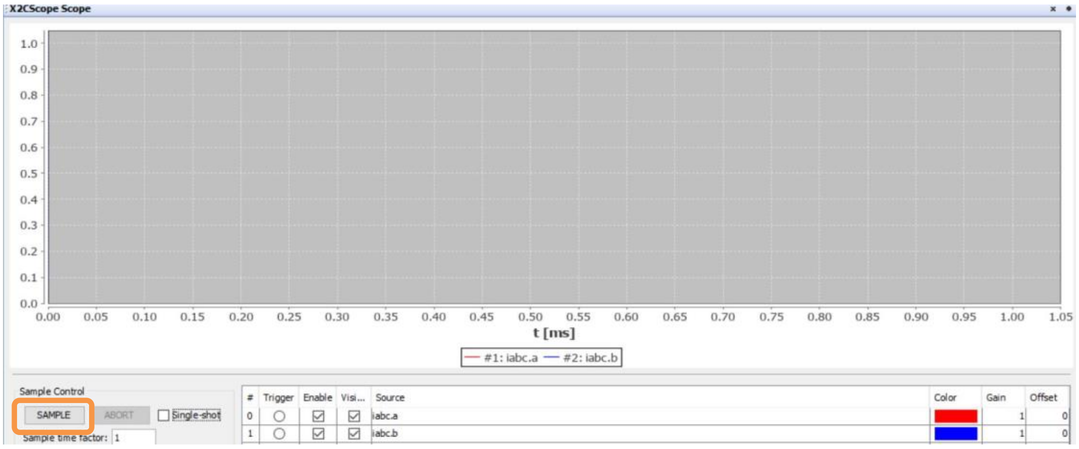
Trigger Level: 130000.0

Trigger Edge: Rising

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

Export Data

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



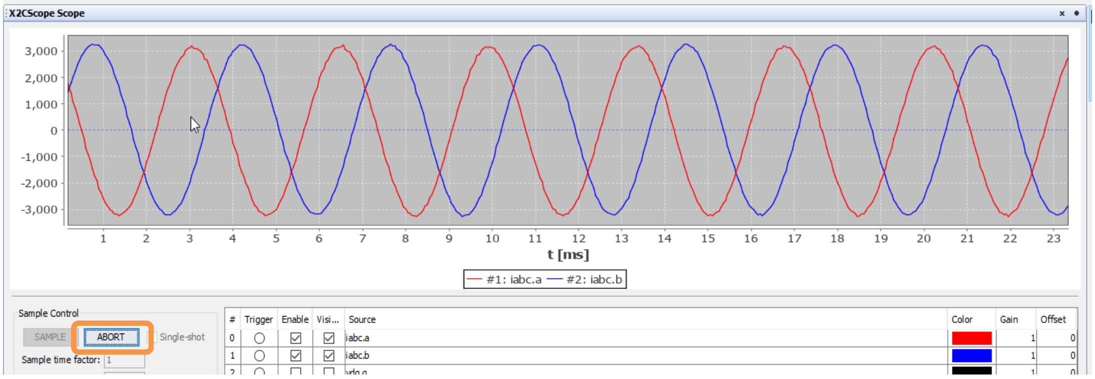
Sample Control

SAMPLE ABORT ☐ Single-shot

Sample time factor: 1

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

15. Click on ABORT to stop.



Sample Control

SAMPLE ABORT Single-shot

Sample time factor: 1

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

6. REFERENCES:

For additional information, refer following documents or links.

1. AN1292 Application Note “Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)”
2. AN1299 Application Note “Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM”
3. dsPIC33CDVC256MP506 Motor Control Development Board User’s Guide (DS50003275)
4. dsPIC33CDVC256MP506 Family datasheet (DS70005484)
5. MPLAB® X IDE User’s Guide (DS50002027) or MPLAB® X IDE help
6. MPLAB® X IDE installation
7. MPLAB® XC16 Compiler installation