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.

1. Suggested Demonstration Requirements
   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**.**

* 1. Software Tools Used for Testing the firmware
* MPLAB® X IDE v6.00 or later
* MPLAB® XC16 Compiler v2.00
* DFP: dsPIC33CD-MP\_DFP v1.0.1
* MPLAB® X IDE Plugin: X2C-Scope v1.3.3 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.

* 1. Hardware Tools Required for the Demonstration
* dsPIC33CDVC256MP506 Motor Control Development Board (P/N: EV32J63A)
* 24V Power Supply (P/N: [AC002013](https://www.microchipdirect.com/dev-tools/AC002013?productLoaded=true&allDevTools=true))
* 24V 3-Phase Brushless DC Motor (P/N: [AC300020](https://www.microchipdirect.com/dev-tools/AC300020?productLoaded=true&allDevTools=true))

**Note:**

All items listed under this section Hardware Tools Required for the Demonstration are available at [microchip DIRECT](http://www.microchipdirect.com/).

1. 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

A picture containing text, electronics, circuit

Description automatically generated

1. 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.

Table, calendar

Description automatically generated

1. 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.

A picture containing text, circuit

Description automatically generated

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

A close-up of a circuit board

Description automatically generated with low confidence

1. 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.

A picture containing text, electronics, circuit

Description automatically generated

1. Alternatively, the device can also be programmed using the programmer/debugger (MPLAB® PICkit™ 4 In-Circuit Debugger - [PG164140](https://www.microchipdirect.com/dev-tools/PG164140?productLoaded=true&allDevTools=true)) 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.

|  |  |
| --- | --- |
|  |  |

1. Software Setup and Run
   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](#RTF32343432313a204865616469) section.

To get help on

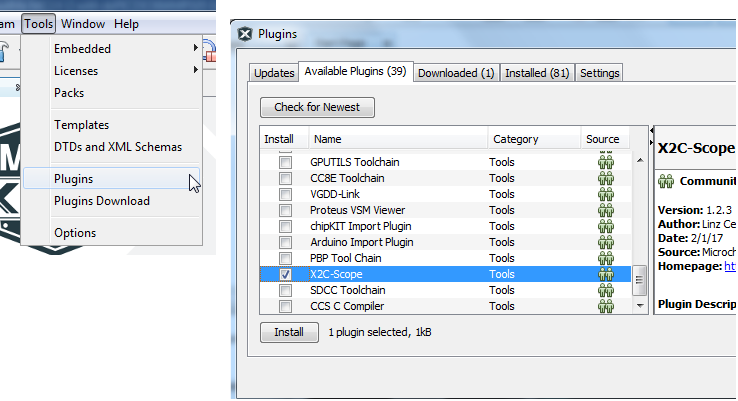
* MPLAB X IDE installation, refer [to link](http://microchipdeveloper.com/mplabx:installation)
* MPLAB XC16 Compiler installation steps, refer [to link](http://microchipdeveloper.com/xc16:installation)

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.”*

* 1. 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.



1. Basic Demonstration
   1. Firmware Description

The firmware version required for the demonstration is mentioned under the [Motor Control Application Firmware Required for the Demonstration](#RTF32343432313a204865616469) 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](http://www.microchip.com/).

**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 directoryAN1292\_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”.*

* 1. Basic Demonstration

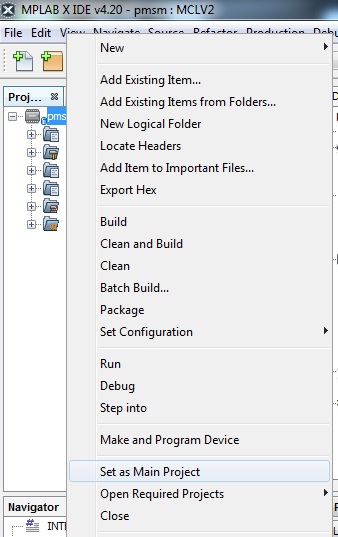
Follow the below instructions step by step to set up and run the motor control demo application:

1. Start MPLAB X IDE and open (File>Open Project) the project *pmsm.X*(..\AN1292\_dsPIC33CDVC256MP506\_MC\_DEV\_BOARD\pmsm.X) with device selection *dsPIC33CDVC256MP506*.

Graphical user interface, text, application

Description automatically generated

1. 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**.



1. Open **userparms.h** (under **pmsm.X > Header Files**) in the project **pmsm.X**.

* Text

  Description automatically generatedEnsure that the macros **TUNING**, **OPEN\_LOOP\_FUNCTIONING**, **TORQUE\_MODE,** and **SINGLE\_SHUNT** is not defined in the header file **userparms.h.**
* When internal amplifiers are used for current amplification (referred to as **internal op-amp configuration**), **define** the macro **INTERNAL\_OPAMP\_CONFIG in userparms.h.**
* Otherwise, if external amplifiers are used for current amplification (referred to as **external op-amp configuration**), **undefine** the macro **INTERNAL\_OPAMP\_CONFIG** in the header file **userparms.h**.

**Note:**

The motor phase currents can be reconstructed from the DC Bus current by appropriately sampling it during the PWM switching period, called a single-shunt reconstruction algorithm. The firmware can be configured to demonstrate **the single shunt reconstruction algorith**m by defining the macro **SINGLE\_SHUNT** in the header file **userparms.h**

For additional information, refer to Microchip application note **AN1299, “Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM.”**

By default, the firmware uses phase currents measured across the phase shunt resistors on two of the half-bridges of the three-phase inverter **(‘dual shunt configuration’**) to implement FOC.

1. 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.](#software)

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

* Ensure the selected **Device** is **dsPIC33CDVC256MP506***.*
* Select the **Connected 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, **dsPIC33CD-MP\_DFP 1.0.1** is selected.
* Select the specific **Compiler Toolchain**from the available list of **XC16** compilers.

In this case, **XC16(v2.00)** is selected.

* After selecting Hardware Tool and Compiler Toolchain, Device Pack, click the button **Apply**

Please ensure that the selected MPLAB® XC16 Compiler and Device Pack support the device configured in the firmware

Graphical user interface, text, application

Description automatically generated

1. A screenshot of a social media post

   Description automatically generatedEnsure that the checkbox **Load symbols when programming or building for production (slows process)** is checked under the **Loading**category of the **Project Properties** window.
2. To build the project (in this case, *pmsm. X*) and program the device dsPIC33CDVC256MP506, click “**Make and Program Device Main project**” on the toolbar.

A screenshot of a cell phone

Description automatically generated

1. If the device is successfully programmed, **LD1 (‘LED1)** will be turned ON, indicating that the dsPIC® DSC is enabled.

A picture containing text, electronics, circuit

Description automatically generated

1. Run or Stop the motor by pressing the push button **SW1**. The motor should start spinning smoothly in one direction in the nominal speed range. Ensure that the motor is spinning smoothly without any vibration. The LED **LD2 (LED2)** is turned **ON** to show the button is pressed to start the motor.

A close-up of a circuit board

Description automatically generated with medium confidence

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

A picture containing text, electronics

Description automatically generated

1. 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.

A close-up of a circuit board

Description automatically generated with medium confidence

1. 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.*

* 1. 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](#Basic_Demo) by following steps 1 through 10.
4. Graphical user interface, application

   Description automatically generatedBuild the project pmsm.X. To do that, right-click on the project pmsm.X and select “Clean and Build.”
5. A screenshot of a social media post

   Description automatically generatedPlease 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.
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.

A screenshot of a cell phone

Description automatically generated

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

Graphical user interface, application

Description automatically generated

1. In the X2C-Scope Configuration window, using the “Select Project” menu, select the **‘pmsm’** project as shown.

Graphical user interface

Description automatically generated

1. 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.

A screenshot of a cell phone

Description automatically generated

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

A screenshot of a cell phone

Description automatically generated

1. 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.

A screenshot of a social media post

Description automatically generated

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

A screenshot of a cell phone

Description automatically generated

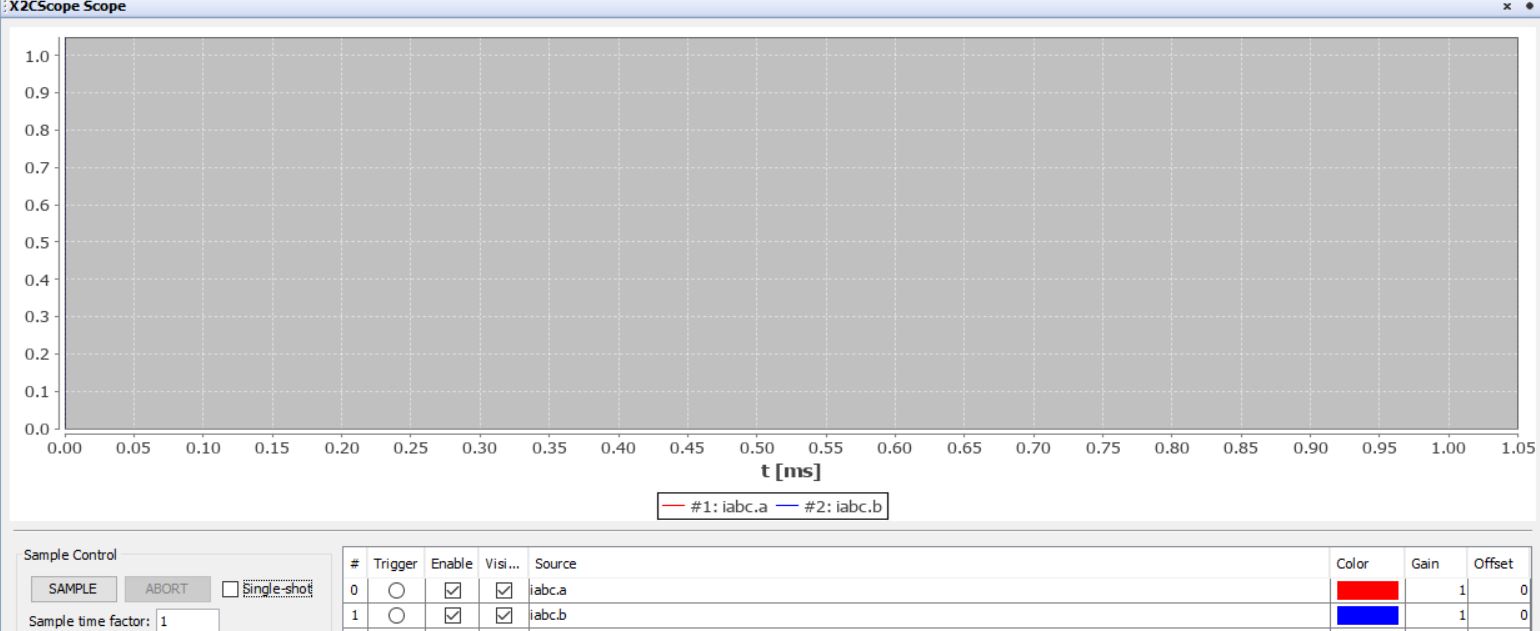
1. In this window, select the variables that need to be watched. To do this, click on the source against each channel, a window Select Variables opens on the screen. From the available list, the required variable can be chosen. Ensure checkboxes Enable & Visible are checked for the variables to be plotted.

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.

Graphical user interface, application, table

Description automatically generated

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



1. Click on ABORT to stop.

Graphical user interface, application

Description automatically generated

1. 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)](https://www.microchip.com/en-us/application-notes/an1292)”
2. AN1299 Application Note “[Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM](https://www.microchip.com/en-us/application-notes/an1299)”
3. dsPIC33CDVC256MP506 and dsPIC33CDV256MP506 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](http://microchipdeveloper.com/mplabx:installation)
7. [MPLAB® XC16 Compiler installation](http://microchipdeveloper.com/xc16:installation)