Sensorless FOC using PLL Estimator and Single-Shunt Current Reconstruction Algorithm for PMSM with Fan Load: Automotive Cooling Fan Reference Design

1. INTRODUCTION

This document describes the setup requirements for driving a Permanent Magnet Synchronous Motor (PMSM) using Sensorless Field Oriented Control (FOC) and PLL Estimator Algorithm for a Fan Load on the hardware platform Automotive Cooling Fan Reference Design.

For details about PLL estimator, 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)”

For details about Single-Shunt Current Reconstruction algorithm, refer to Microchip application note AN1299 "Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM".

Enhance your embedded applications with Microchip's high-performance dsPIC® Digital Signal Controllers (DSCs). Visit our Motor Control and Drive page to stay updated on the latest motor control solutions from Microchip.

2. SUGGESTED DEMONSTRATION REQUIREMENTS

2.1 Motor Control Application Firmware Required for the Demonstration

To clone or download this application firmware on GitHub,

- Navigate to the main page of this repository and
- On the tab **<> Code**, above the list of files in the right-hand corner, click Code, then from the menu, click **Download ZIP** or copy the repository URL to **clone.**

> **Note:**
> In this document, hereinafter this firmware package is referred as **firmware.**

2.2 Software Tools Used for Testing the firmware

- MPLAB® X IDE **v6.25**
- Device Family Pack (DFP): **dsPIC33CD-MC_DFP v1.2.127**
- PICkit™ Tool Pack : **PICkit4_TP v2.8.2226**
- MPLAB® XC-DSC Compiler **v3.21**
- MPLAB® X IDE Plugin: **X2C-Scope v1.7.0**
- LIN Serial Analyzer Application: **v3.0.0**

> **Note:**
> The software used for testing the firmware prior to release is listed above. It is recommended to use

> these or later versions of the tool for building the firmware. All previous versions of Device Family Packs (DFP) and Tool Packs can be downloaded from Microchip Packs Repository.

2.3 Hardware Tools Required for the Demonstration

- Automotive Cooling Fan Reference Design
- MPLAB® PICkit™ 4 In-Circuit Debugger (PG164140) or MPLAB® PICkit™ 5 In-Circuit Debugger (PG164150)
- MCP2200 Isolated USB to UART Board (ADM00276)
- Microchip LIN Serial Analyzer (APGDT001)
- 12V Power Supply

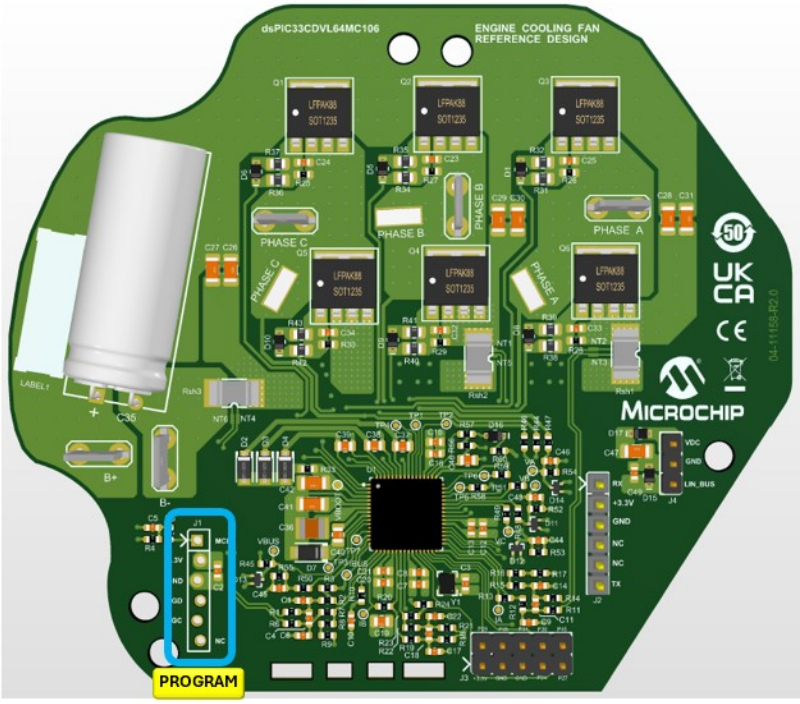
> **Note:**
> The programmer/debugger and communication interface boards for the demonstration are available at microchip DIRECT

3. HARDWARE SETUP

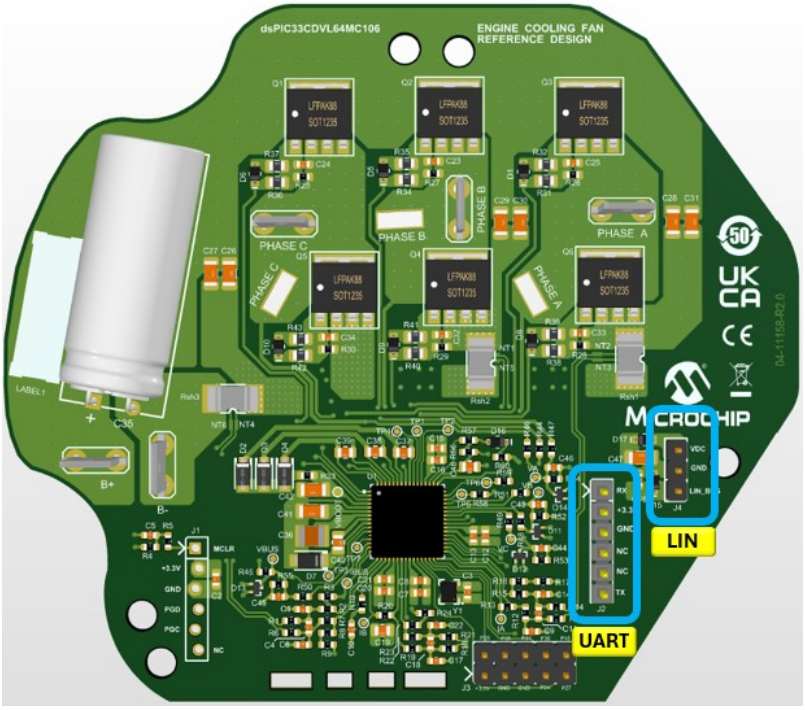
This section describes the hardware setup required for the demonstration.

> **Note:**
> In this document, hereinafter the Automotive Cooling Fan Reference Design is referred as **board**.

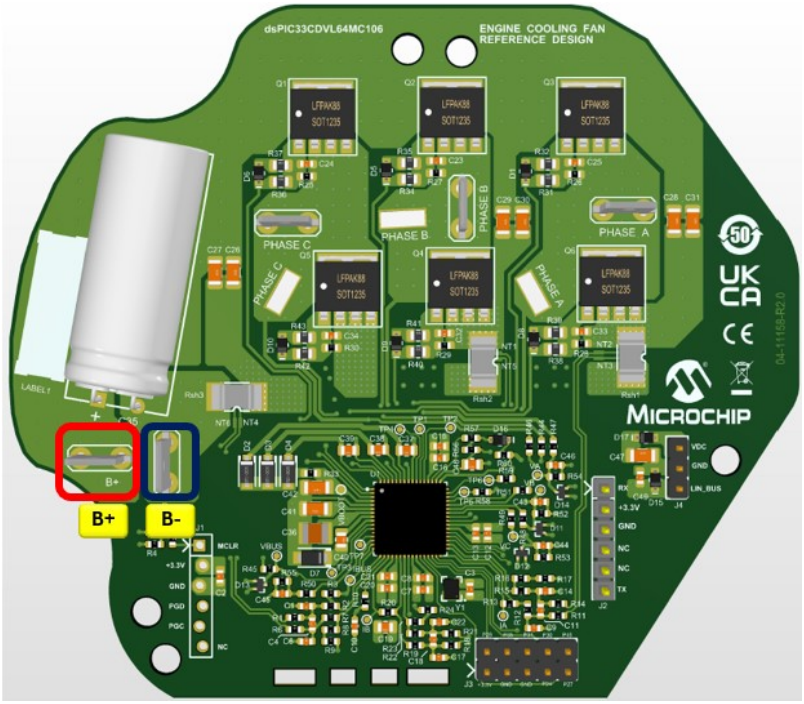
1. Interface the **MPLAB® PICkit™ 4 In-Circuit Debugger** onto the board through **connector J1**. Make sure the programmer/debugger is orineted correctly.



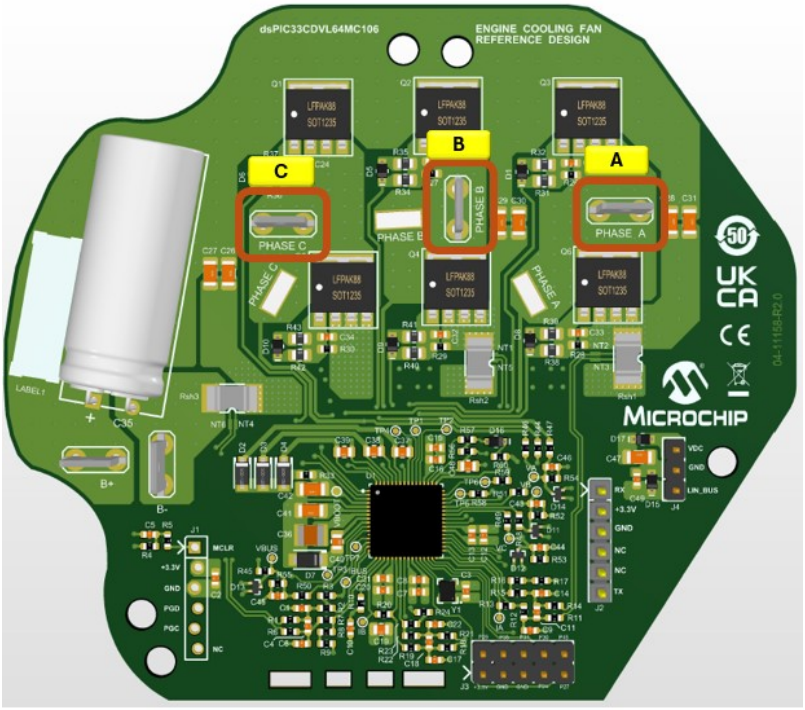
2. Connect the **MCP2200 Isolated USB-UART Board** to the **connector J2** and the **LIN Serial Analyzer** to the **connector J4**.



3. Connect the +12V DC power supply on the terminal **B+** and **B-** on the board. The polarity must be ensured while connecting.



4. Motor phases can be interafaced to the terminals **PHASE A**, **PHASE B** and **PHASE C**. The phase sequence needs to be verified with the direction of rotation of fan load.



4. SOFTWARE SETUP AND RUN

4.1 Setup: MPLAB X IDE and MPLAB XC-DSC Compiler

Install **MPLAB X IDE** and **MPLAB XC-DSC Compiler** versions that support the device **dsPIC33CDVL64MC106** and **PICkit4**. The MPLAB X IDE, MPLAB XC-DSC 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 link
- MPLAB XC-DSC Compiler installation steps, refer link

If MPLAB IDE v8 or earlier is already installed on your computer, then run the MPLAB driver switcher (Installed when MPLAB® X IDE is installed) to switch from MPLAB IDE v8 drivers to MPLAB X IDE drivers. If you have Windows 8 or 10, you must run the 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 more details, refer to the 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 developers to interact with an application while it runs. 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-Scope, the plugin must be installed. To set up and use X2C-Scope, refer to the instructions provided on the web page.

5. BASIC DEMONSTRATION

5.1 Firmware Description

The firmware version needed for the demonstration is mentioned in the section Motor Control Application Firmware Required for the Demonstration section. This firmware is implemented to work on Microchip's Digital signal controller (dsPIC® DSC) **dsPIC33CDVL64MC106**. For more information, see the **dsPIC33CDVL64MC106 Family datasheet (DS70005441)**.

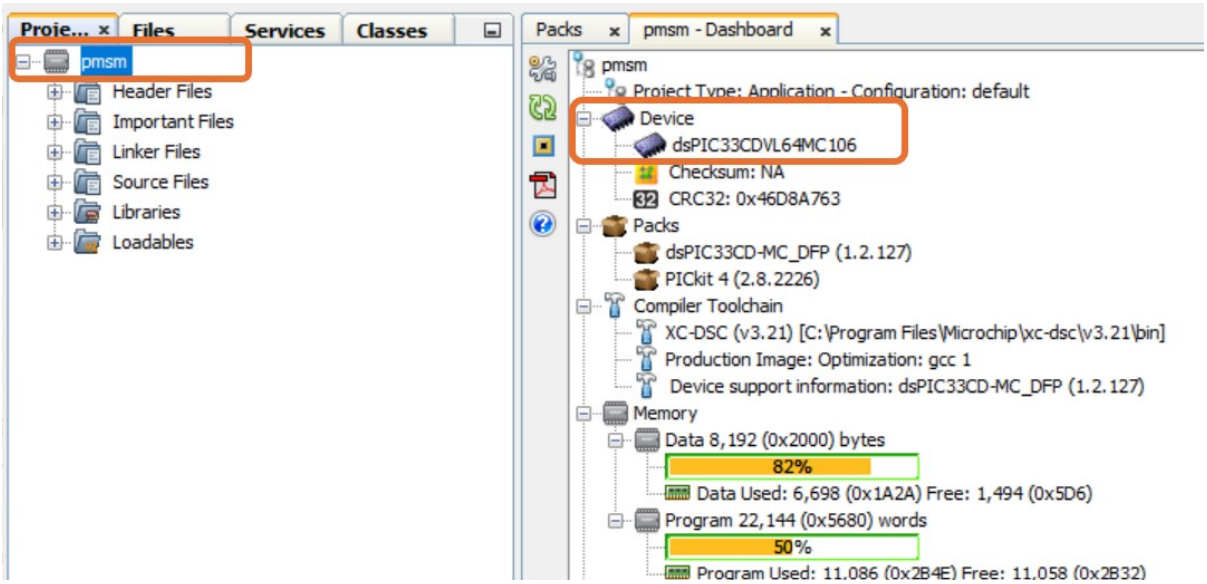
The Motor Control Demo application uses either X2C Scope or LIN Communication with the host PC to control the motor operation.

> **Note:**
> The project may not build correctly in Windows OS if the Maximum path length of any source file in the project is more than 260 characters. In case the absolute path exceeds or nears the maximum length, do any (or both) of the following:
>
> - Shorten the directory name containing the firmware used in this demonstration. If you renamed the directory, consider the new name while reading the instructions provided in the upcoming sections of the document.
> - Place firmware in a location such that the total path length of each file included in the projects does not exceed the Maximum Path length specified.
>
> Refer to MPLAB X IDE help topic **"Path, File, and Folder Name Restrictions"** for details.

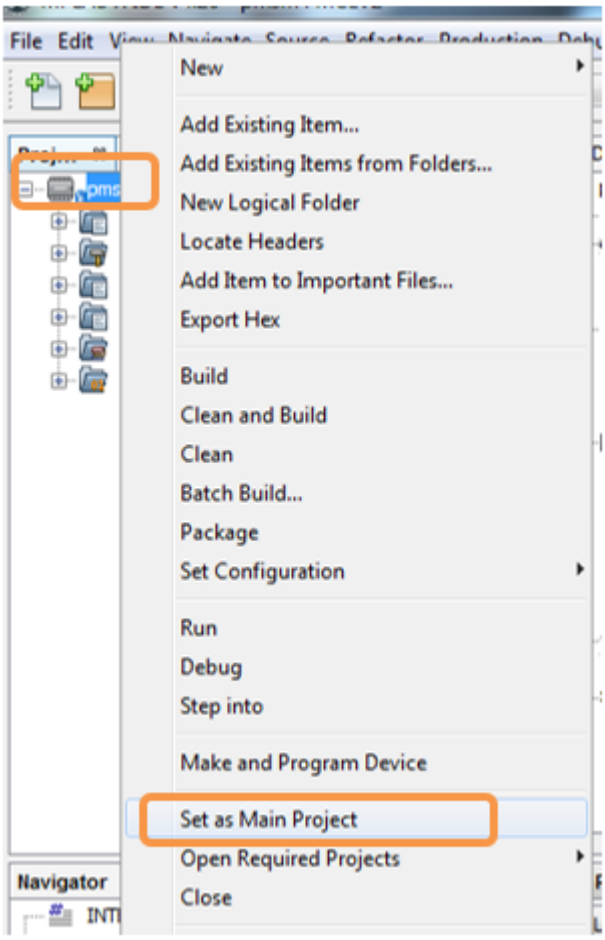
5.2 Firmware Setup

Follow the below instructions, step by step, to set up the motor control demo application firmware:

1. Start **MPLAB X IDE** and open the project **pmsm.X (File > Open Project)** with device selection **dsPIC33CDVL64MC106.**



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.X** will then appear in **bold.**



3. Open `userparms.h` (**pmsm.X > Header Files**) in the project **pmsm.X.**

   - Ensure that the macros `OPEN_LOOP_FUNCTIONING` and `TORQUE_MODE` are not defined, and `SINGLE_SHUNT` is defined.

     ```
     #undef OPEN_LOOP_FUNCTIONING

     #undef TORQUE_MODE

     #define SINGLE_SHUNT
     ```

   - Enable or Disable the macros `WINDMILL_FUNCTIONALITY`,`BRAKE_FUNCTIONALITY` and `IPD_FUNCTIONALITY` as per the requirement.

     ```
     #define  WINDMILL_FUNCTIONALITY     ENABLE

     #define  BRAKE_FUNCTIONALITY        ENABLE

     #define  IPD_FUNCTIONALITY          ENABLE
     ```

   - Motor start/stop and speed demand operations can be controlled using X2CScope or LIN Serial Analyzer. `Define` the macro `LIN_CONTROL` to control the motor using LIN communication or `undefine` the macro to control the motor using X2CScope.

     ```
     #define LIN_CONTROL
     ```

4. The firmware is configured to run a fan load with following motor parameters. To run a motor with different motor parameters, the firmware needs to be re-configured using the excel sheet `tuning_parameters` available in the `docs` folder.

Pole pairs	4	-
Stator resistance (Rs)	0.026	Ohm
Stator inductance (Ls)	3.69E-05	H
Voltage constant (Kfi)	3.62	Vll_pk/KRPM
Maximum Speed	2700	RPM

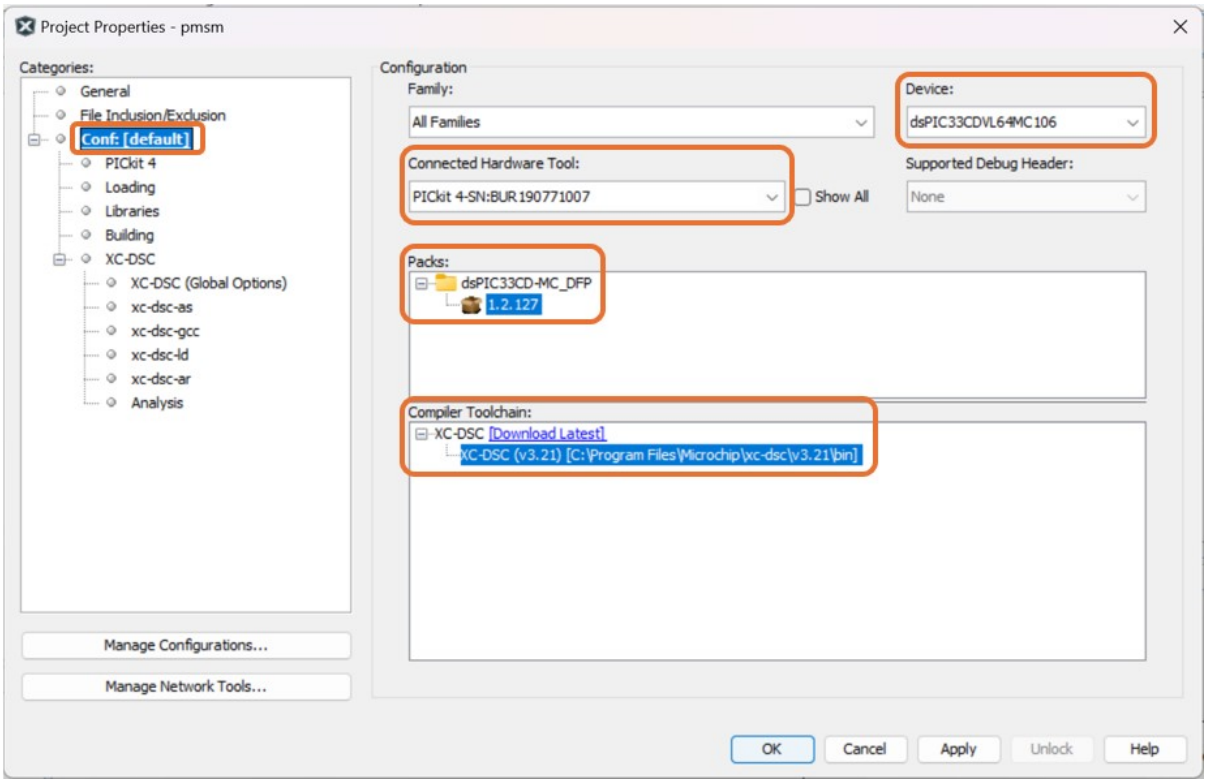
> **Note:**
> The motor phase currents are reconstructed from the DC Bus current by appropriately sampling it during the PWM switching period, called as single-shunt reconstruction algorithm. The firmware is configured to demonstrate **the single shunt reconstruction algorithm** by defining the macro `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."**

5. 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.

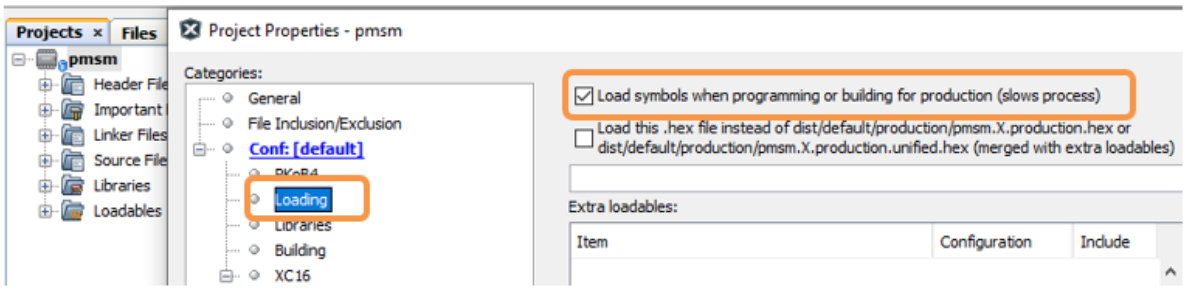
   In the **Conf:[default]** category window:

   - Ensure the selected **Device** is **dsPIC33CDVL64MC106.**
   - 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-MC_DFP 1.2.127** is selected.
   - Select the specific **Compiler Toolchain** from the available list of **XC-DSC** compilers. In this case, **XC-DSC(v3.21)** is selected.
   - After selecting Hardware Tool and Compiler Toolchain, Device Pack, click the button **Apply**

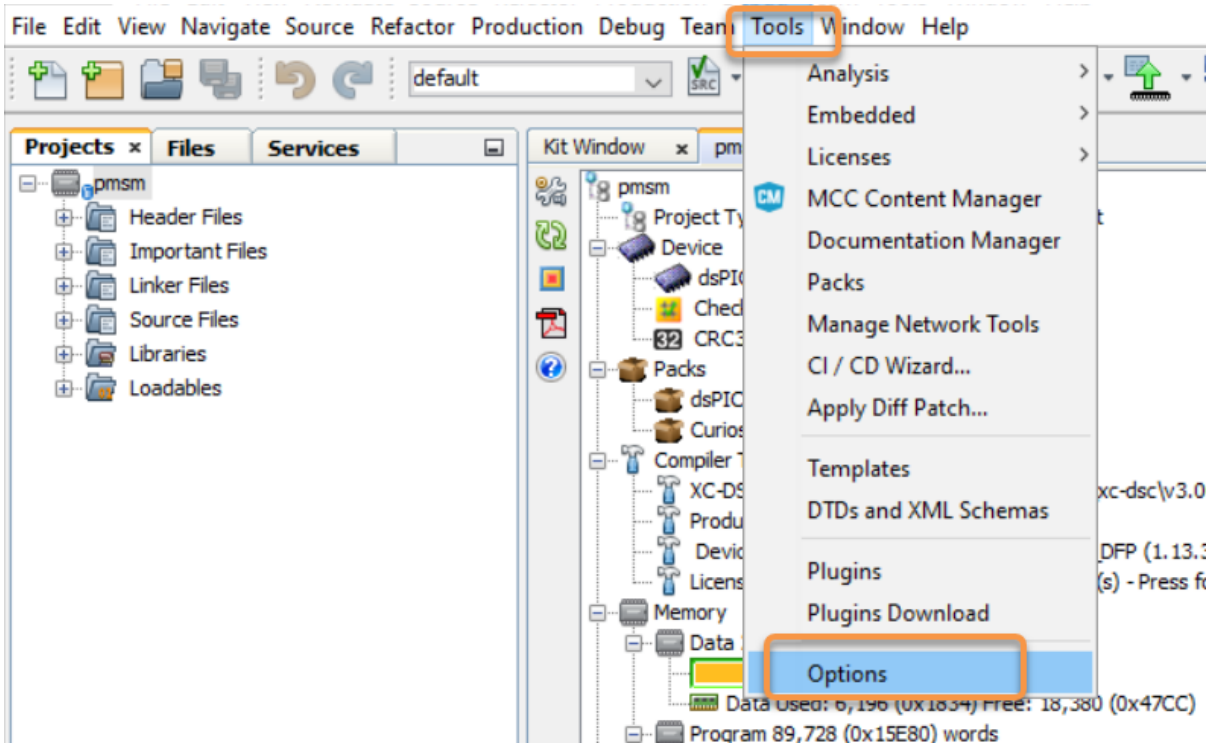
   Please ensure that the selected MPLAB® XC-DSC Compiler and Device Pack support the device configured in the firmware



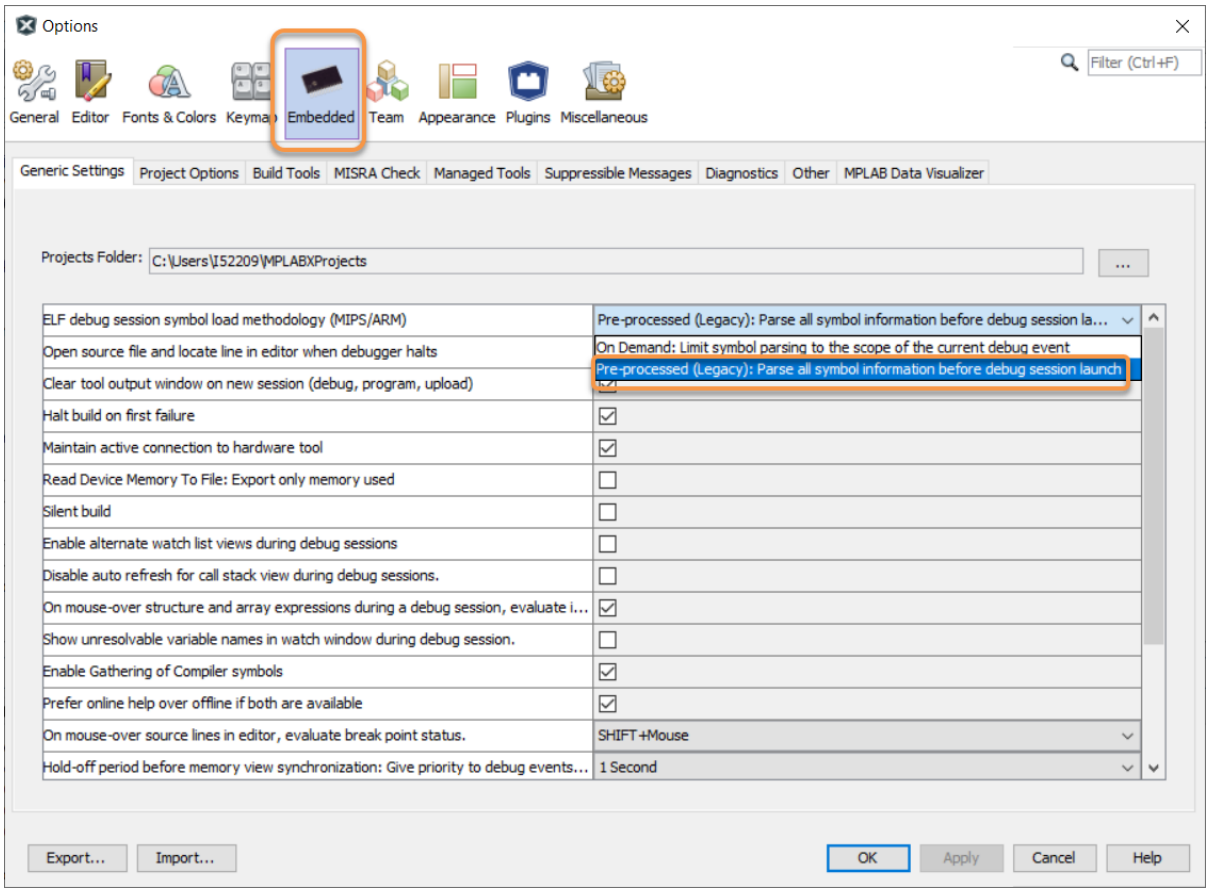
6. Ensure that the checkbox **Load symbols when programming or building for production (slows process)** is checked under the **Loading** category of the **Project Properties** window.



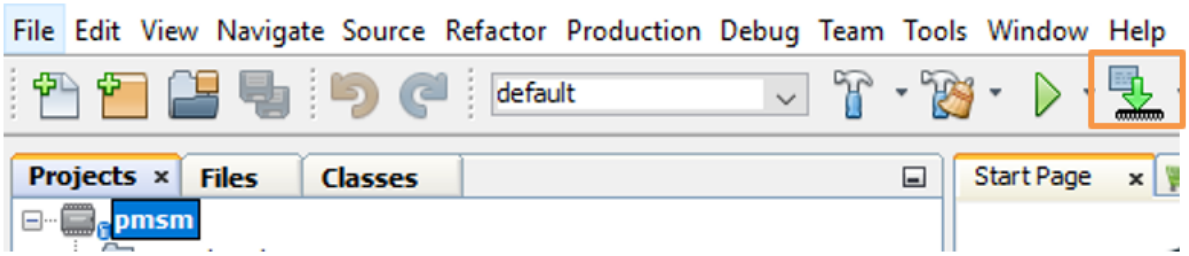
Also, go to **Tools > Options** , and



Open the **Embedded > Generic Settings** tab and ensure that the **ELF debug session symbol load methodology (MIPS/ARM)** is selected as **Pre-procesed (Legacy)** from the drop down.



7. To build the project (in this case, **pmsm.X**) and program the device dsPIC33CDVL64MC106, click **Make and Program Device Main project** on the toolbar



> **Note:**
> The macros `END_SPEED_RPM`, `NOMINAL_SPEED_RPM`, and `MAXIMUM_SPEED_RPM` are specified in the header file `userparms.h` included in the project **pmsm.X.** The macros `NOMINAL_SPEED_RPM` and `MAXIMUM_SPEED_RPM` are defined as per the Motor manufacturer's specifications. Exceeding manufacture specifications may damage the motor or the board or both.

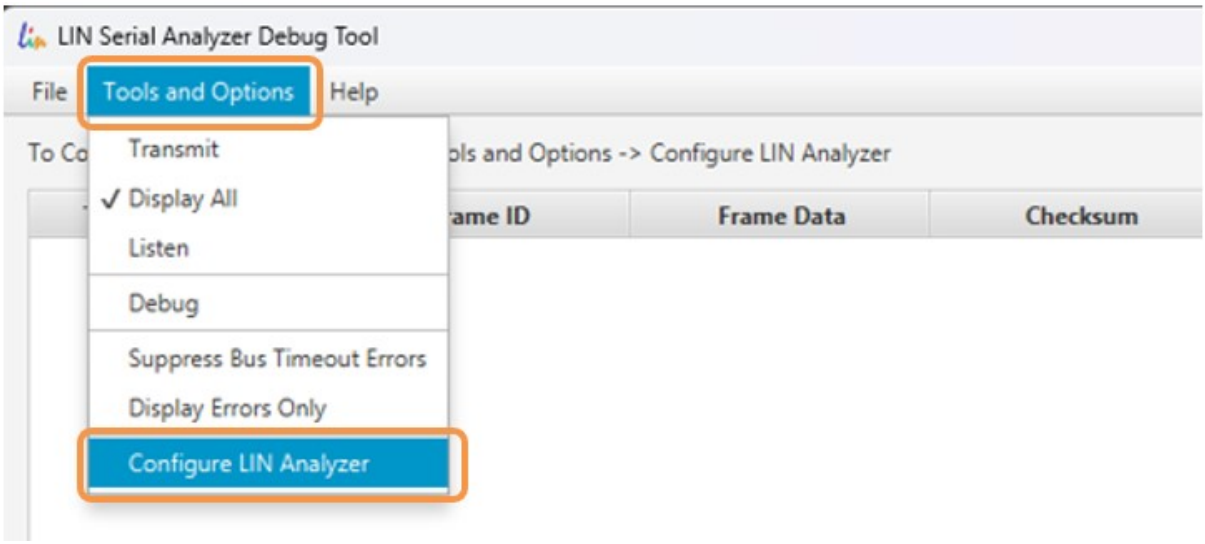
5.3 Motor Operations using LIN Serial Analyzer

The firmware is set to operate the motor using LIN communication by default. The bidirectional LIN commmunication will allow the user to start/stop the motor, set the speed reference and receive the speed feedback by transmitting specific LIN frames.

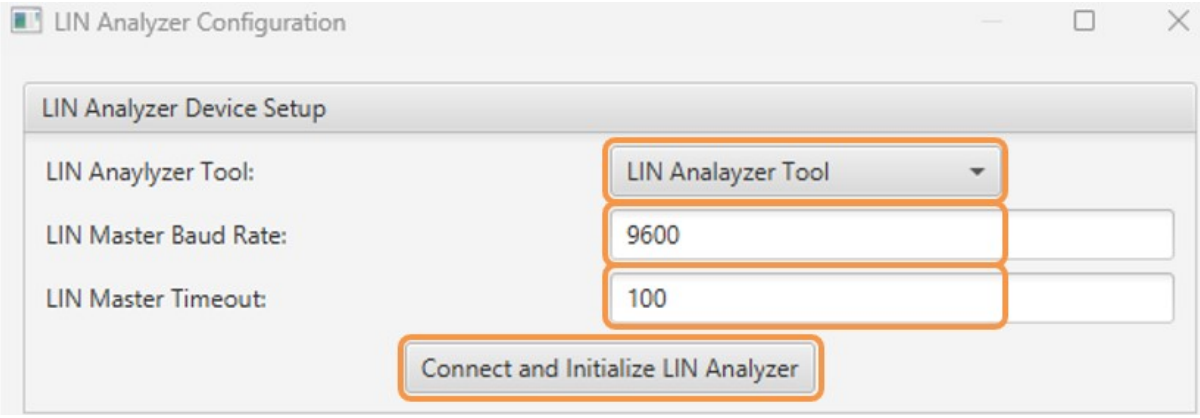
5.3.1 Setting up the LIN Serial Analyzer tool

Follow the below instructions, step by step, to set up the LIN serial analyzer tool:

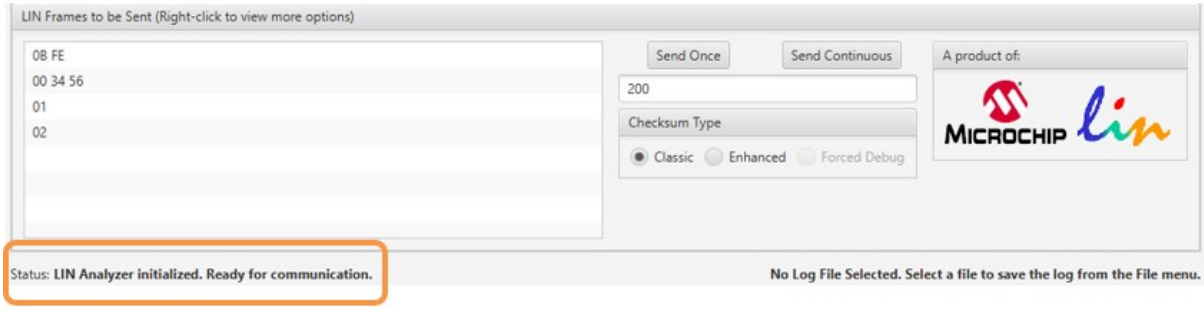
1. Ensure the firmware is configured as described under section 5.2 Firmware Setup by following steps 1 through 7.
2. Make sure the LIN Serial Analyzer (APGDT001) is connected between the **connector J4** on the board and the Host PC.
3. Open the LIN analyzer application installed on the Host PC as described under section 2.2 Software Tools Used for Testing the firmware
4. Go to `Tools and Options` and click on the `Configure LIN Analyzer`



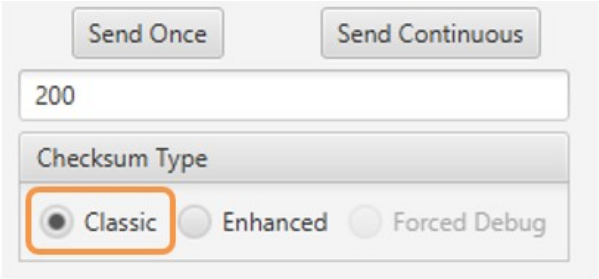
5. Select the `LIN Analyzer Tool` from the drop down. Set the Baud Rate with with `9600` and Timeout with `100` and click on the `Connect and Initiate LIN Analyzer`



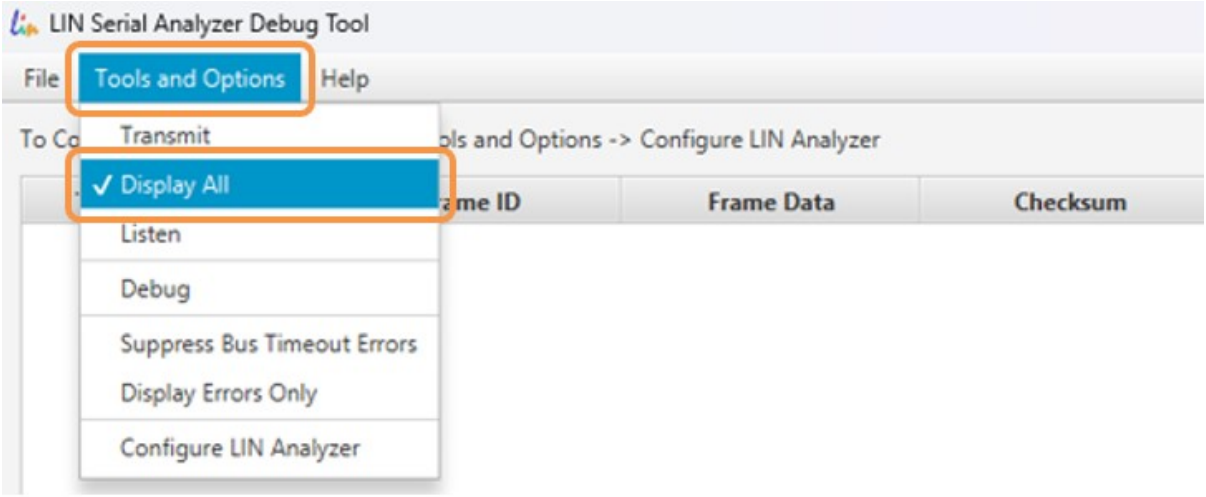
6. Connection status can be verified at the left bottom of the GUI



7. Keep the checksum type as `Classic`



8. Go to `Tools and Options` and ensure `Display All` is selected, so that all frames on the LIN bus are displayed



5.3.2 Start/Stop the motor and Set the speed reference using LIN Serial Analyzer

Motor operations like start, stop and setting the speed reference are implemented by sending specific frames of command through LIN communciation.

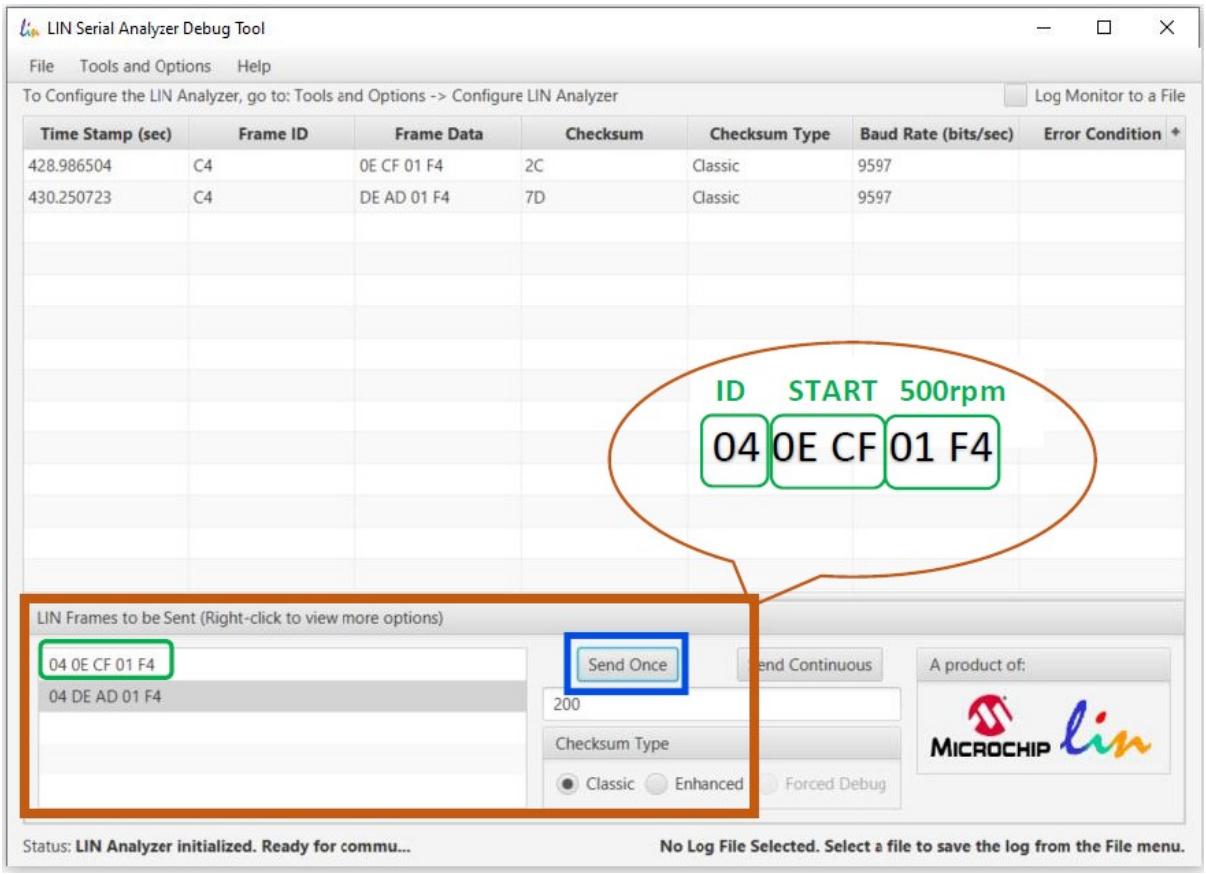
1. Construct a LIN frame with starting PID byte of `04`
    - The first two data bytes in the frame form the start/stop command of the motor

Command	Hexadecimal
START	0E CF
STOP	DE AD

    - The last two data bytes in the frame set the motor speed in Mechanical RPM

Speed (RPM)	Hexadecimal
300	01 2C
500	01 F4
1000	03 E8
2000	07 D0

2. Enter the LIN frame and click on the `Send Once` to transmit the frame. In the below example, the LIN frame `04 0E CF 01 F4` will start the motor with a speed reference of 500RPM. Similarly, motor can be **stopped** by sending a LIN frame of `04 DE AD 01 F4`.

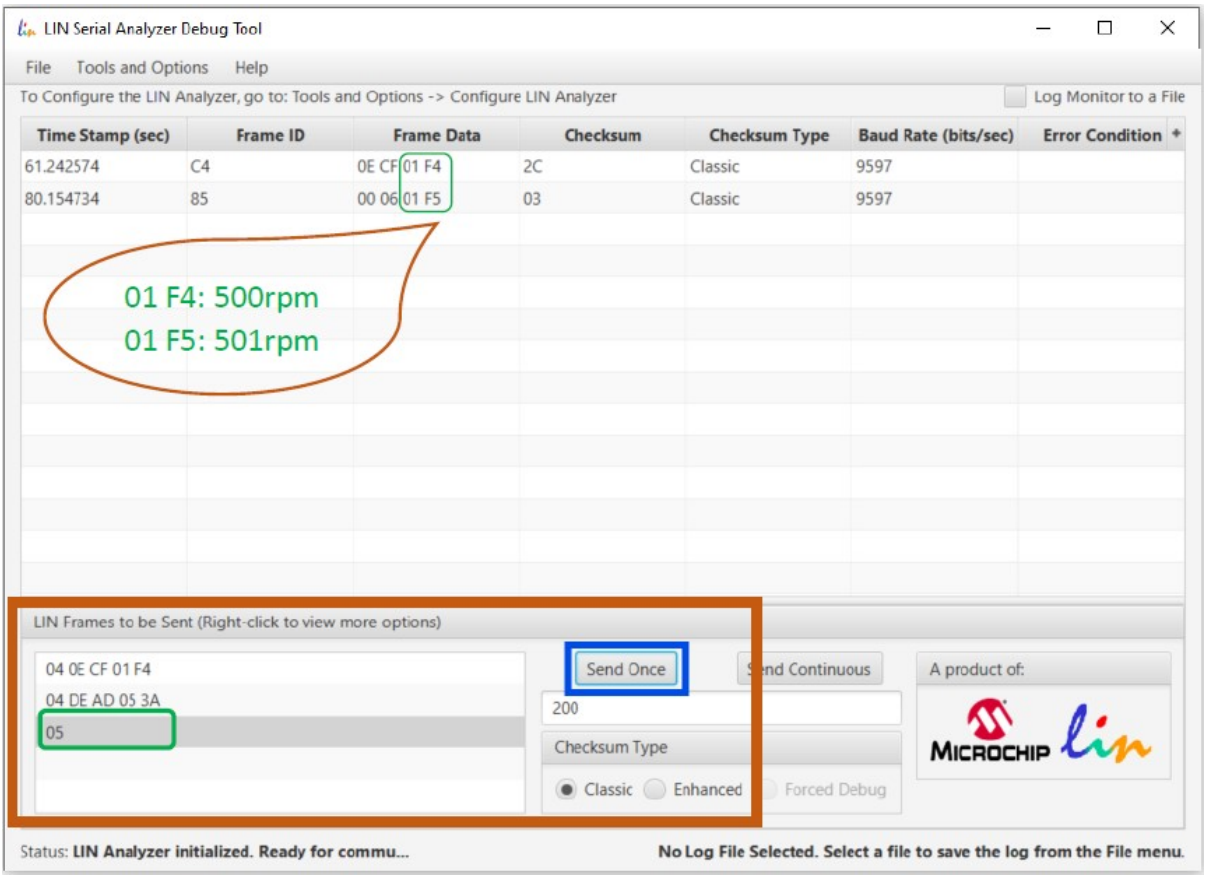


> **Note:**
> Ensure the speed reference is not exceeding the `NOMINAL_SPEED_RPM` defined in the header file `userparms.h`

5.3.3 Receive the speed feedback using LIN Serial Analyzer

Motor speed feedback which is estimated by the firmware, can be read using LIN serial analyzer by following steps,

1. Construct a LIN frame with PID `05` and click on the `Send Once`. The firmware will respond with four data bytes. The first two data bytes contain the status bits for debugging. The last two data bytes are the motor speed in mechanical RPM.



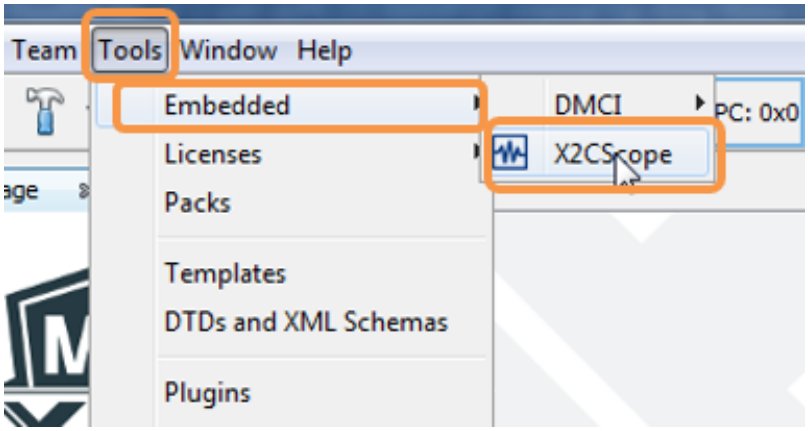
5.4 Data Visualization and Motor Operation through X2C-Scope Plug-in of MPLAB X

X2C-Scope is a third-party plug-in in MPLAB X, which helps in real-time diagnostics. The application firmware comes with the initialization needed to interface the controller with the host PC to enable data visualization through the X2C-Scope plug-in. Ensure the X2C-Scope plug-in is installed. For more information on how to set up a plug-in, refer to either the Microchip Developer Help page or the web page.

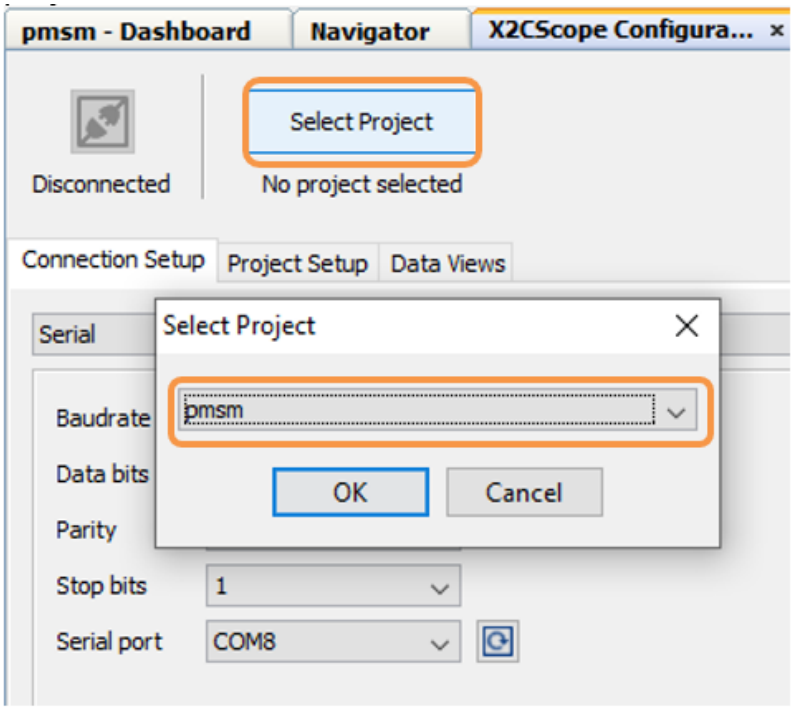
1. To establish serial communication with the host PC, connect a MCP2200 Isolated USB-UART board between the host PC and **connector J2** on the board.

2. Ensure the application is configured as described under section 5.2 Firmware Setup by following steps 1 through 6.

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

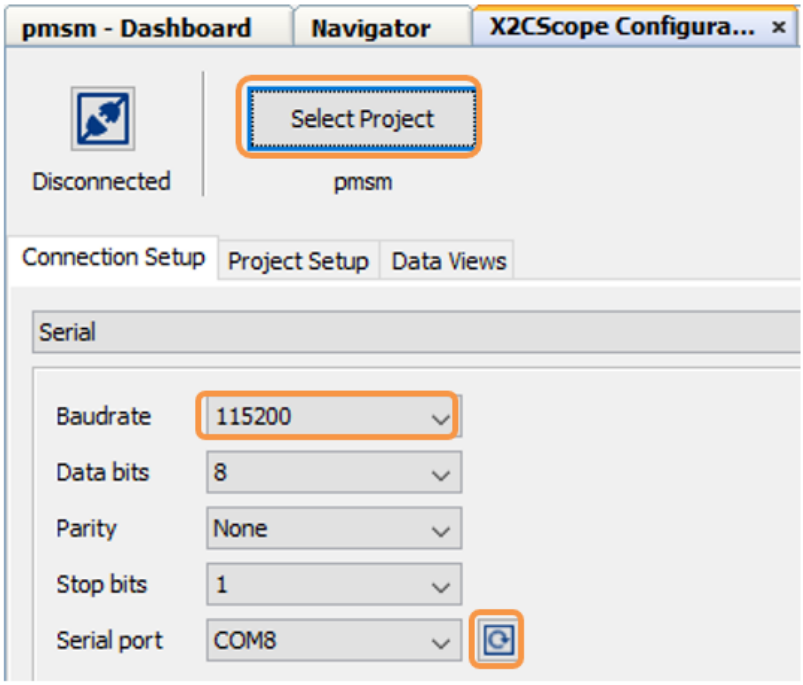


4. **In the X2C-Scope Configuration** window, open the **Connection Setup** tab and click **Select Project.** This opens the drop-down menu **Select Project** with a list of opened projects. Select the specific project **pmsm** from the list of projects and click **OK.**

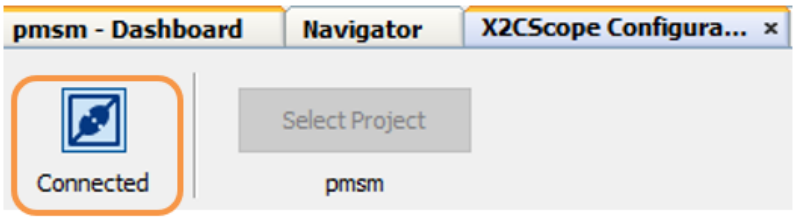


5. To configure and establish the serial communication for **X2C-Scope**, open the **X2CScope Configuration** window, click on the **Connection Setup** tab and:

   - Set **Baudrate** as **115200**, which is configured in the application firmware.
   - Click on the **Refresh** button to refresh and update the list of the available Serial COM ports connected to the Host PC.
   - Select the specific **Serial port** detected when interfaced with the development board. The **Serial port** depends on the system settings

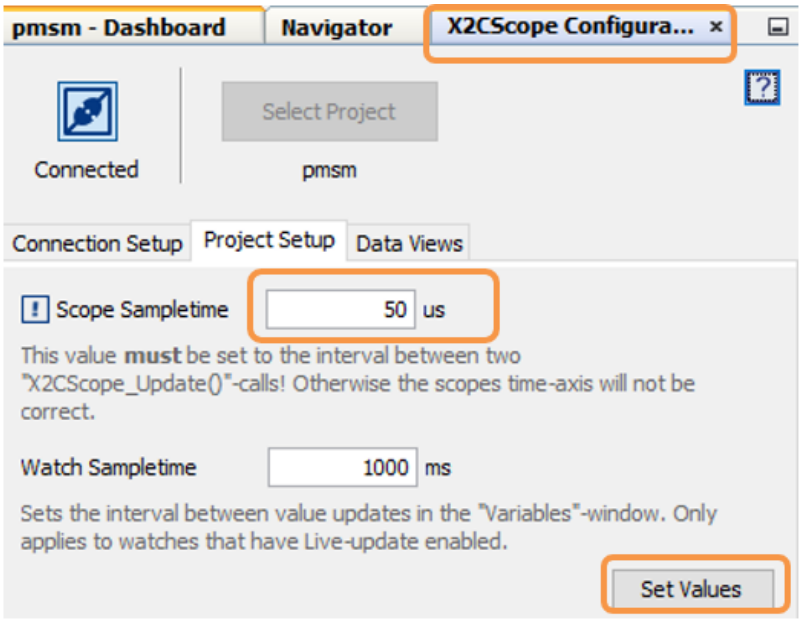


6. Once the **Serial port** is detected, click on **Disconnected** and turn to **Connected**, to establish serial communication between the Host PC and the board.



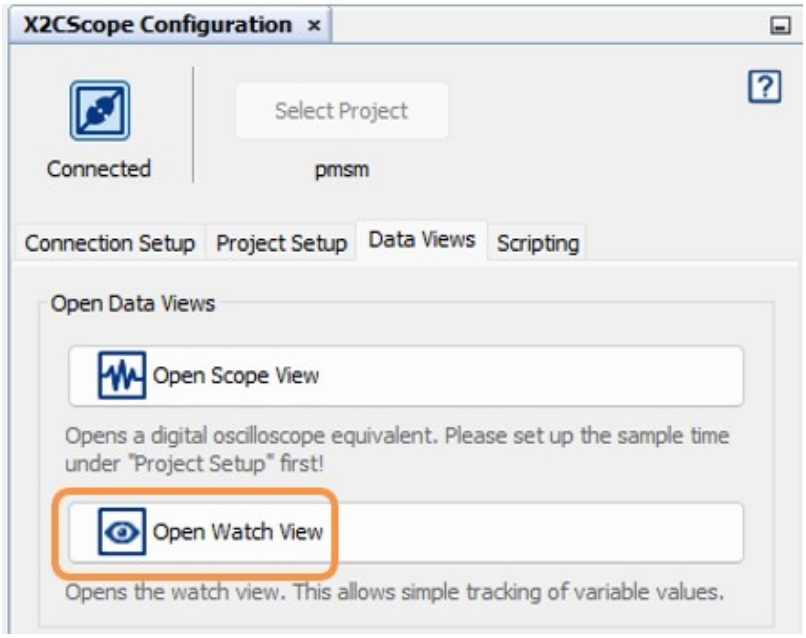
7. Open the **Project Setup** tab in the **X2CScope Configuration** window and,

   - Set **Scope Sampletime** as the interval at which `X2CScopeUpdate()` is called. In this application, it is every `50µs`.
   - Then, click **Set Values** to save the configuration.

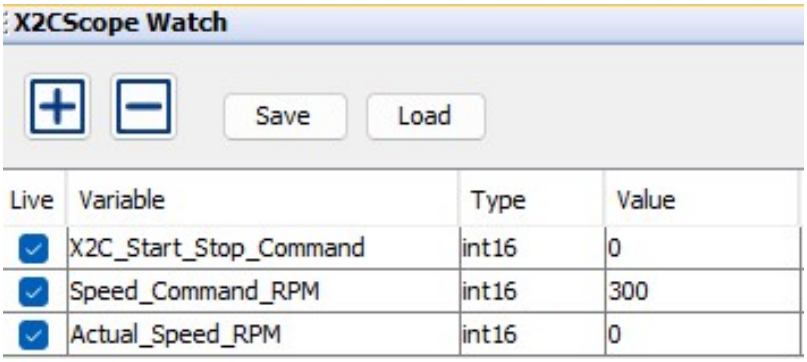


8. If the motor operations are controlled using X2CScope,then

- Click on **Open Watch View** (in the **Data Views** tab of the **X2CScope Configuration** Window); this opens **Watch Window.**

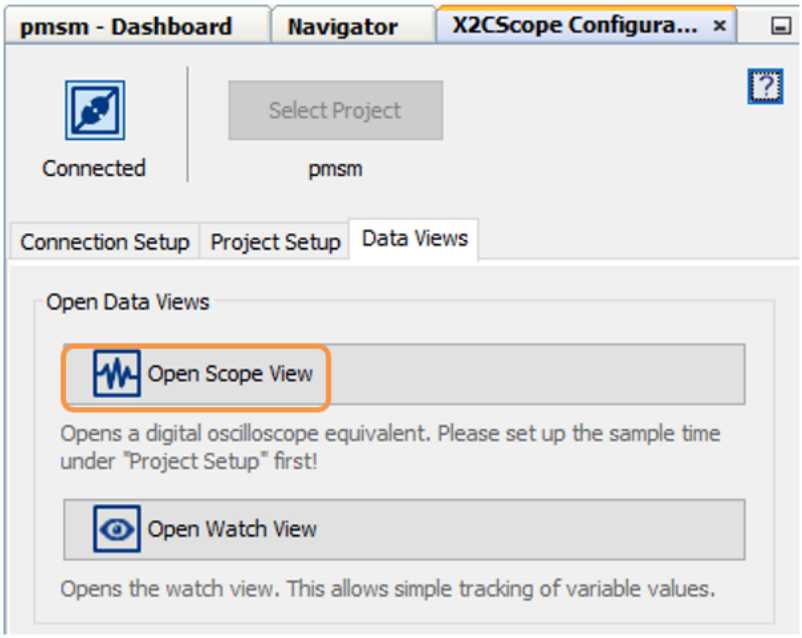


- Set the variable `X2C_Start_Stop_Command` with value **1** to start or stop the motor. Motor speed can be controlled using the variable `Speed_Command_RPM` in mechanical RPM. The `Actual_Speed_RPM` gives the speed feedback in mechanical RPM.

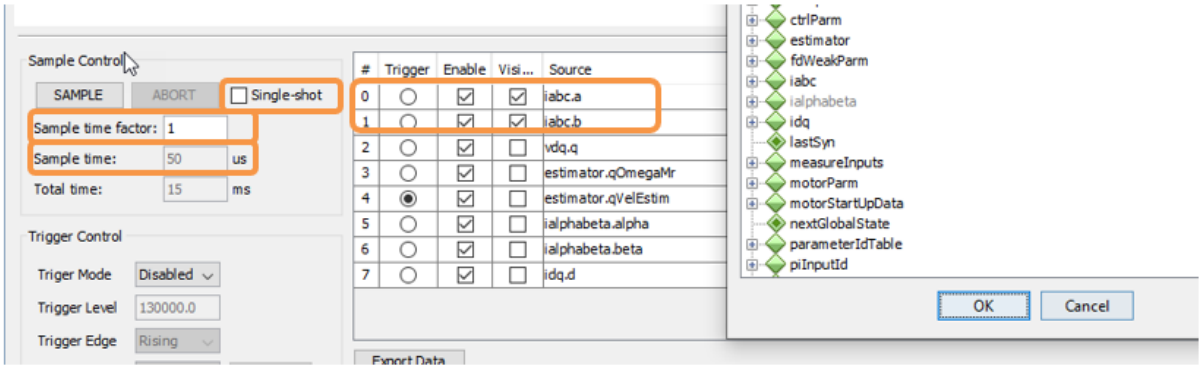


Live	Variable	Type	Value
☑	X2C_Start_Stop_Command	int16	0
☑	Speed_Command_RPM	int16	300
☑	Actual_Speed_RPM	int16	0

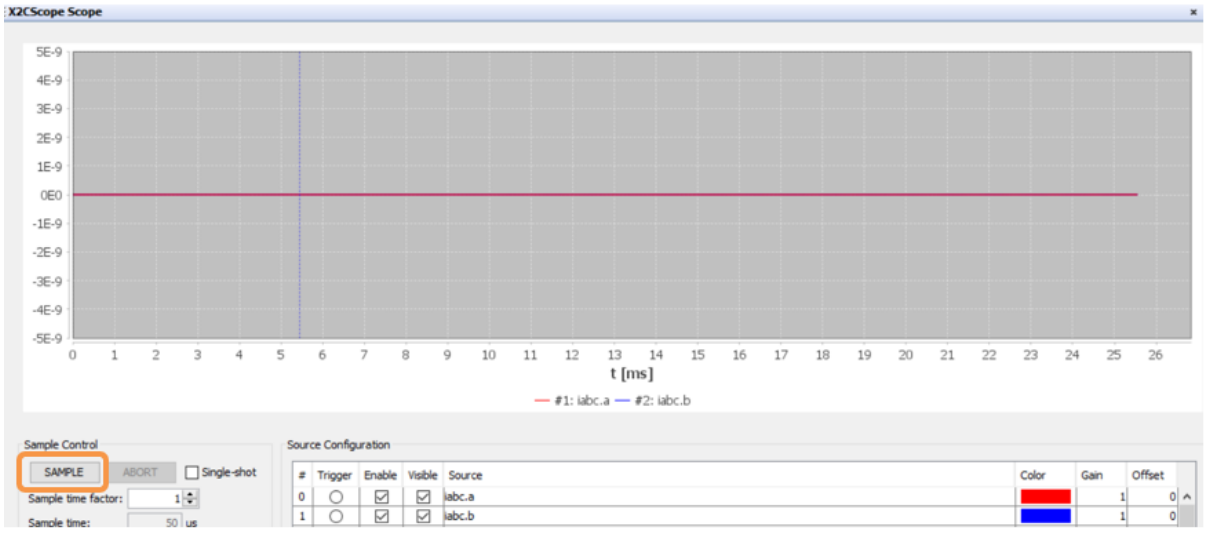
9. Click on **Open Scope View** (in the **Data Views** tab of the **X2CScope Configuration** Window); this opens **Scope Window.**



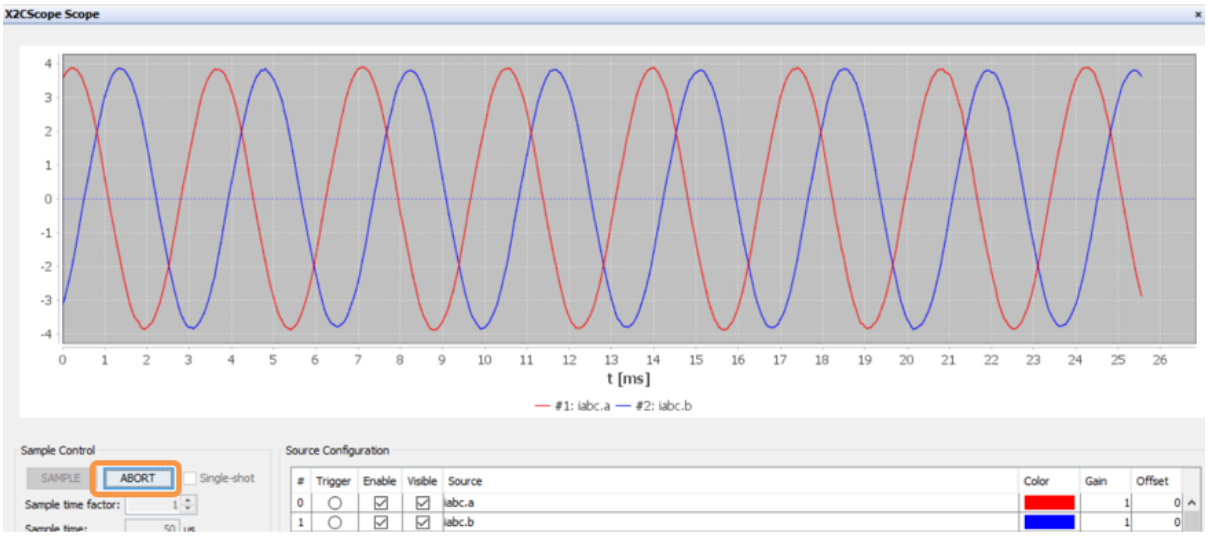
10. In the **Scope Window**, select the variables that must be watched. To do this, click on the **Source** against each channel, and a window **Select Variables** opens on the screen. From the available list, the required variable can be chosen. Ensure checkboxes **Enable** and **Visible** are checked 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 by **Sample time** decides the time difference between any two consecutive data points on the plot.



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



12. Click on **ABORT** to stop.



6. REFERENCES:

For additional information, refer following documents or links.

1. AN1299 Application Note "Single-Shunt Three-Phase Current Reconstruction Algorithm for Sensorless FOC of a PMSM"
2. AN1292 Application Note "Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)"
3. Automotive Cooling Fan Reference Design User's Guide (DS70005523)
4. dsPIC33CDVL64MC106 Family datasheet (DS70005441)
5. MPLAB® X IDE User's Guide (DS50002027) or MPLAB® X IDE help
6. MPLAB® X IDE installation
7. MPLAB® XC-DSC Compiler installation
8. Installation and setup of X2Cscope plugin for MPLAB X
9. Microchip Packs Repository
10. LIN Serial Analyzer User's Guide (DS51675)