

motorBench 2.15.0 Release Notes

Created by Nidhi Shah, last modified just a moment ago

- [What is motorBench® Development Suite library](#)
- [System Requirements](#)
 - [Supported Hardware](#)
 - [Low-voltage hardware](#)
 - [High-voltage hardware](#)
 - [Other hardware required with both low-voltage and high-voltage setups](#)
- [Installing motorBench® Development Suite 2.15.0](#)
- [What's New](#)
- [Hardware Setup](#)
 - [MCLV-2](#)
 - [MCHV-2 and MCHV-3](#)
- [Operating Instructions](#)
 - [Getting started](#)
 - [Configuration](#)
 - [Motor Import and Export](#)
 - [Importing a motor](#)
 - [Exporting a motor](#)
 - [Self-Commissioning](#)
 - [Common types of Self-Commissioning errors and reasons they can occur](#)
 - [Autotuning](#)
 - [Customize](#)
 - [Generate Code](#)
 - [Managing Errors in Code Generation](#)
 - [How to identify code generation errors](#)
 - [How to report specific failure details to Microchip for assistance](#)
 - [Common types of code generation errors and reasons they can occur](#)
 - [Building Code](#)
 - [Running the Application Framework](#)
 - [MCLV-2](#)
 - [MCHV-2 and MCHV-3](#)
 - [Real-Time Diagnostics](#)
- [Repairs](#)
 - [Motor Control Fixed Issues](#)
 - [Changes since revision 2.0](#)
- [Known Issues](#)
 - [Motor Control Issues](#)
- [Troubleshooting Issues with some motors](#)
- [Supported Device](#)
 - [Software Limitations](#)
 - [Motor Control Limitations](#)
 - [Supported Motor Parameters](#)
 - [Changes since revision 2.0](#)
- [Customer Support](#)
 - [The Microchip Web Site](#)
 - [Additional Support](#)

What is motorBench® Development Suite library

Microchip motorBench Development Suite is a graphical, interactive development environment designed to help motor control engineers to design and implement motor control systems, from very basic to very sophisticated ones.

motorBench® Development Suite allows the user to:

- [configure a motor system](#)
- [measure motor parameters](#)
- [tune the controller gains](#)
- [generate code to spin the motor](#)

System Requirements

- [MPLAB X 5.10](#) or later.
- XC16 compiler version: Firmware generated by motorBench® Development Suite has been tested with XC16 1.35. Compiler versions XC16 1.26 and later are expected to work with motorBench® Development Suite but have not been extensively tested.
- [MPLAB Code Configurator®\(MCC\) Plugin Version 3.66](#) or later
- [PIC24/dsPIC33/PIC32MM library 1.115](#) or later

Supported Hardware

This release of motorBench® Development Suite supports both low-voltage and high-voltage setups

Low-voltage hardware

1. dsPICDEM MCLV-2 Development Board [Part Number: [DM330021-2](#)]
2. dsPIC33EP256MC506 External OpAmp Motor Control PIM [Part Number: [MA330031-2](#)] with silicon revision A8 or later.
3. Hurst 24V BLDC motor DMA0204024B101 [Part Number: [AC300022](#)]
4. 24V power supply [Part Number: [AC002013](#)] - ensure this connects to AC mains using a 2-prong cable. If you have an AC002013 with a 3-prong cable, please contact Microchip.

High-voltage hardware

1. dsPICDEM MCHV-2 Development Board [Part Number: [DM330023-2](#)] or dsPICDEM MCHV-3 Development Board [Part Number: [DM330023-3](#)]
AC mains voltages 120VAC 60Hz and 220VAC 50Hz have been tested.
2. dsPIC33EP256MC506 External OpAmp Motor Control PIM [Part Number: [MA330031-2](#)] with silicon revision A8 or later.
3. Leadshine 400W BLDC motor EL5-M0400-1-24 [Part Number: [AC300025](#)]

Other hardware required with both low-voltage and high-voltage setups

1. A USB-to-logic-level-UART converter from the following list:
 - a. Saelig [USB-COM-U](#) or [USB-COM-U13](#)
 - b. [TRENDnet TU-S9 v2.0](#)
2. Programming tool - one of the following tools: Real ICE, ICD3
3. Custom calibration board [if needed, contact MCU16 Motor Control application team for more details]

Please note:

We do not recommend using the MCP2200 USB to RS232 Demo Board [Part number: [MCP2200EV-VCP](#)] with this release of motorBench® Development Suite. While testing, we have observed more frequent occurrence of a serial communication timeout issue while running Self-Commissioning process using this cable. See [Known Issues](#) section of this document for more information (MCGUI-1141)

Installing motorBench® Development Suite 2.15.0

To install the MPLAB® Code Configurator v3.66 Plugin

1. In the MPLAB® X IDE, select **Plugins** from the Tools menu
2. Select the **Available Plugins** tab
3. Check the box for the MPLAB® Code Configurator v3, and click on **Install**

To install different peripheral library version or motorBench® Development Suite version when connected to internet

1. Create a project with dsPIC33EP256MC506 or use the sample project.
2. Open MPLAB® Code Configurator
3. In the Versions tab under **PIC24/dsPIC33/PIC32MM** MCUs, find the multiple library versions (loaded version is indicated by the green check mark)
4. Right-click on the required version of the library and select **Mark for Load**
5. In the Versions tab under **motorBench® Development Suite** find the multiple library versions (loaded version is indicated by the green check mark)
6. Right-click on the 2.15.0 version of the library and select **Mark for Load**
7. Click on **Load Selected Libraries** button to load the marked libraries.

To install different peripheral library version or motorBench® Development Suite version when not connected to internet

1. In the MPLAB® X IDE, select **Options** from the **Tools** menu
2. Select **Plugins** tab
3. Click on **Install Library**
4. Add **pic24-dspic33-pic32mm_v1.115.mc3lib**
5. Add **motorBench_2.15.0.mc3lib**
6. Restart MPLAB® X IDE

What's New

1. Motor Control Application Framework (MCAF) R4 – see MCAF User's Guide for more information.
 - a. Changes from R3 to R4: Full integration with MCC, QEI support, UI-configurable parameters, other minor fixes/improvements
 - b. Changes from R2 to R3: Integration with MCC, MCHV-2 and MCHV-3 support, other minor fixes/improvements
 - c. Changes from R1 to R2: Support for a wider range of motors, support for overmodulation and DC link voltage compensation
2. Configure Phase
 - a. Support import and export of motors
 - b. Support QEI estimator
3. Customize
 - a. Allow advanced customization of MCAF code generation
4. MCC Integration
 - a. All peripheral and system initialization code is generated by MCC.

Hardware Setup

The following sections outline the hardware setup of MCLV-2, MCHV-2, and MCHV-3.

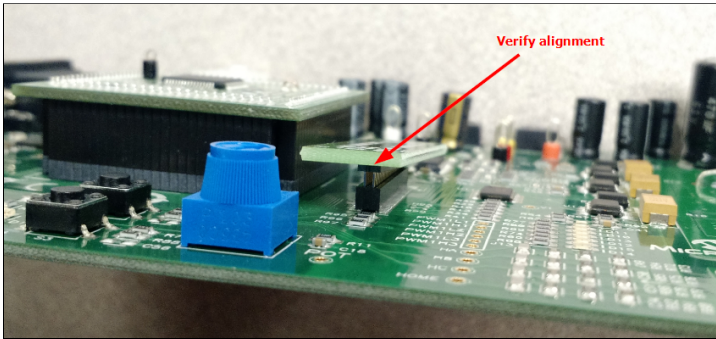
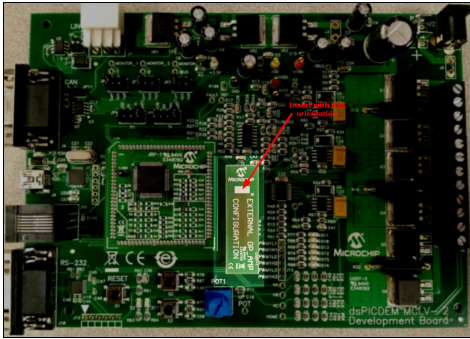
MCLV-2

› Hardware setup: MCLV-2

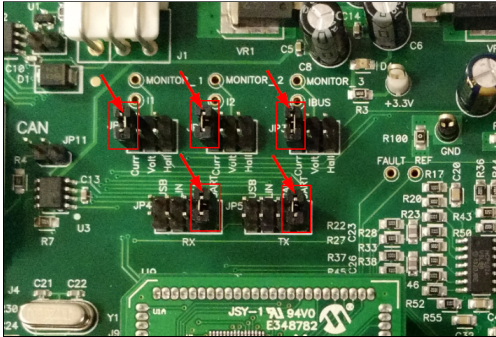
This section provides detailed steps that will help you setup your hardware to work with the motorBench® Development Suite:

1. MCLV-2 board comes pre-installed out of the box with dsPIC33EP256MC506 Internal OpAmp Motor Control PIM (MA330031); replace this PIM with the dsPIC33EP256MC506 External OpAmp PIM (MA330031-2) specified above.
2. Make sure that the dsPIC33EP256MC506 External OpAmp PIM is populated with a silicon mask rev-A8 or later. To verify this, read out the device revision from MPLAB X and verify that the Device ID revision is equal to or greater than `0x4008`.

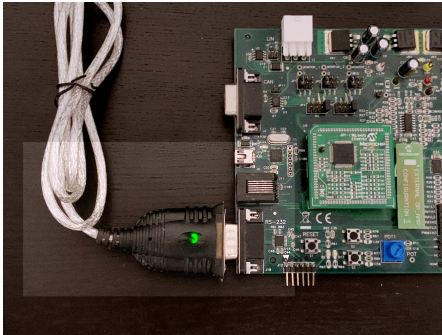
3. Install the External OpAmp Configuration matrix board that comes with the MCLV-2 board into J14



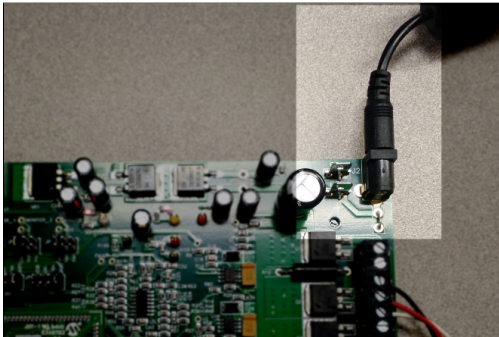
4. Update the jumpers JP1-JP2-JP3 to Curr position and JP4-JP5 to UART position as shown below:



5. Connect the USB-to-logic-level-UART converter cable into J10 of MCLV-2 board and one of the USB ports on your PC:

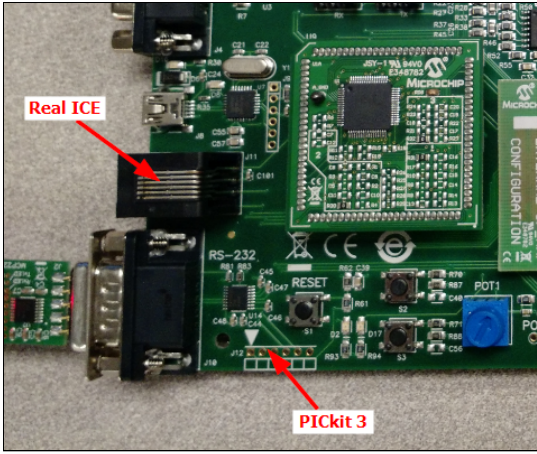


6. Plug in the 24V power supply barrel connector into J2 of MCLV-2 board and then plug in the power supply into the mains outlet:

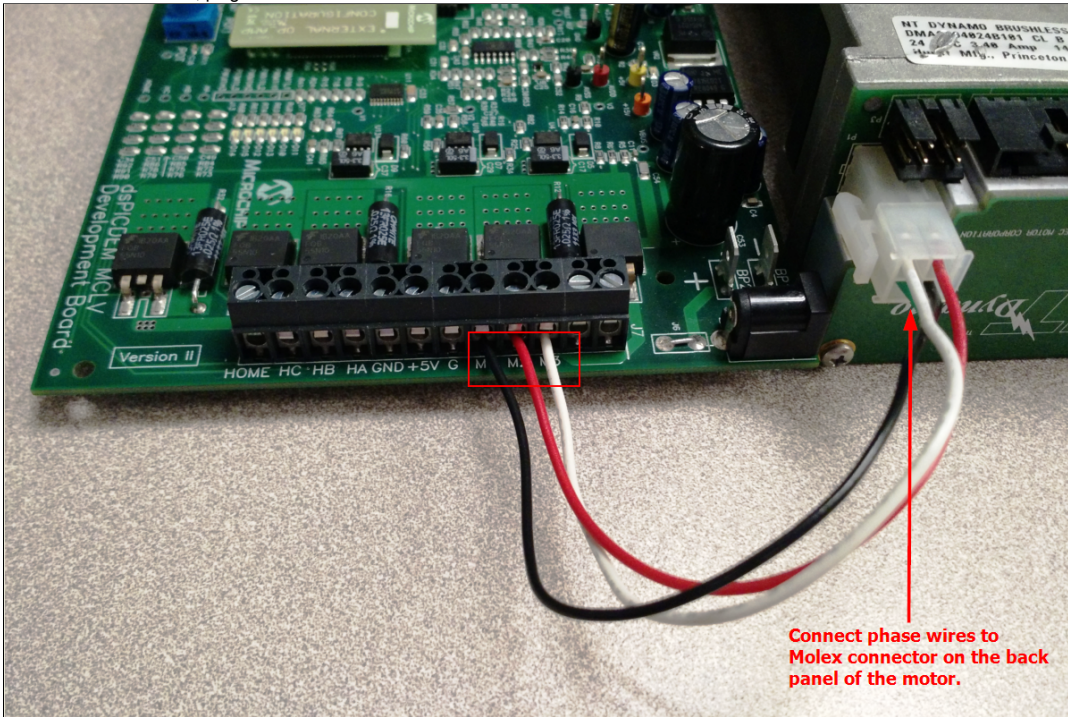


7. Connect your Real ICE / ICD3 / PICKit3 to one of the USB ports on your PC

8. Connect the Real ICE / ICD3 to J11 of MCLV-2 board. If you are using PICKit 3, then connect it to J12 of this board.

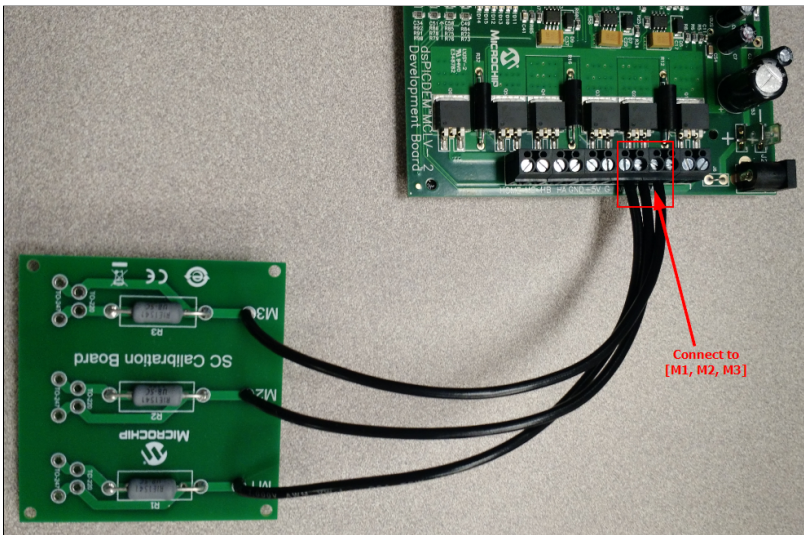


9. Connect the [black, red, white] phase wires of the 24V BLDC motor to [M1, M2, M3] terminals of J7 connector on the MCLV-2 board. (Sequence / order of this connection is not important if the direction of rotation does not matter.) You may leave the green color phase wire unconnected or connect it to the G terminal of J7 connector on the MCLV-2 board. Then, plug in the other end with a Molex connector to the 24V BLDC motor.



10. Keep the motor on a stable surface and use a clamp (if available) to secure the motor from jumping around. (If using a metal C-clamp, make sure there is a thin shim of rubber, cloth, wood, or other mechanically-compliant material between the clamp and the motor, to avoid deforming the motor housing.) Alternatively, you can also place the motor on a rubber mat. Also, do not disturb the motor or hold its shaft while Self-Commissioning is running.

11. While running the Board Calibration feature in motorBench® Development Suite, you will need to use the Calibration board instead of the 24V BLDC motor. In this case, start by stripping off the insulation on the wires coming from the m1, m2 and m3 terminals of the Calibration board. Then connect these wires into M1, M2, M3 terminals of J7 connector on the MCLV-2 board.



12. When asked within motorBench® Development Suite, use these values:

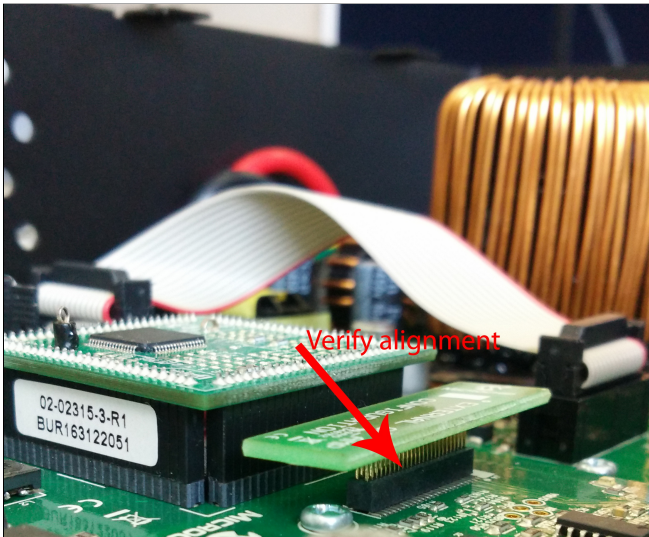
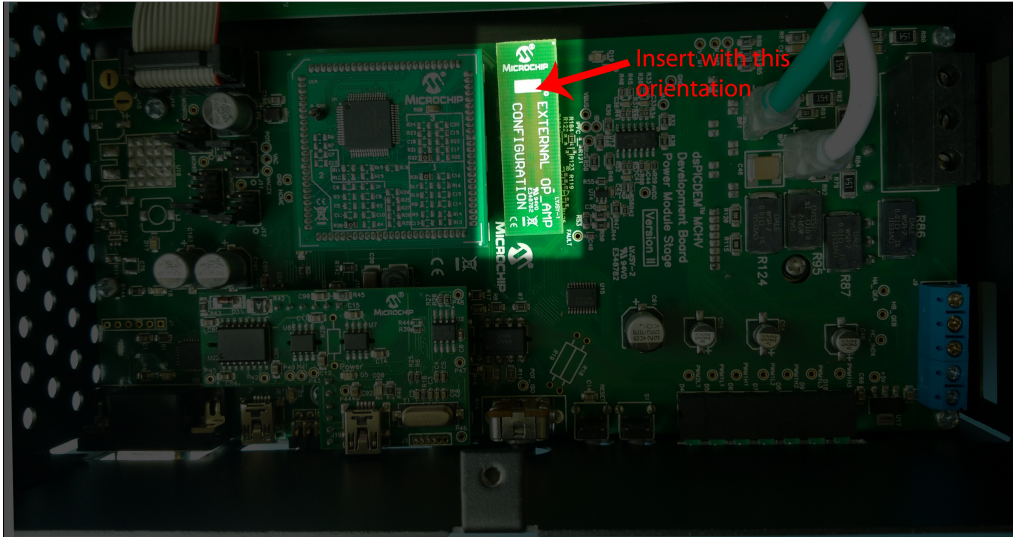
- a. Calibration resistor (R_{CAL}) = 12Ω
 - b. Calibration bus voltage (V_{BUS}) is typically around 24V - this needs to be measured with a digital multi-meter (DMM) for accurate SC results.
13. If, for whatever reason, you need to reset the firmware that runs on the MCLV-2 board, then use push button S1 for this.

MCHV-2 and MCHV-3

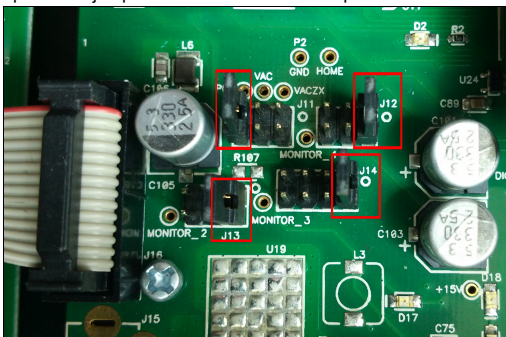
- > [Hardware setup: MCHV-2 and MCHV-3](#)
- Hardware setup: MCHV-2 and MCHV-3

This section provides detailed steps that will help you setup your hardware to work with the motorBench® Development Suite: Although the setup is similar for MCHV-2 or MCHV-3, pictures of MCHV-3 are shown in the below section

1. MCHV-3 board comes pre-installed out of the box with dsPIC33EP256MC506 Internal OpAmp Motor Control PIM (MA330031); replace this PIM with the dsPIC33EP256MC506 External OpAmp PIM (MA330031-2) specified above.
2. Make sure that the dsPIC33EP256MC506 External OpAmp PIM is populated with a silicon mask rev-A8 or later. To verify the revision of silicon, read out the device revision from MPLAB X and make sure that the Device ID revision is equal to or greater than 0x4008.
3. Install the External OpAmp Configuration matrix board that comes with the MCHV-3 board into J4



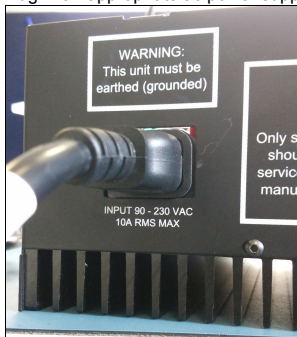
4. Update the jumpers J12-J13-J14 to 1-2 position and J11 to 5-6 position as shown below:



5. Connect the USB-to-UART converter cable into RS232 labeled port on MCHV-3 board and one of the USB ports on your PC: Please note that in the event of mechanical interference with the USB-to-UART cable, you may need to use an UART extension cable.



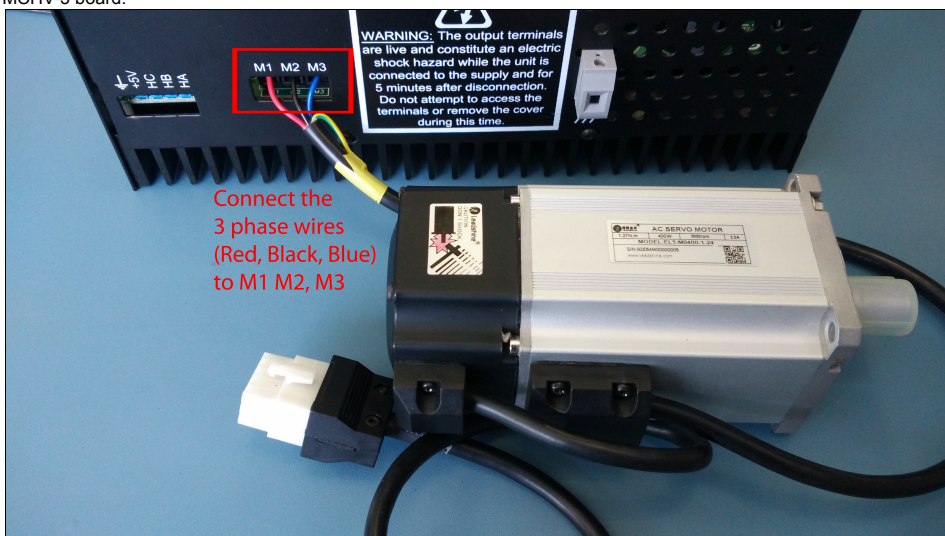
6. Plug in an appropriate ac power supply cable to MCHV-3 board and then plug in the ac power supply cable into the mains outlet:



7. Connect the provided USB cable to PROGRAM/DEBUG connector on MCHV-3 board and to one of the USB ports on your PC.



8. Connect the [red, black, blue] phase wires of the Leadshine 400W 220VAC Servo Motor (AC300025) to [M1, M2, M3] terminals on the MCHV-3 board. (Sequence / order of this connection is not important if the direction of rotation does not matter.) Connect the green-yellow combination color wire to the Ground terminal provided on the MCHV-3 board.



9. Keep the motor on a stable surface and use a clamp (if available) to secure the motor from jumping around. (If using a metal C-clamp, make sure there is a thin shim of rubber, cloth, wood, or other mechanically-compliant material between the clamp and the motor, to avoid deforming the motor housing.) Alternatively, you can also place the motor on a rubber mat. Also, do not disturb the motor or hold its shaft while Self-Commissioning is running.

10. While running the Board Calibration feature in motorBench® Development Suite, you will need to use the HV Calibration board instead of the Leadshine 400W 220VAC Servo Motor (AC300025). In this case, start by stripping off the insulation on the three wires coming from the HV Calibration board box. Then connect these wires into M1, M2, M3 terminals of connector on the MCHV-3 board.

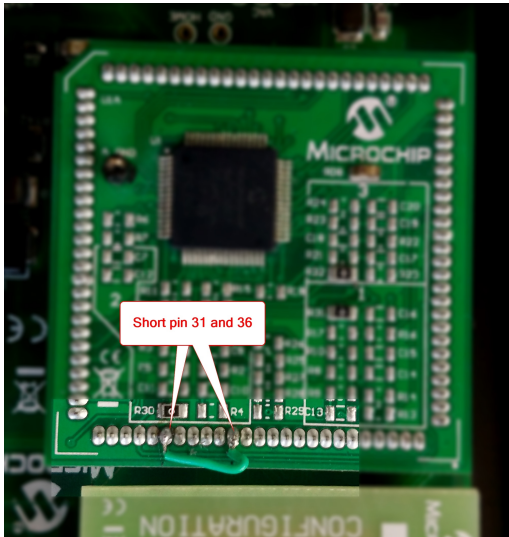


Whenever connecting or disconnecting a motor/ calibration board from MCHV-2/ MCHV-3 board, please disconnect the power to the MCHV-2/ MCHV-3 board. Also, use an insulated screwdriver that is recommended for high voltage rating.

11. When asked within motorBench® Development Suite for MCHV-2/ MCHV-3, use these values:
- Calibration resistor (R_{CAL}) = 50 Ω
 - Calibration bus voltage (V_{BUS}) is typically around 311V (for an AC input of 220V RMS) - this can be changed in the board parameters during Define stage if a different voltage source is used
12. If, for whatever reason, you need to reset the firmware that runs on the MCHV-3 board, then press the button labeled RESET.

- ⓘ There is a known hardware design limitation with MCHV-2 and MCHV-3 boards that can cause the dsPIC device to reset when running motors with large phase currents. If you are observing dsPIC device reset issues while using MCHV-2 or MCHV-3 board with certain test motors, **power down the board, unplug the AC power cable, wait until LED D13 is OFF** and then make the following modification:

Use a short length of jumper wire to connect digital ground signal to analog ground signal on the PIM. To do this, connect pins 31 and 36 on the dsPIC33EP256MC506 External OpAmp Motor Control PIM:

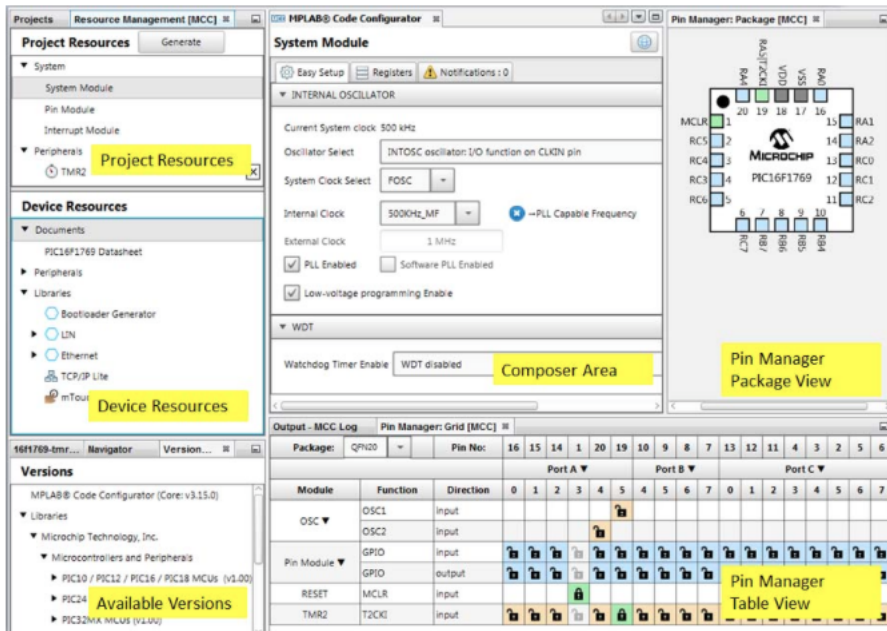



Operating Instructions

Getting started

Please start with MPLAB X project `sample-motorBench-project-mclv2.X.zip` or `sample-motorBench-project-mchv2.X.zip` – the compiler, linker, and MCC System and Peripheral settings have been preset for use with the Motor Control Application Framework.

- Unzip either file onto your computer
- Open the sample-motorBench-project in MPLAB X
- Right-click and set as the main project in MPLAB X
- Right-click and open project properties. Select an in-circuit programmer (we have successfully tested ICD3 and REAL ICE) and XC16 compiler.
- The MCC tool, once installed, can be launched from the MPLAB® X Tools menu under the Embedded selection, or by Clicking on the MCC icon in MPLAB X Toolbar.
- The sections of the MCC tool are shown in the image below. We will refer to these sections by name throughout this release notes.



- The MCC help can be invoked by clicking on the 

Configuration

1. motorBench® Development Suite requires all peripheral and system settings to be setup in MCC. The sample-motorBench-project-mclv2 or sample-motorBench-project-mchv2 would setup all peripheral, system, compiler, and linker settings. To set up MCC manually refer to the Peripheral Configuration Guide.
2. Double-click on motorBench® Development Suite Development Suite from device resources to add it to project resources.
3. If you haven't registered, you will be given the option to register motorBench, or enter a 30 day trial period. You will not be able to use motorBench after 30 days. For registration steps on an offline machine please contact motorbench-support@microchip.com
4. motorBench® Development Suite has three separate views
 - a. Composer view : This allows users to define and measure the system characteristics, tune the system characteristics and customize MCAF generation parameters. In composer view, the user can
 - i. Configure the system: Allows user to configure the motor, board and estimator details. Users have the option to measure the electrical and mechanical parameters. Users can export new motors.
 - ii. Tune the system: Once a motor has been imported, you can tune the system.
 - iii. Customize the MCAF parameters: Allows users to customize MCAF parameters that are used in code generation.
 - b. Pin Manager View: This allows users to see the various pins that are being used by the system.
 - c. Summary View: This allows users to see the various settings that are made by the user in the system.
5. Once you register, the "Home" page of the Composer view would show "Registered", otherwise it would show "Not yet Registered"
6. You can navigate between the "Configure", "Tune" and "Customize" stages using their respective buttons.
7. Once all the components fully defined and there are no configuration errors, a puzzle icon will appear with status as "Ready to Generate". If the puzzle icon doesn't appear complete, check if there are any messages present in the "Notifications[MCC]" tab below that prevent completion, and resolve them.

Motor Import and Export

Each motor has a different characteristics, motorBench® Development Suite ships with two standard motor definitions: **Hurst300** and **Leadshine400**.

You can Import either of these motors and modify if needed for the specific motor. Once modified you can export the changes as a different motor and use that later or share it with others if needed.

The process of selecting a standard motor shipped with motorBench® Development Suite or created using export is called **Import**. Modifying and persisting it called **Export**.

Importing a motor

When the project is started for the first time, there would be no motor selected. Users have to click on the **Import Motor** button to select one of the exported motors or one of the motorBench® Development Suite standard motors.

Begin by importing an existing motor. Clicking the 'Import Motor' button automatically takes you to the location on your machine where the default motors are stored.

Location of default motors on Windows

C:\Users\<username>\.mcc\libraries\<versioned motorBench® Development Suite folder>\motors

Select your desired motor, and click 'Open'. The motor will be imported into motorBench® Development Suite.

motorBench® Development Suite is now ready to generate code. If you don't have any modifications to make to the project, you can generate code.

Exporting a motor

Once you have imported an existing motor, you may choose to change its parameter values.

Motor and your project



Note that once you import a motor into motorBench® Development Suite, there is no link between your project, and the file from which it was imported. You can see the file from which the motor was imported in the 'Output - MPLAB® Code Configurator' window, but any changes you make are local to your project and do not affect the file.


All the motor parameters currently displayed in the composer view are editable, except the 'Measured Values' column in the 'Electrical and Mechanical Parameters' section. 'Measured Values' are automatically calculated by the Self-Commissioning process discussed separately.

All numeric values are shown in Engineering Notation, but if you hover over them, you will be able to see their exact 'double' representations. This is the value that you will be editing. Currently, we only support entering motor values in the units shown by default beside each parameter.

If you want to save the changes that you made to the motor parameters, you have two choices -

1. If you exit MCC, MPLAB X, or close the project, you will be asked whether you want to save the current MCC configuration. If you click 'Yes', all the changes will be preserved for this project. The next time you open MCC for this project, you will find all your values intact. This will not export the motor file.
2. If you want to use your motor parameters for other/new projects, you can choose to export the motor. Clicking the 'Export Motor' button will bring up a dialog box that will allow you to create your own motor with a new name. In a different project, you may choose to import this file instead of the default files provided with this installation.

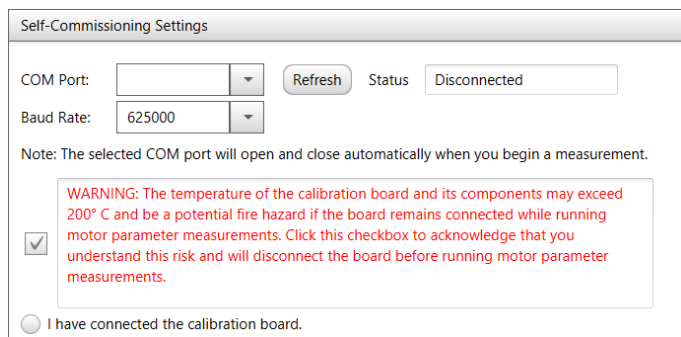
 Note that only the 'Active Values' in the 'Electrical and Mechanical Parameters' section are exported. 'Measured Values' cannot be exported. Measured values will, however, be preserved for that specific project. If you have measured values generated by the Self-Commissioning process, you may choose to copy them to the 'Active Values' column by clicking the  or 'Use all' buttons.


 **Editing exported files**
We do not recommend manually editing exported files. Using the UI to export motors is the best way of making any changes to the parameters you desire. If you still do edit the file manually, there is a possibility that importing that motor into a project might fail without any warning or notification.

Self-Commissioning

After importing a motor, click on the "Measure Now" button in the Motor's Electrical and Mechanical Parameters to open up the Self-Commissioning Settings.


1. Before you begin measurement, make sure that you have selected an ICD tool and a compiler.
2. Select the appropriate COM port from the combo-box. If you are unsure of the right COM port, you can look under the 'Ports (COM & LPT)' list in the Device Manager on Windows, or type "ls /dev/" on Mac. If you connected the board and ICD tool after you landed on the Measure page, you would want to hit 'Refresh' to update the list of COM ports.
3. Acknowledge the heat-warning. You will not be able to proceed without it.



 The default communication baud rate setting during Self-Commissioning measurement is 625000. However, the RS232 transceiver devices used in MCLV-2 and MCHV-2 development boards (i.e. MAX3232CUE) are specified to operate reliably at baud rates up to 120kbps. Due to this limitation, while using MCLV-2 development board or MCHV-2 development board during Self-Commissioning, if you observe serial communication related issues, then retry measurement after reducing the communication baud rate to a lower setting.

4. Verify that the Calibration Bus Voltage is correctly set for your hardware setup. **This step is necessary even if you are planning on skipping the Calibration process.** To update this value, edit the value in `Configure → Board → Voltage Source → Output`. For the dsPICDEM MCHV-2 and MCHV-3 Development boards, this value can be estimated from the nominal value of AC input voltage, as depicted in the table below:

Nominal AC input voltage (V_{RMS})	Value to be entered in <code>Configure → Board → Voltage Source → Output</code>
110	155.5
120	169.7
220	311.1
240	339.4

5. To calibrate, connect the Calibration board and click the "I have connected the calibration board" radio button. Hit "Calibrate".
6. To run the motor parameter measurements,
 - a. Disconnect the calibration board and connect the motor instead.
 - b. Click the "I have disconnected the calibration board" and "I have connected the motor" buttons.
 - c. Click the "Start" button to begin measurement. You will see the measured values appear for each parameter.
7. The measured values are updated in the 'Measured Values' of the Motor's Electrical and Mechanical Parameters section. Click on the  icon to set a measured value as the active value. The 'Use all' button can be used to set all measured values as active values. The 'Active Values' are used for Autotuning and code generation.

Common types of Self-Commissioning errors and reasons they can occur

General guidance: The usual cause of errors during Self-Commissioning is due to incorrect definition of motor and/or board specifications in the "Configure" page, specifically these parameters:

1. `Configure → PMSM Motor → Rated current: Continuous`
2. `Configure → PMSM Motor → Number of pole pairs`
3. `Configure → PMSM Motor → Nominal speed`
4. `Configure → Board → Voltage Source → Output`

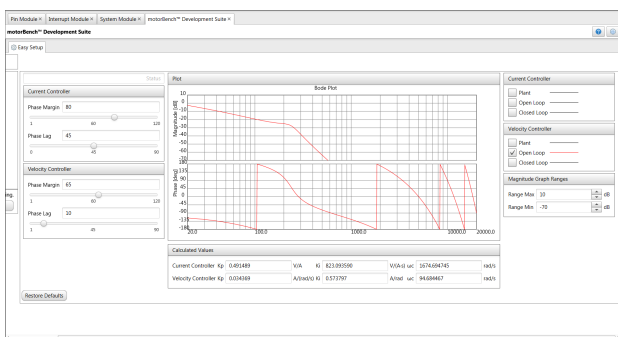
Table below provides some more specific examples of typical issues and their possible causes.

Symptom	Possible causes	Possible fix
---------	-----------------	--------------

Hardware over-current fault during R, Ld or Lq measurements (Fault Code #10)	Rated current specification of the motor is close to / higher than the full scale current range of the board. In this case, with motors that have relatively low values of electrical time constant, current ripple can trigger the over-current comparator on the board.	Retry the measurement after reducing the value in Configure → PMSM Motor → Rated current: Continuous to a lower value.
Motor stalls during inertia measurement with a Fault Code #107	The specified Nominal speed of the motor is either too low or too high for the given voltage source output at the inverter.	Correct the value in Configure → PMSM Motor → Nominal speed and retry the measurement.
Motor stalls during Ke measurement with a Fault Code #10, #102, #103, #104, #105 or #106	If the test motor has a relatively low value of stator inductance (typical Ld and Lq ~ 100µH or less), then transients in the current ripple can trigger the hardware over-current fault comparator causing the motor to stall during startup.	Increase the Measure → Motor Parameters → Current Controller → Phase Margin (slider) from its default value of 95° in 5° increments up to a maximum value of 115° and retry the measurement.
	If the test motor has a relatively high value of stator inductance (typical Ld and Lq ~ 1mH or more), then the current controller may need to be more aggressive during the startup phase in order to be able to suppress any open loop oscillations that may cause disturbances in the current control loop.	Decrease the Measure → Motor Parameters → Current Controller → Phase Margin (slider) from its default value of 95° to 91° and retry the measurement.
	If the test motor has significant inertia (typical J ~ 100µNm/(rad/s ²) or more) then, Measure → Motor Parameters → Approx. Spin Down Time parameter may be set to value that is too low for the given motor.	Increase the value that is set in Measure → Motor Parameters → Approx. Spin Down Time and retry the measurement. This will allow the measurement algorithm to wait for the motor to spin down completely before attempting to restart the motor in between measurements.
Measurement fails during R measurement with a message "Unable to detect a motor on the board inverter output. Please verify that the leads of the motor under test are connected securely to the board inverter output terminals."	All three phase wires of the test motor are not correctly connected to the inverter output terminals.	Verify that all three phase wires of the motor are connected correctly to the inverter output terminals.
	Test motor has a stator inductance value that is out of supported measurement range.	-
	Voltage source that is connected to the inverter stage is not powered on.	Verify if the inverter stage is connected to a voltage source that is powered on.

Autotuning

1. Click the 'Tune' button to go to the Tuning stage. There is no required user action in this stage. You can check out the Bode plots, or use different settings for phase margin or PI phase lag at crossover, if you wish to do so.
2. We recommend the defaults of 80/45 and 65/10 for current and velocity controllers, respectively.



Customize

The customize page allows users to modify advanced parameters used for MCAF code generation. This is an optional step for advanced users and does not need to be modified to generate code. The default values found in the customize page are good for most motors. Refer to the MCAF User's Guide for more detailed guidance.

Generate Code

1. To generate code, click "Generate" under the Project Resources section. Look for the generation progress under the MPLAB® Code Configurator console window. Once generation is complete, you will see a banner comment like so -

```

Notifications Output [M] Notifications [MCC] Pin Manager: Grid View
Project Loading Warning [M] MPLAB® Code Configurator [M]
16:34:59.647 INFO: .....
16:34:59.648 INFO: Generation complete (total time: 27260 milliseconds)
16:34:59.648 INFO: .....
16:34:59.648 INFO: Generation complete.

```

You will also be able to see the generated files added to your project.

2. If you want to run generation again, we suggest that before you do so, you remove the previously generated files. To do that, navigate to your project folder on your file system and remove the generated files under the "parameters" directory. This will avoid having to merge newly-generated files with earlier-generated files.
3. You are now ready to build and run the project, and spin the motor.

Managing Errors in Code Generation

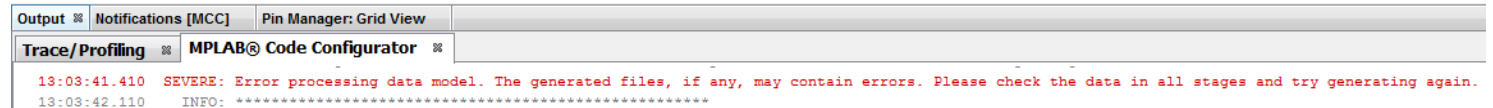
If motor parameters are extreme values, either because the Self-Commissioning process did not work correctly, or because the motor itself is unusual, code generation may fail.

Important things to know are

- How to recognize that an error has occurred
- How to report specific failure details to Microchip for assistance
- Common types of code generation errors and reasons they can occur

How to identify code generation errors

If an error has occurred during code generation, the MPLAB® Code Configurator console window will have a SEVERE category message like shown below -

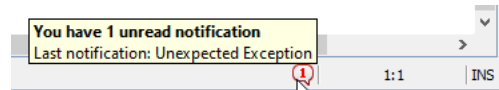


```
Output  Notifications [MCC]  Pin Manager: Grid View
Trace/Profiling  MPLAB® Code Configurator
13:03:41.410 SEVERE: Error processing data model. The generated files, if any, may contain errors. Please check the data in all stages and try generating again.
13:03:42.110 INFO: .....
```

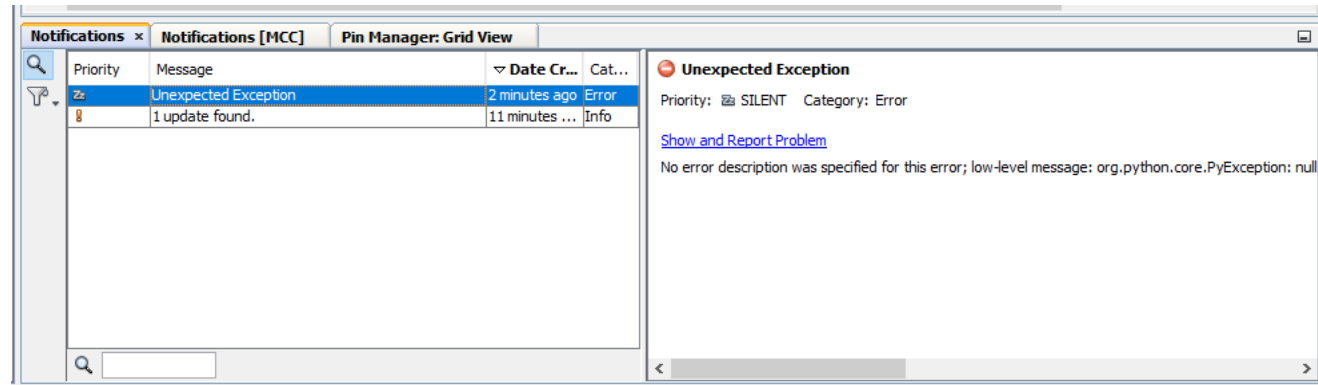
On building the code, the compiler throws the following warning -

```
"C:\Program Files (x86)\Microchip\xcl6\v1.27\bin\xcl6-gcc.exe" mcc_generated_files\pin_manager.c -o build/default/production/mcc_generated_files\pin_manager.o -c -mc
mcc_generated_files/motorBench/mcaf_main.c:48:2: warning: #warning "Unable to generate MCAF, refer to the motorBench log for detailed message regarding the failure."
mcc_generated_files/motorBench/mcaf_main.c:49:2: warning: #warning "Please refer to the Release Notes for motorBench(TM) Development Suite for additional guidance."
"C:\Program Files (x86)\Microchip\xcl6\v1.27\bin\xcl6-gcc.exe" -o dist/default/production/screenshots.X.production.elf build/default/production/mcc_generated_files/re
```

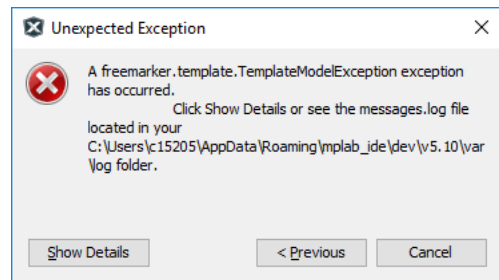
A notification will also be added to MPLAB X. In some cases a dialog box will popup, but in others it may be added silently to MPLAB X's Notifications pane. Keep on the lookout for a red error bubble in the lower right hand corner of the MPLAB X window:



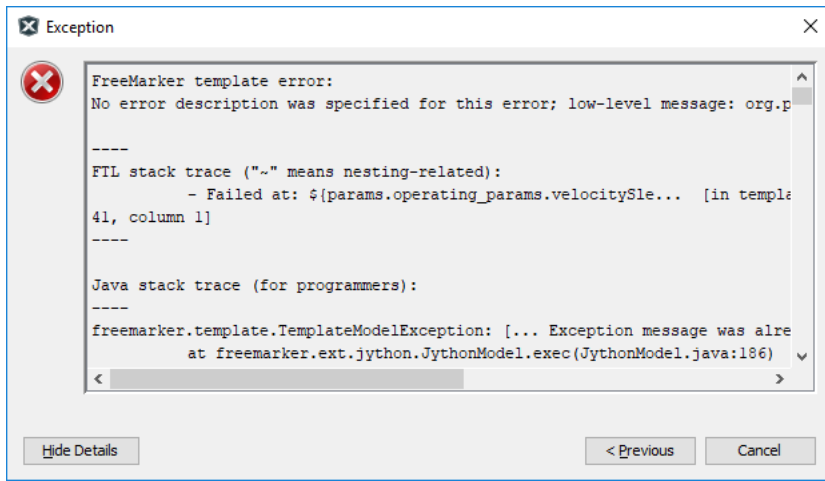
Click on the red bubble and the Notifications pane will display the notification in question:



Click on **Show and Report Problem** to display the error dialog box:



Click on **Show Details** to display the exception message:



How to report specific failure details to Microchip for assistance

If contacting Microchip staff for assistance, please copy the entire text in the dialog box (not a screenshot).

The most important portion of the error message will typically be a `ValueError` towards the end, or a `ZeroDivisionError`, for example:

- `ValueError: Value 'kwp' out of range: 1.69777e+06 Q8.8 = 26101.3 A/(rad/s)`
- `ValueError: Value 'velocitySlewratelimitDecel' out of range: 4.67089e+06 Q1.15 = 2.67361e+09 rad/s`
- `ZeroDivisionError: float division`

This provides important clues that may indicate the cause of the problem and how to address it.

Common types of code generation errors and reasons they can occur

General guidance: The usual cause of code generation errors is a motor parameter that is extremely high or extremely low. Please see the section "Motor Control Limitations" which discusses ranges of motor parameters that work well with motorBench® Development Suite. There are a few reasons a motor parameter could cause a code generation error:

- The parameter has not been measured or entered correctly. (For example, a datasheet states 50µH but is entered as 50mH = 0.05H.)
- The motor has unusual motor parameters. (Slotless motors may fall into this category; they have very low inductance.)
- The motor is not well-matched to the motor drive circuitry, for example use of a 6V motor with a 24V motor drive.

More likely values that are out of range are shown in the table below.

Symptom	Possible cause	Possible fix
<code>ZeroDivisionError</code>	One of the motor parameters is zero. (This should never be the case.)	Make sure none of the motor parameters are zero.
<code>kwp</code> out of range	This is the velocity loop proportional gain. Out-of-range errors can occur if the inertia (J) is very high.	Make sure the inertia value is reasonable. An increase or decrease in velocity loop phase margin in the Autotuning page may be required for high-inertia motors. (Increasing phase margin generally lowers <code>kwp</code> whereas decreasing phase margin raises it.)
<code>velocitySlewratelimitDecel</code> out of range	May occur if Coulomb friction torque (Tfr) is out of range.	Make sure the Coulomb friction value is reasonable.
<code>normLsdt</code> out of range	May occur if inductance (Ld and/or Lq) is out of range.	Make sure the motor inductance values are reasonable and well-matched to the motor drive.
<code>normRs</code> out of range	May occur if resistance (Rs) is out of range.	Make sure the motor resistance values are reasonable and well-matched to the motor drive.

Building Code

At this point, work in motorBench® Development Suite is complete. Switch to using MPLAB X, and click the Run button to build and program the device.

⚠ Reminder: the Motor Control Application Framework [requires certain compiler and linker settings](#); without these settings it may not run correctly.

ⓘ Warning during build
Please Note: on building the code the compiler throws the following warning. It is safe to ignore this warning as it does not cause any functional issues.

```

"C:\Program Files (x86)\Microchip\xc16\v1.36\bin\xc16-gcc.exe" mcc_generated_files/motorBench/mcaf_main.c -o build/default/p
"C:\Program Files (x86)\Microchip\xc16\v1.36\bin\xc16-gcc.exe" mcc_generated_files/motorBench/monitor.c -o build/default/pro
mcc_generated_files/motorBench/diagnostics.c: In function 'X2CScope_isSendReady':
mcc_generated_files/motorBench/diagnostics.c:117:5: warning: 'UART1_StatusGet' is deprecated or unsupported (declared at mcc_ge
"C:\Program Files (x86)\Microchip\xc16\v1.36\bin\xc16-gcc.exe" mcc_generated_files/motorBench/ui.c -o build/default/producti
```

Running the Application Framework

Directions for using MCAF with MCLV-2 and MCHV2/MCHV-3 are slightly different:

MCLV-2

- Press button S2 to start/stop the motor
- Press button S3 to reverse direction
- Turn the potentiometer to control speed
- In the event of an error, both LEDs (D2 and D17) will flash together to indicate an error code; see the Motor Control Application Framework documentation on the [motorBench® Development Suite webpage](#) for further information.

MCHV-2 and MCHV-3

- Press the button labeled PUSHBUTTON to start/stop the motor
- While the motor is running, long press the button PUSHBUTTON (minimum 3 seconds) to reverse the direction of rotation.
- Turn the potentiometer to vary speed.
- In the event of an error, both LEDs (D17 and D19) will flash together to indicate an error code; see the Motor Control Application Framework documentation on the [motorBench® Development Suite webpage](#) for further information.

Real-Time Diagnostics

The Motor Control Application Framework includes out-of-the-box support for [X2C-Scope](#), a third-party plugin for MPLAB X which facilitates real-time diagnostics. X2C-Scope is available in the same **Available Plugins** tab used to install motorBench® Development Suite.

Repairs

Motor Control Fixed Issues

Changes since revision 2.0

The following aspects of motorBench® Development Suite and the Motor Control Application Framework (MCAF) have been updated:

- MCAF has been updated from R1 to R4, including
 - Changes in R2:
 - Support for DC link compensation
 - Support for overmodulation
 - Support for wider range of low-voltage motors
 - Updated HAL for future MCHV2 support
 - Updated Motor Control Library
 - Numerous minor fixes
 - Changes in R3:
 - MCC system module compatibility
 - MCHV-2 and MCHV-3 support
 - Inverter maximum current now has a 1:1 ratio with the maximum commanded dq-frame current of the drive, operating in FOC (in R2 this incorporated a derating factor)
 - Other minor fixes
 - Changes in R4:
 - MCC peripheral support
 - Parameter customization
 - Quadrature encoder support
 - Added new startup method (Weathervane startup)
 - Other minor fixes
- Sections in this release notes affected:
 - Other Requirements
 - Limitations
 - Supported Motor Parameters

Known Issues

Issue Key	Summary	Workaround
MBPLAN-932	Exception during attempted creation of a runtime properties class	No workaround needed, this issue doesn't have an impact on the functionality.
MBPLAN-1095	Switching projects after loading motorBench erroneously allows motorBench code to generate for new project	<ul style="list-style-type: none">• Close MCC in first project• Set new project as Main project and open MCC
MBPLAN-673	Serial port does not get closed programmatically when MCC exits during self-commissioning	If you exit SC during execution, restart MPLAB X.
MBPLAN-1056	Can not generate code and Customize UI is missing	<ul style="list-style-type: none">• Open the directory <code>c:\Users\%USERNAME%\AppData\Local\Temp\mplab_ide\</code> in File Explorer• Remove pmsm-foc-framework.zip• Close MCC and reopen.
MBPLAN-1098	Deprecated warning, when compiling MCAF	No workaround needed. No impact on functionality
MBPLAN-1099	The units are in rad/s instead of rpm	
MBPLAN-1100	motorBench Development Suite should be used with (R) symbol, but at some places still (TM) is used	

Motor Control Issues

Issue Key	Summary
DB_MC-411	Current calibration happens only once (at part reset) rather than upon entry to MCSM_RESET state
DB_MC-560	Speed controller exhibits chattering behavior at voltage saturation hysteresis boundary (MCAF)
DB_MC-978	"Soft start" gate drive in board_service.c has duty cycle that is too small
DB_MC-1092	PLL estimator may not converge into rotor reference frame while using the Classic startup method in MCAF
DB_MC-1167	motor.idqCmdRaw doesn't interact properly with test harness
DB_MC-1396	PLL calculations in code generation do not allow motor.velocity.nominal to be more than 1250Hz electrical (=20kHz/8/2)
DB_MC-1415	With some motors and 12V operation, increased velocity margin improves startup but creates unstable estimator
DB_MC-1430	Quantum MT4012 unstable in closed-loop operation at 4200 RPM speed and above
DB_MC-1489	Hurst 300 motor Stall detection shows false positives for TC3 (12V testing)
DB_MC-1491	With Quantum MT4012, MCAF may not detect stall
DB_MC-1492	Quantum MT4012 Stalls on pressing 'S3'(reverse) at low speeds and on changes to speed command potentiometer
DB_MC-1495	Anaheim BLY342D-24V-3000, BLY342D-48V-3200 motors creates hardware over-current during stall-detect testing
DB_MC-1521	Closed loop speed step response overshoot - MCHV2, Leadshine 400
DB_MC-1892	Some motors with extreme parameters may produce out-of-range error for stall_detect.group.timerCountsVarianceDetect (detected in Monte Carlo analysis)
DB_MC-1920	Board service isrCount-based timing is not guaranteed
DB_MC-1922	LED patterns not displayed when in the TEST_DISABLE or TEST_ENABLE states
DB_MC-2005	Default value of startup current needs to be reduced from 91% to 75%
DB_MC-2122	BLWS232D motor startup in QEI mode causes a false detect for stall-detection
DB_MC-2212	mcaf_traps.c depends on MCC DMA Module (it should not)
DB_MC-2213	Deadtime needs to be changed in both MCC and motorBench to affect code
DB_MC-2215	Use of incorrect deprecated API for MCC UART driver (produces warning during build)
DB_MC-2275	Large current rampup times may not start (STARTUP_TORQUE_RAMPUP_RATE = 0)
DB_MC-2306	Incorrect comment in test_harness.h describes overrideCommutationOnOff.period rather than overrideCommutationOnOff.maxCount
DB_MC-2309	QEI tracking loop Kp and Ki produce out-of-range errors for low-speed motors

Troubleshooting Issues with some motors

We have observed some issues during our testing under the following circumstances:

1. Starting up small motors with large inertia loads
2. Starting up certain motors with low speed reference
3. Instability in current loop

These are some solutions that may work for some motors:

1. Starting up small motors with large inertia loads using the classic startup method may have a difficult time with the transition from forced commutation to closed-loop commutation. Use the weathervane startup method in these circumstances.
2. Starting up certain motors with low speed reference fails to start. This may be due to either an issue in the startup method or the behavior of the sensorless estimator during the transition to closed-loop commutation. Workarounds for this issue would be to use the weathervane startup method or to ensure that the speed command is not at the low end of its range.
3. If instability in current loop is observed an increase of 5° or 10° in current loop phase margin may help.

Supported Device

1. dsPIC33EP256MC506 <https://www.microchip.com/wwwproducts/en/dsPIC33EP256MC506>

Software Limitations

motorBench® Development Suite is tested for serial communication using Windows 7 and Windows 10 platforms. Other platforms may work with standard baud rates, but this operation has not yet been verified.

Motor Control Limitations

Following are the known limitations for this release of motorBench® Development Suite:

1. **One mechanical load** - constant load. This represents a mechanical load with constant inertia, viscous damping, and friction. The velocity control loop can generally reject external disturbance torques, within the rated current of the motor and board, and within the bandwidth of the velocity control loop. Mechanical loads with time-varying or angle-varying inertia, viscous damping, and friction, such as a blower, compressor, or pump, are currently not supported.
2. **One motor type** - PMSM

3. **Self commissioning does not support modifications to MCLV-2, MCHV-2, and MCHV-3 that affect current sensing, voltage sensing, or the power stage.**

Use of Self-Commissioning with MCLV-2 at DC link voltages other than 24V is supported (note that jumper J6 may need to be removed).

MCLV-2:

The reference motor is the Nidec Hurst motor DMA0204024B101 (MicrochipDirect part number [AC300022](#)). Microchip has also validated motorBench® Development Suite (including Self-Commissioning) with motors with parameters plotted below. Please also read the following section on Supported Motor Parameters. If motorBench® Development Suite is unable to spin a motor successfully, please contact Microchip staff for additional assistance.

	units	min	max	plot
Resistance (R)	ohms (line-line)	0.1714	5.664	
Inductance, q-axis (Lq)	mH (line-line)	0.02982	5.269	
Inductance, d-axis (Ld)	mH (line-line)	0.03576	5.172	
Back-emf (Ke)	Vrms/KRPM (line-line)	0.8808	6.49	
Friction (Tf)	mNm	0.556	54.03	
Viscous damping (B)	uNm/(rad/s)	1.012	196.9	
Inertia (J)	uNm/(rad/s²)	0.362	1162	
Electrical time constant L/R	ms	0.1913	3.889	
Mechanical time constant 2/3 JR/Ke²	ms	2.27	47.25	

(Note: Mechanical time constant (2/3)×JR/Ke² represents the time constant of velocity acceleration under an open-loop synchronous-frame voltage step, neglecting the effects of inductance, with J, R, and Ke expressed in canonical metric units. R is expressed as line-neutral resistance = half of line-line resistance, and Ke is expressed as V/(rad/s) line-neutral zero-peak = Vrms/KRPM (line-line) × 0.007796968)

MCHV-2/MCHV-3:

The reference motor is the Leadshine 400W motor EL5-M0400-1-24 (MicrochipDirect part number: [AC300025](#)). Microchip has validated motorBench® Development Suite (including Self-Commissioning) with motors with parameters plotted below. Please also read the following section on Supported Motor Parameters. If motorBench® Development Suite is unable to spin a motor successfully, please contact Microchip staff for additional assistance.

	units	min	max	plot
Resistance (R)	ohms (line-line)	1.514	13.02	
Inductance, q-axis (Lq)	mH (line-line)	3.474	42.19	
Inductance, d-axis (Ld)	mH (line-line)	3.176	37.47	
Back-emf (Ke)	Vrms/KRPM (line-line)	21.2	39.23	
Friction (Tf)	mNm	7.812	66.53	
Viscous damping (B)	uNm/(rad/s)	17.91	108.4	
Inertia (J)	uNm/(rad/s²)	16.21	240	
Electrical time constant L/R	ms	0.9745	3.059	
Mechanical time constant 2/3 JR/Ke²	ms	0.8905	2.797	

4. **Board**

- a. **dsPICDEM™ MCLV-2 development board.** This release of motorBench® Development Suite is compatible with modifications to the board to alter its rated current or voltage. Contact your local Microchip office to obtain the document "Using MCLV-2 with motorBench® Development Suite to support alternative current and/or voltage ratings", which provides guidance for such modifications. Other modifications may not be compatible.

Self-Commissioning must be run on an MCLV-2 with unmodified current sense, voltage sense, and PWM circuitry. Motor parameters determined with Self-Commissioning may then be applied for use with modified MCLV-2 boards by manually overriding the values in the Motor Parameters section on the Measure page.

- b. **dsPICDEM™ MCHV-2 and MCHV-3 development boards.** This release of motorBench® Development Suite is compatible with unmodified MCHV-2 and MCHV-3 development boards.

- 5. **Motors should be well-matched to the board and operating voltage.** The nominal DC link voltage of the MCLV-2 board is 24V. This voltage can be changed by cutting jumper J6 and using an appropriate power supply connected to the appropriate terminals of J7. Use of a mismatched motor (for example, a 12V motor used with a 24V DC link voltage) may cause a hardware overcurrent fault; in this case Self-Commissioning may fail with the message "Fault Code #10: Undefined Fault". Retry with an appropriate DC link voltage.
- 6. **One PIM and one device** - dsPIC33EP256MC506 External OpAmp PIM with silicon revision A8 or later. (Please see the Hardware Setup section above for important modifications to the PIM for use in MCHV-2 and MCHV-3.)
- 7. **One algorithm** - FOC
- 8. **Two estimators** - PLL, QEI
- 9. **Self-commissioning:**
 - a. **Performance criteria adjustment is not presently supported.** This includes adjustment of phase margin and PI phase lag at crossover in the current loop; Microchip has not completed validation and documentation of these adjustments.
- 10. **Autotuning:**
 - a. **Performance criteria adjustment of the current loop is not presently supported.** This includes adjustment of phase margin and PI phase lag at crossover; Microchip has not completed validation and documentation of these adjustments.
 - b. **Use of performance criteria adjustment of the velocity loop is not fully documented or tested.** We recommend not adjusting phase margin or PI phase lag unless necessary; cases where this is likely to occur are large inertias where $\alpha_J = JR/LK_m^2 > 10$, for which an increase of phase margin is appropriate. Phase margin values between 70 and 85 degrees are recommended in this case, with larger values providing additional stability at the cost of lower velocity bandwidth.
- 11. **Axis management not currently implemented** - supports only one axis.
- 12. **Code generation:**
 - a. **Some board parameters are not taken into consideration.** Some of the important ones that fall under this category are
 - i. Processor clock
 - ii. PWM switching frequency
 - b. **Integration with external user-supplied code may involve substantial changes.** Some guidelines for this are given in the documentation for the Motor Control Application Framework. While it is possible to integrate the code generated from motorBench® Development Suite with external code, it is the responsibility of the end user to validate this combination.
- 13. **Required compiler settings:**
 - a. **Optimization**
 - -O1 or greater; -O0 and -Os will both compile without errors but do not execute fast enough to complete within the 50 microsecond ADC ISR. Note: at higher optimization levels, in-circuit debugging using MPLAB X will behave unreliably with respect to breakpoints and single-stepping through C code.
 - The "Omit frame pointer" and "Unroll loops" settings must be enabled.
 - b. **Memory model:**
 - Large data model (handles using pointers, not direct addressing, to allow for more than 8K of program variables)
 - Small scalar model
 - c. **Additional options:** -Wno-volatile-register-var -finline
 - d. **Test harness:** In order for the test harness to be enabled, the symbols MCAF_TEST_PROFILING and MCAF_TEST_HARNESS should be defined.

Supported Motor Parameters

Changes since revision 2.0

This release of motorBench® Development Suite expands the range of supported motors, subject to the following notes:

- **Ranges of motor parameters** (including rated values and computed metrics) **must be within the limits noted in either range-limits-mclv2.html or range-limits-mchv2.html**. These ranges were tested to ensure that code generation produced firmware constants that were within bounds.
- **Self-Commissioning does not need to complete successfully but valid motor parameters are required**. Some motors may have too low of an inductance or resistance, and may fail Self-Commissioning.
- **Other particular issues that may cause incompatibility with motorBench® Development Suite include**
 - **Large inertia values** – in this case, increasing voltage loop phase margin may prevent stability problems. (See "Autotuning" in the Limitations section of this document.)
 - **Rotor magnetic saliency** – Ld and Lq should be within 10% of each other. Higher mismatch between Ld and Lq is typically found in interior-permanent magnet (IPM) motors, and is an intentional feature of the design.
 - **Large back-EMF harmonics** – a quasi-sinusoidal back-emf is assumed
 - **Issues involving individual motor control algorithms**, such as PLL estimator, motor startup, or stall detection
 - **High cogging torque**
 - **Mismatch between motor and drive** (namely using a motor with current and/or voltage requirements significantly different from that of the hardware)
- **Microchip cannot guarantee that motorBench® Development Suite will work correctly with all motors**. If a particular motor does not work properly, please contact the MCU16 Motor Control Team for further guidance.

Customer Support

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com>. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- Product Support – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- General Technical Support – Frequently Asked Questions (FAQs), technical support requests, online discussion groups/forums (<http://forum.microchip.com>), Microchip consultant program member listing
- Business of Microchip – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Additional Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineering (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is available on our web site.

Technical support is available through the web site at: <http://support.microchip.com>