Release Notes for motorBench™ Development Suite v1.15

Published date: 01 Mar 2017

Table of Contents

Introduction

motorBench™ Development Suite, a plugin for MPLAB X IDE, is designed to help motor control engineers to design and implement motor control systems, from very basic to very sophisticated ones. This plugin allows the user to:

- Configure a motor system
- Measure motor parameters
- Tune the controller gains
- Generate code to spin the motor

Visit the motorBench™ Development Suite webpage for the latest information, including documentation on the Motor Control Application Framework.

Note:
Information listed in this release notes supersedes information provided in any other form of documentation that is included with motorBench™ Development Suite.

System Requirements

Minimum system configuration

MPLAB X system requirements are listed at http://microchip.wikidot.com/mplabx:requirements
Supported OS versions: Microsoft ® Windows ® 7 and Microsoft® Windows® 10 .

Other requirements

- MPLAB X IDE v3.55
- XC16 v1.26

Supported Hardware

This release of motorBench™ Development Suite supports/works with the following hardware:

1. dsPICDEM MCLV-2 Development Board (microchipDirect part# DM330021-2) — as delivered by Microchip
2. dsPIC33EP256MC506 External OpAmp Motor Control PIM (microchipDirect part# MA330031-2) with silicon revision A8 or later
3. Hurst 24V BLDC motor DMA0204024B101 (microchipDirect part# AC300022)
4. 24V power supply (microchipDirect part# AC002013 or an equivalent model)
5. A USB-to-logic-level-UART converter from the following list:
    1. Saelig USB-COM-U or USB-COM-U13

    2. TRENDnet TU-S9 v2.0
6. Programming tool - one of the following tools: Real ICE (microchipDirect part# DV244005), ICD3 (microchipDirect part# DV164035), PICkit 3 (microchipDirect part# PG164130)
7. Optional: Precision resistors for Self-Commissioning calibration process - see help files for more details
    1. Recommended: 10Ω, 1% tolerance and rated to handle an average pulse load of ≥10.2 W for a 2 ms pulse train (i.e. energy = 20.4 mJ); For example: Vishay MBB02070C1009FCT00 (10Ω, ±1%, 0.6W)
    2. Valid range: 3.4Ω to 13.3Ω

Note:
Certain power supply models with three-prong AC input internally connect the AC ground signal to their 24V DC (-) output. This can cause ground loop issues on certain hardware setups. In order to avoid this issue, we recommend using a 24V power supply model that has a two-prong AC input.

Installation Procedure

Basic steps of installing motorBench™ Development Suite plugin are given here.

To install motorBench™ Development Suite plugin:

1. In MPLAB X IDE, select **Plugins** from the **Tools** menu
2. Select the **Available Plugins** tab
3. Check the box for **motorBench™ Development Suite**, and click on **Install**

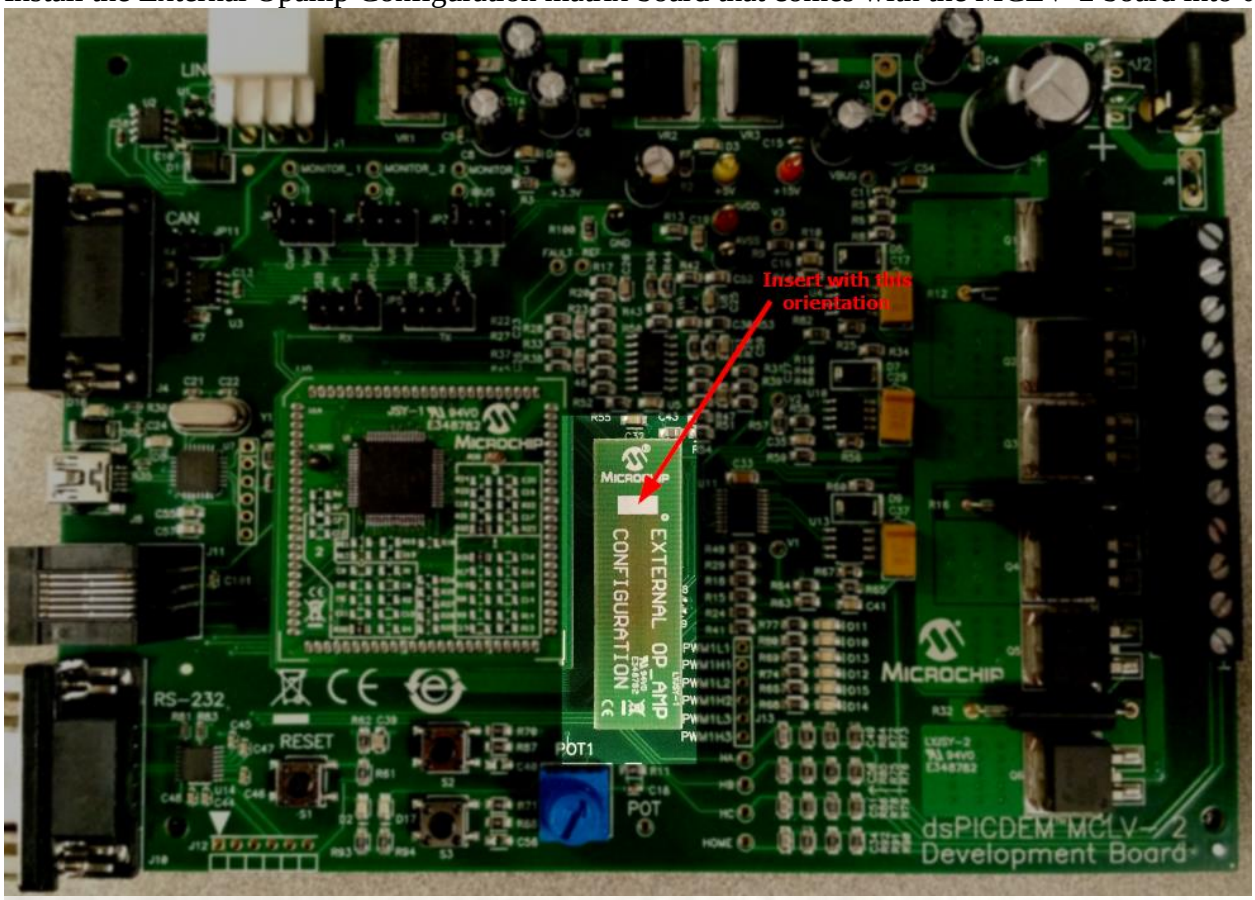
Once the plugin is installed, proceed to the motorBench™ Development Suite webpage to obtain a sample MPLAB X project (sample-motorBench-project.X.zip) that is needed to start using this plugin.

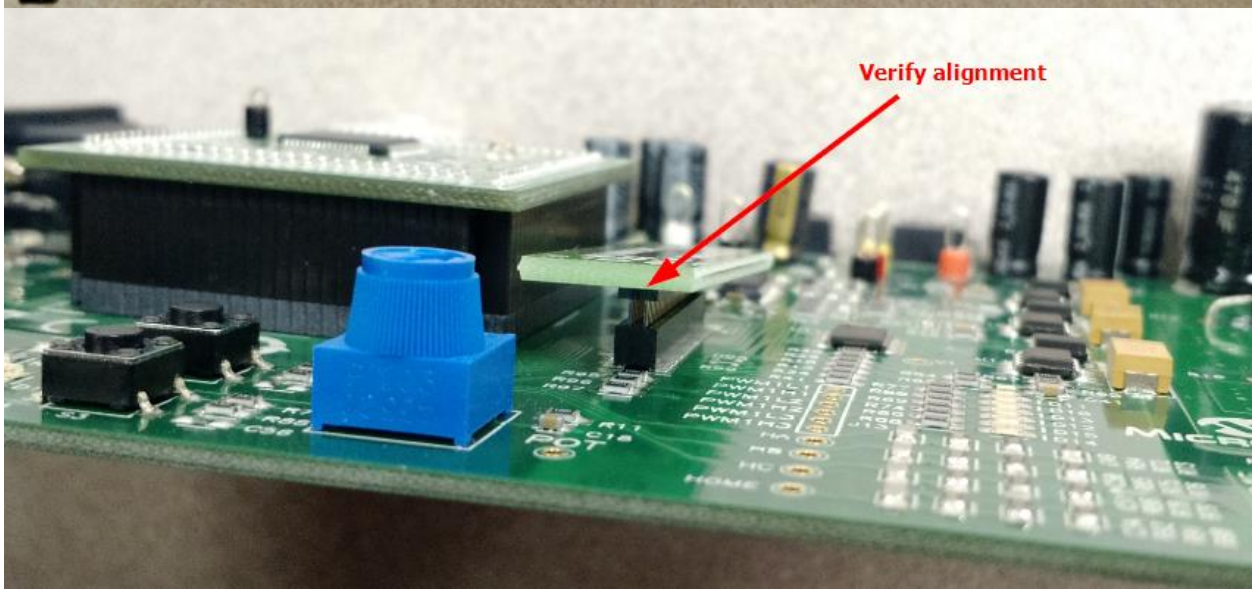
Hardware Setup

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

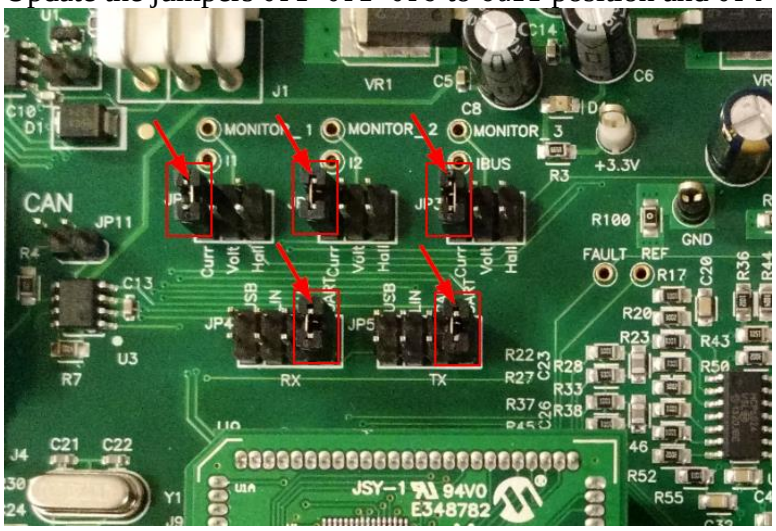
1. MCLV-2 board comes pre-installed out of the box with dsPIC33EP256MC506 Internal Opamp Motor Control PIM (MA330031); replace this PIM with the dsPIC33EP256MC506 External Opamp PIM (MA330031-2) specified above.
2. Make sure that the dsPIC33EP256MC506 External Opamp PIM is populated with a silicon mask rev-A8 or later. To verify this, read out the device revision from MPLAB X IDE 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

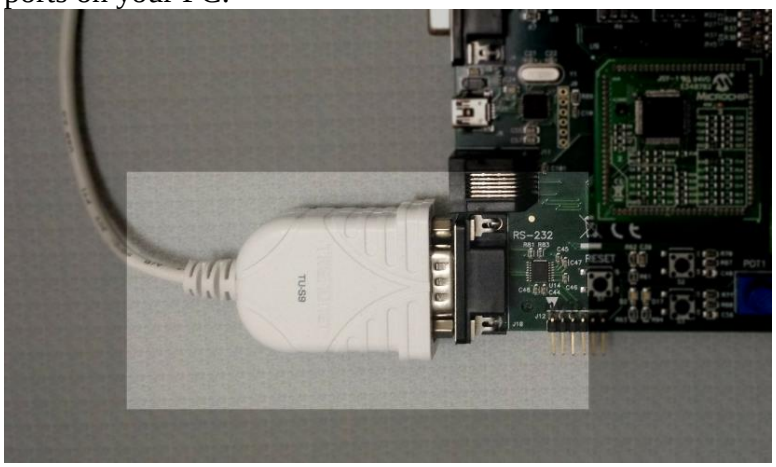




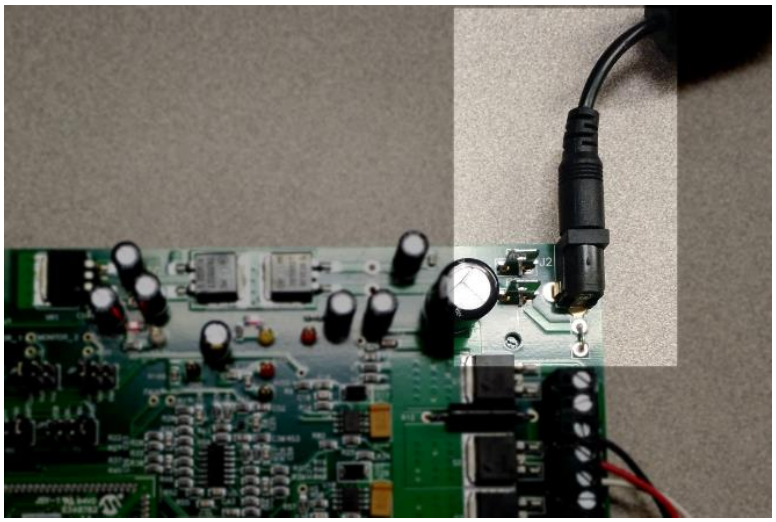
4. Update the jumpers `JP1-JP2-JP3` to `Curr` position and `JP4-JP5` to `UART` position as shown below:



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:

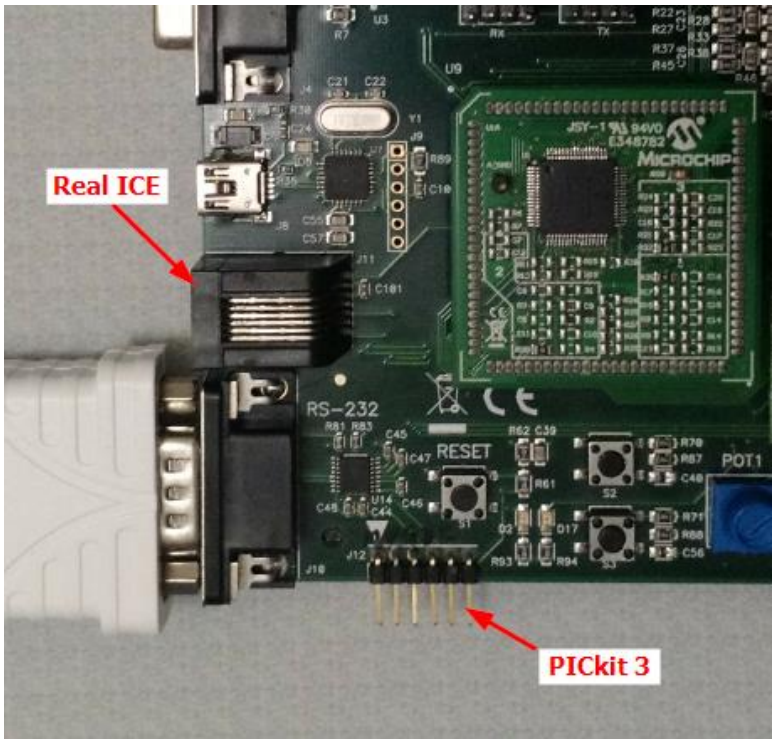


6. Plug in the 24V power supply barrel connector into `J2` of MCLV-2 board and then plug in the power supply into an AC mains outlet:

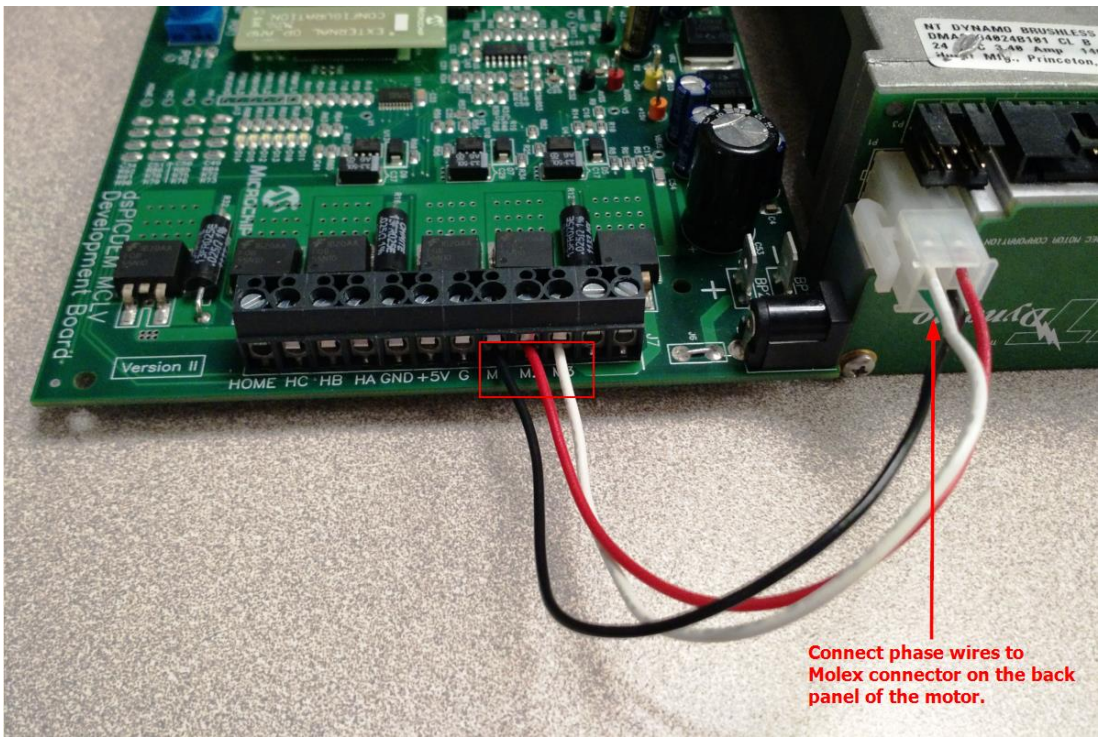


7. Connect your Real ICE / ICD3 / PICkit3 to one of the USB ports on your PC

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 process is running.
11. During Self-Commissioning Analysis process if, for whatever reason, you need to reset the firmware that runs on the MCLV-2 board, use push button `S1` for this purpose.

Limitations

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

1. **One mechanical load** - constant load, i.e. motor with no external mechanical load or torque load that does not change over time or velocity. Mechanical loads like blower, compressor, pump, etc. are currently not supported
2. **One motor type** - PMSM
3. **One motor element** - Hurst300 motor (DMA0204024B101) — Microchip has also validated motorBench™ Development Suite using additional motors with parameters plotted below:

	units	min	max	plot
Rs	ohms (line-neutral)	0.0701	2.7532	
Lsq	mH	0.0244	2.4738	
Lsd	mH	0.0235	2.3206	
Ke	Vrms/KRPM (line-line)	0.305329	2.309976	
Friction	Nm	0.000644	0.0286	
Viscous Damping	uNm/rad/s	0.7319	144.2	
Inertia	uNm/rad/s^2	0.3247	1180.0	
Electrical time constant	ms	0.298257	4.18117	
Mechanical time constant	s	0.44364	10.865562	
				10^{-4} 10^{-3} 10^{-2} 10^{-1} 10^{0} 10^{1} 10^{2} 10^{3}

4. **One board** - dsPICDEM™ MCLV-2 development board (as delivered by Microchip)
5. **One PIM and one device** - dsPIC33EP256MC506 External Opamp PIM with silicon revision A8 or later
6. **One algorithm** - FOC
7. **One estimator** - PLL
8. **Self-commissioning:**
    1. **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.
9. **Autotuning:**
    1. **High inertia loads (more than 10× the motor inertia) are not presently supported.** In certain cases these have been observed to cause instability in the velocity control loop.
    2. **Performance criteria adjustment is not presently supported.** This includes adjustment of phase margin and PI phase lag at crossover in the current and velocity loops; Microchip has not completed validation and documentation of these adjustments.
10. **Axis management not currently implemented** - supports only one axis.
11. **Code generation:**

1. **Most board parameters are not taken into consideration.** Some of the important ones that fall under this category are
    1. Current Sensor full-scale value
    2. Processor clock
    3. PWM switching frequency and PWM dead time
    4. Minimum and maximum DC link voltage
2. **Voltage supply (Vbus) is not adjustable** - we require the use of a 24V supply
3. **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.

12. **Required compiler settings**:
    1. **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" setting must be enabled.
    2. **Memory model:**
        - Large data model (handles using pointers, not direct addressing, to allow for more than 8K of program variables)
        - Small scalar model
    3. **Additional options:** `-Wno-volatile-register-var -finline -mno-override-inline`

Operating Instructions

This section lists the operating instructions for specific sections of motorBench™ Development Suite.

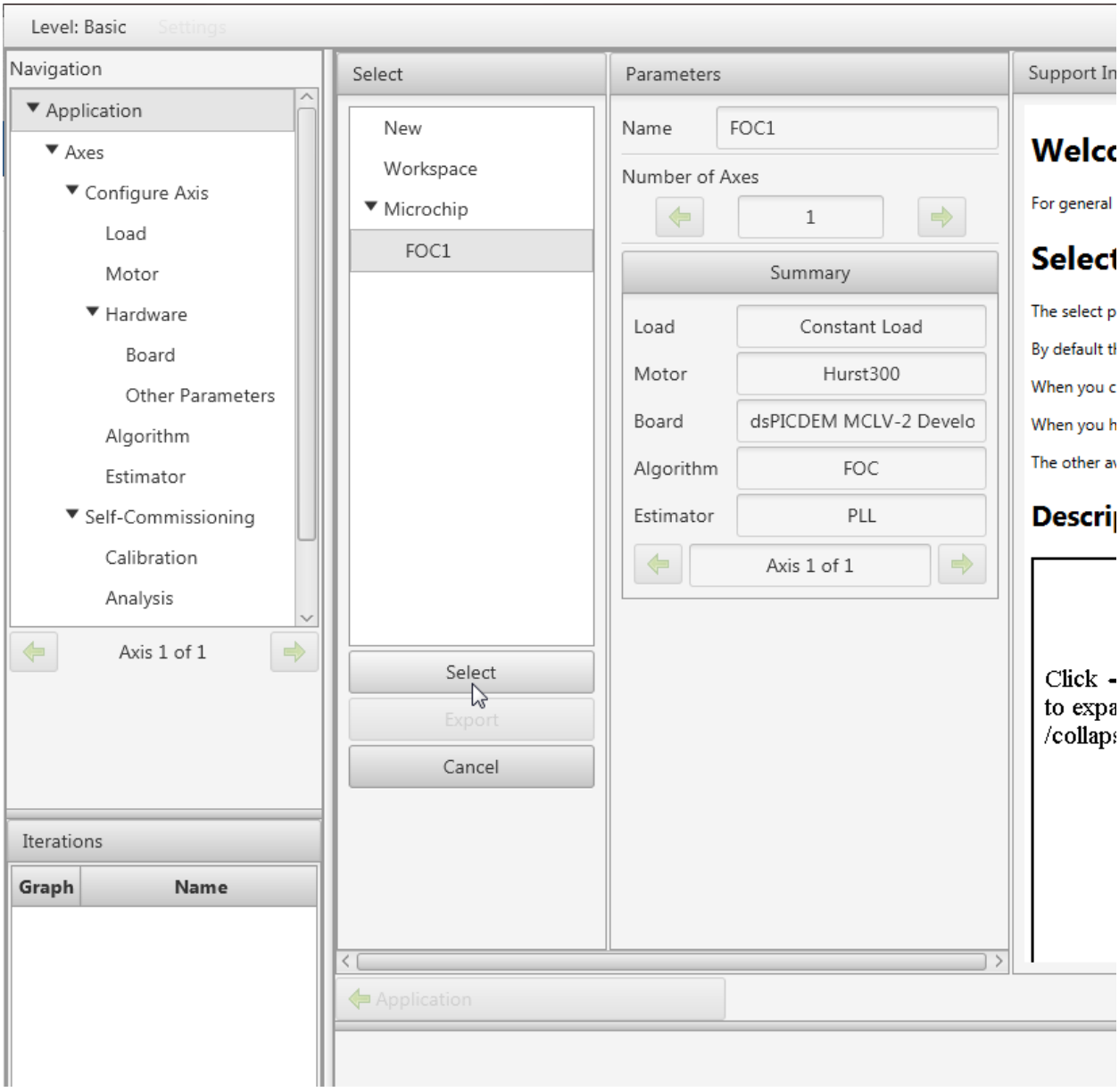
Getting started

- Visit the motorBench™ Development Suite webpage to obtain a copy of the "sample-motorBench-project.X.zip" file. This archive file includes an empty sample MPLAB X project with compiler and linker settings that have been preset for use with the Motor Control Application Framework.

  Although motorBench™ Development Suite can be used with any MPLAB X project, the Motor Control Application Framework requires the compiler and linker settings described above.
- Unzip this file onto your computer
- Open the "sample-motorBench-project.X.zip" in MPLAB X
- Right-click and set as main project in MPLAB X
- Follow the steps described in Hardware Setup section of these release notes to setup your hardware correctly
- Right-click and open project properties. Select an in-circuit programmer and XC16 compiler.
- Open motorBench™ Development Suite (Tools → Embedded → motorBench™ Development Suite or using the toolbar icon)

Configuration

- Use the predefined "FOC1" application rather than a custom application; this will work correctly with the hardware mentioned above. To use the predefined "FOC1" application in motorBench™ Development Suite, in the Application panel, select Microchip → FOC1 and click the Select button.



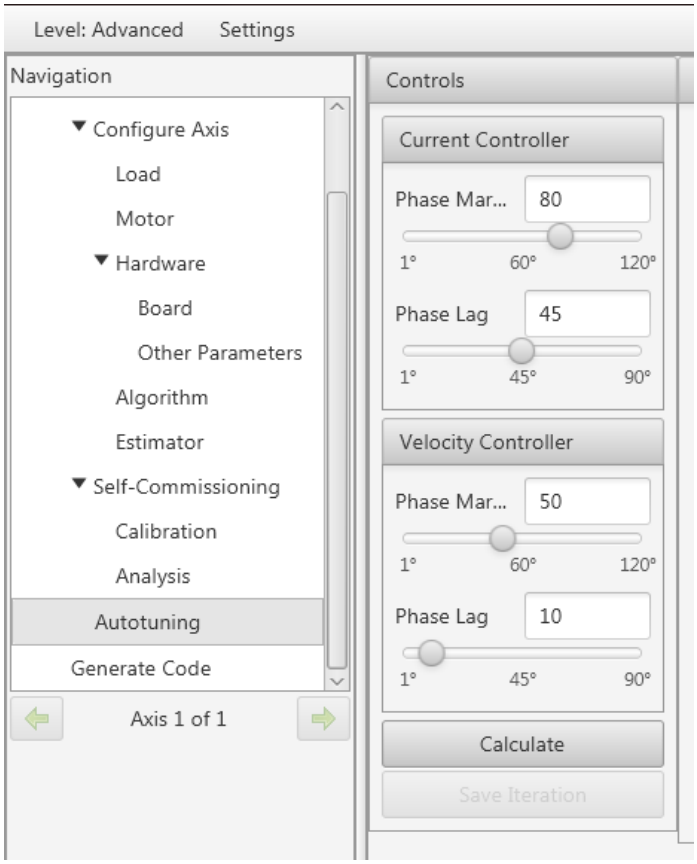
Self-Commissioning

- In the "Analysis" page, default values of the sliders (for Current Controller phase margin and PI phase at crossover) have been chosen to optimally work with a wide range of motors. **Adjustment of the sliders is not necessary for the Hurst DMA0204024B101 motor, so please leave these at their default values.**

- In case of an emergency at any point of time during the "Analysis" process, press button S1 (RESET) on the MCLV-2 board - this will cause the inverter stage to turn off its drive voltage and the motor will coast down (if it was spinning). This will automatically stop the "Analysis" process.
- In the "Analysis" page, the parameter "Approx. Motor Spin Down Time" is an estimate of the typical time it takes for a motor to spin down. In case of the Microchip-provided "Hurst300" motor, it is recommended to specify a spin down time of **1.0s**.

Autotuning

- In Advanced mode, there are sliders to modify the default values of phase margin and PI phase lag at crossover, for both the current and velocity controllers. **Adjustment of the sliders is not necessary for the Hurst DMA0204024B101 motor, so please leave these at their default values.** (from top to bottom: 80, 45, 50, 10). These values were chosen to optimize the disturbance rejection behavior over a wide range of motors. Adjustments are necessary only in some cases for high-inertia motors.



- Click on the "Calculate" button (this will do the actual autotuning calculations and display some Bode plots for frequency response) and then the "Save Iteration" button before going to the next page ("Generate Code")

Generate Code

The Generate Code page in motorBench™ Development Suite will generate C code from the Motor Control Application Framework into the MPLAB X project. (For more information on the Motor Control Application Framework, see the MCAF documentation published on the motorBench™ Development Suite webpage.)

- Click "Select Files", leave them all selected, and click OK
- Click "Generate Code" – this will generate C files and add them to the MPLAB X project

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.

Running the Application Framework

- 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 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.

Known Issues

This section lists the known issues and their workarounds (when available) for this release.

List of Known Issues

Key	Issue Summary
MCGUI-1384	The Motor Control Application Framework's configurable parameters generated for the sensorless estimator may cause certain motors to lose lock and eventually stall during phases of rapid acceleration or deceleration. This issue does not affect the "Hurst300" motor (DMA0204024B101) that is supported in this release of motorBench™ Development Suite.
MCGUI-1364	Under rare circumstances, Self-Commissioning Analysis or Calibration process may terminate with a "Serial Communication Timeout" issue, despite using one of the supported USB-to-logic-level-UART converter cables. To manually resolve this issue, restart the process by pressing the "play" button again.
MCGUI-1345	While running the Self-Commissioning Analysis process, under certain conditions, the user input for "Approximate Motor Spin Down Time (s)" in the Controls panel may be incorrectly used as a zero value. If this happens while the motor is spinning and if the motor has a relatively large inertia, then it may result in a hardware over-current fault. To recover from this fault, please restart the Self-Commissioning Analysis process.
MCGUI-1329	The name of MPLAB X main project that is listed in the title bar of motorBench™ Development Suite may get out of sync. To avoid this issue after switching the main project, restart MPLAB X IDE before opening the motorBench™ Development Suite plugin.
MCGUI-1304	After generating files from motorBench™ Development Suite, the MPLAB X IDE "package a project" feature will not automatically include the required library files that were generated by motorBench™ Development Suite. These library files have to be manually copied into the packaged project folder.
MCGUI-1247	Self-Commissioning Analysis process fails to complete if the motor's mechanical time constant is greater than 1.63835 seconds.

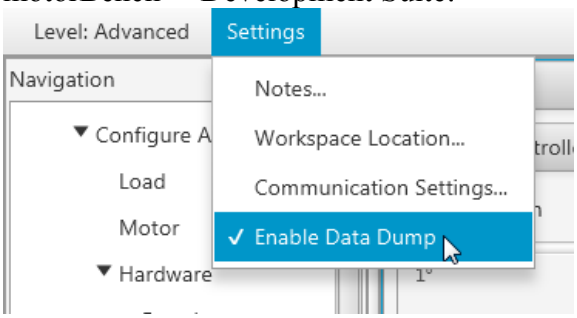
6 issues

Customer Support

If you need assistance, have questions or feedback related to motorBench™ Development Suite, please post them on the Microchip Development Tools forum, under "motorBench™ Development Suite" section.

Guidelines

- In many cases we need both the MPLAB X project settings, and data dump files to analyze information from motorBench™ Development Suite. Please make sure you enable the data dump every time you run motorBench™ Development Suite:



- The easiest way to providing the complete set of relevant debugging information is to zip up the entire MPLAB X project folder and attach this to your forum post. Please remove the "build" sub directory in the MPLAB X project folder in order to keep file sizes minimal.
- In the event that you run into an error message dialog, please copy and paste the entire text of the dialog box. This includes the complete stack trace information and is more useful than a screen snapshot.

In general, 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.