DSPMC Mach4
Software Integration

Ethernet Motion Controller
Data Acquisition System
PID Controller

Document Revision 1.1
(Updated May 5, 2016)

Phoenix, AZ USA
For more information please visit the product web page:
www.vitalsystem.com/dspmc

Table of Contents

License Agreement

Before using the DSPMC and accompanying software tools, please take a moment to go thru this License agreement. Any use of this hardware and software indicate your acceptance to this agreement.

It is the nature of all machine tools that they are dangerous devices. In order to be permitted to use DSPMC on any machine you must agree to the following license:

I agree that no-one other than the owner of this machine, will, under any circumstances be responsible, for the operation, safety, and use of this machine. I agree there is no situation under which I would consider Vital Systems, or any of its distributors to be responsible for any losses, damages, or other misfortunes suffered through the use of the DSPMC board and its software. I understand that the DSPMC board is very complex, and though the engineers make every effort to achieve a bug free environment, that I will hold no-one other than myself responsible for mistakes, errors, material loss, personal damages, secondary damages, faults or errors of any kind, caused by any circumstance, any bugs, or any undesired response by the board and its software while running my machine or device.

I fully accept all responsibility for the operation of this machine while under the control of DSPMC, and for its operation by others who may use the machine. It is my responsibility to warn any others who may operate any device under the control of DSPMC board of the limitations so imposed.

I fully accept the above statements, and I will comply at all times with standard operating procedures and safety requirements pertinent to my area or country, and will endeavor to ensure the safety of all operators, as well as anyone near or in the area of my machine.

WARNING: Machines in motion can be extremely dangerous! It is the responsibility of the user to design effective error handling and safety protection as part of the system. VITAL Systems shall not be liable or responsible for any incidental or consequential damages. By Using the DSPMCv2 motion controller, you agree to the license agreement.

Introduction

IMPORTANT

This document makes the assumption that the reader has thoroughly reviewed the DSPMC User Manual, has completed the proper hardware setup, and possesses basic knowledge and understanding of Mach4 CNC Software.

This document **DOES NOT** serve as a primer or tutorial for the use of Mach4. As such, readers without basic understanding of Mach4, and other software components not associated with Vital System Inc. are advised to consult the appropriate user manual or software vendor.

This document only covers integrating the DSPMC with Mach4.

Mach4 CNC Software is an off-the-shelf Milling and Lathe machine control software. The trial version of the software can be downloaded from the machsupport website.

The DSPMC board can be integrated with Mach4 to form a high performance machining center. The DSPMC Software Tools provide the necessary drivers and configuration files to interface with Mach4 software.

NOTE: *Several notes such as this can be found throughout this document which list important key points and comments worth noting.*

Mach4 Configuration

DSPMC Mach4 Plugin Setup

To setup the DSPMC plugin with Mach4, please follow the steps below.

1. Download the latest Mach4 DSPMC plugin here.

2. Open the zip file and the plugin files "***M4DSPMC.m4pw***" and "***M4DSPMC.sig***" can be found within.

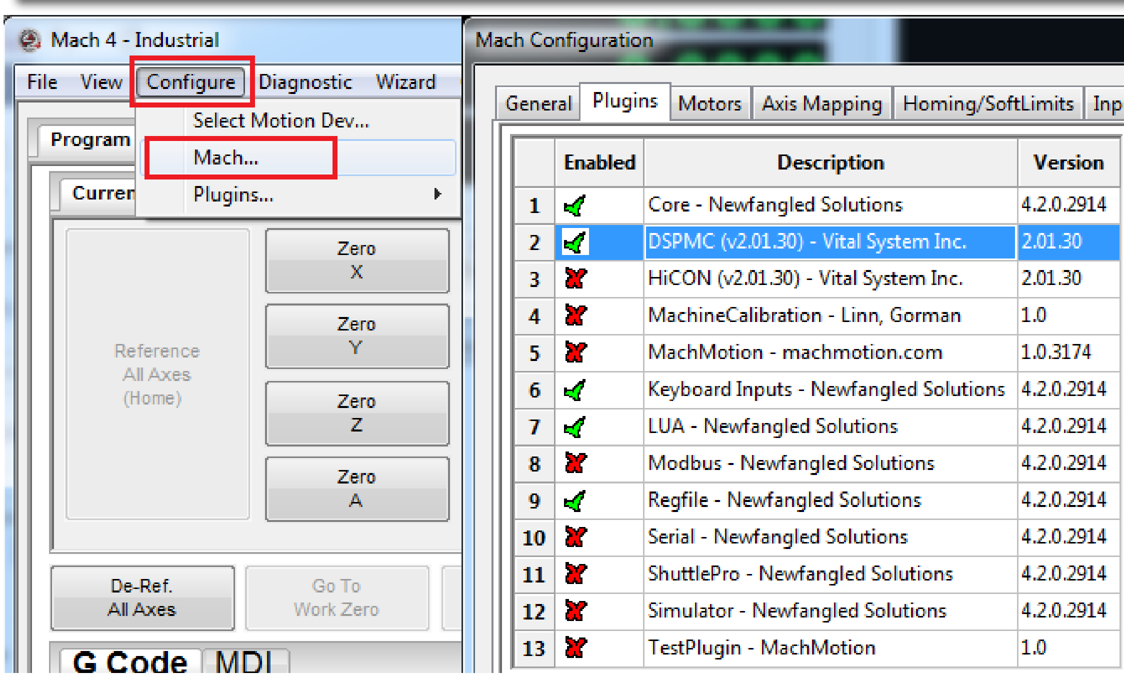
3. To use the DSPMC plugin for Mach4, extract or copy the "***M4DSPMC.m4pw***" and "***M4DSPMC.sig***" files to the ***Plugins*** folder in the Mach4 install directory (*usually C:/Mach4Hobby/Plugins*).

4. To launch Mach4 with DSPMC plugin, double-click on the "***Mach4GUI.exe***" software icon on the desktop (or in the Mach4 install folder).

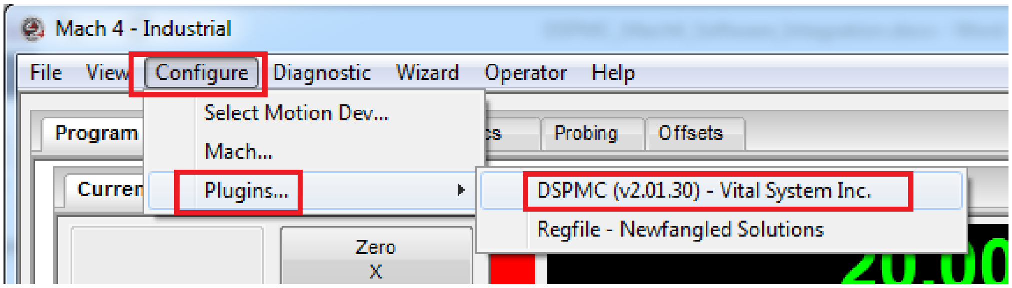
5. Enable the DSPMC plugin. The DSPMC plugin configuration window can be accessed by going to the Mach4 main window, then the "*Configure*" menu item (top of the main window), then "*Plugins…*" which will provide you with the following screen.

NOTE: Ensure Mach4 is disabled before accessing the *Configuration menus or they will be disabled.*

NOTE: *Enabling/Disabling plugins will only take effect after Mach4 is restarted. If enabling the DSPMC plugin for the first time, restart Mach4 now.*



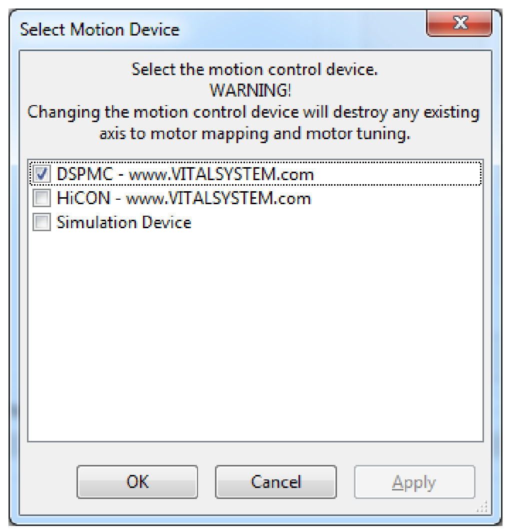
6. The "DSPMC Plugin Configuration" window can be accessed by going to the Mach4 main window, then the following menu sequence (from the top of the main window) "*Configure->Plugins->DSPMC*".



> **NOTE**: *Take note of this procedure to access the DSPMC Plugin Configuration as a lot of operational parameters are configured from the DSPMC plugin window.*

Starting Mach4 with DSPMC

If the steps in the plugin setup were followed correctly, you should be provided with the dialog box to select the motion device with the DSPMC as an option on Mach4 startup. Make sure this plugin is selected and click 'OK'. (*See image to the right*)

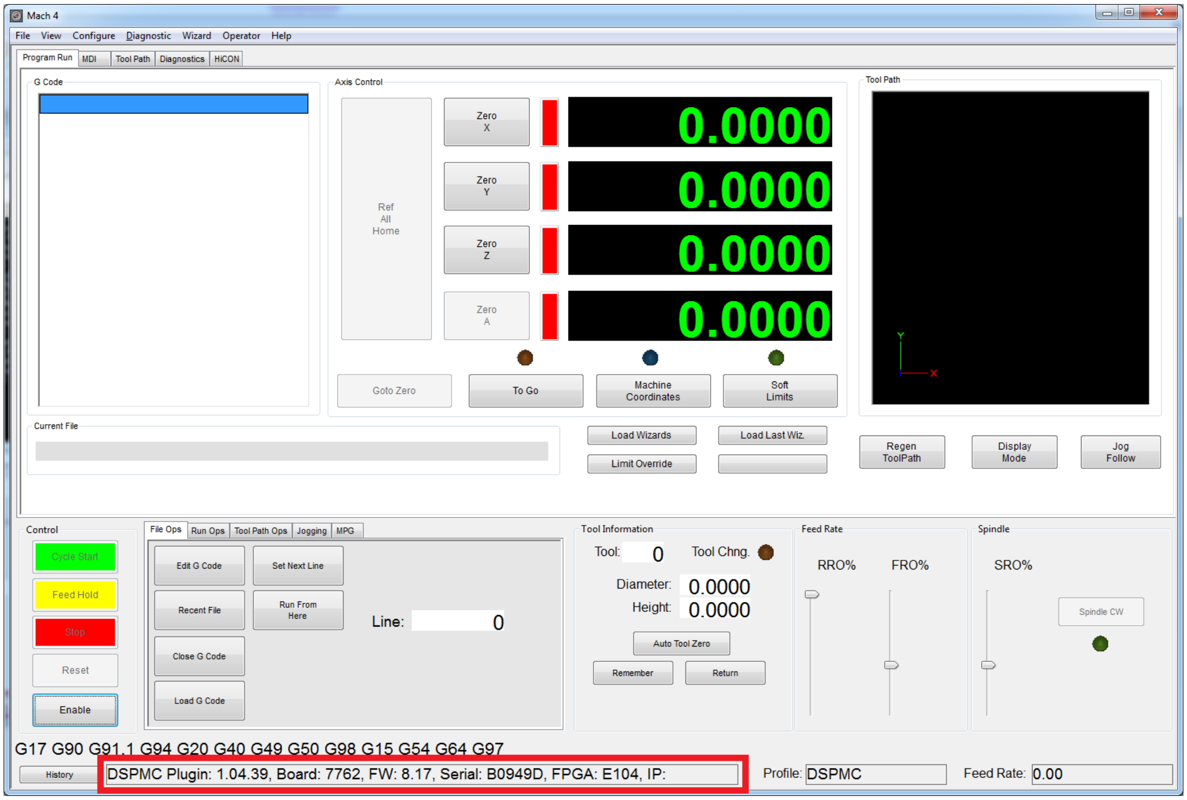


> **NOTE**: *Mach 4 must be restarted when the current motion device is changed or a new one is selected.*

The "Select Motion Device" window can also be accessed from the "Configure" menu item (top menu in the main window), then "Set Motion Device..."

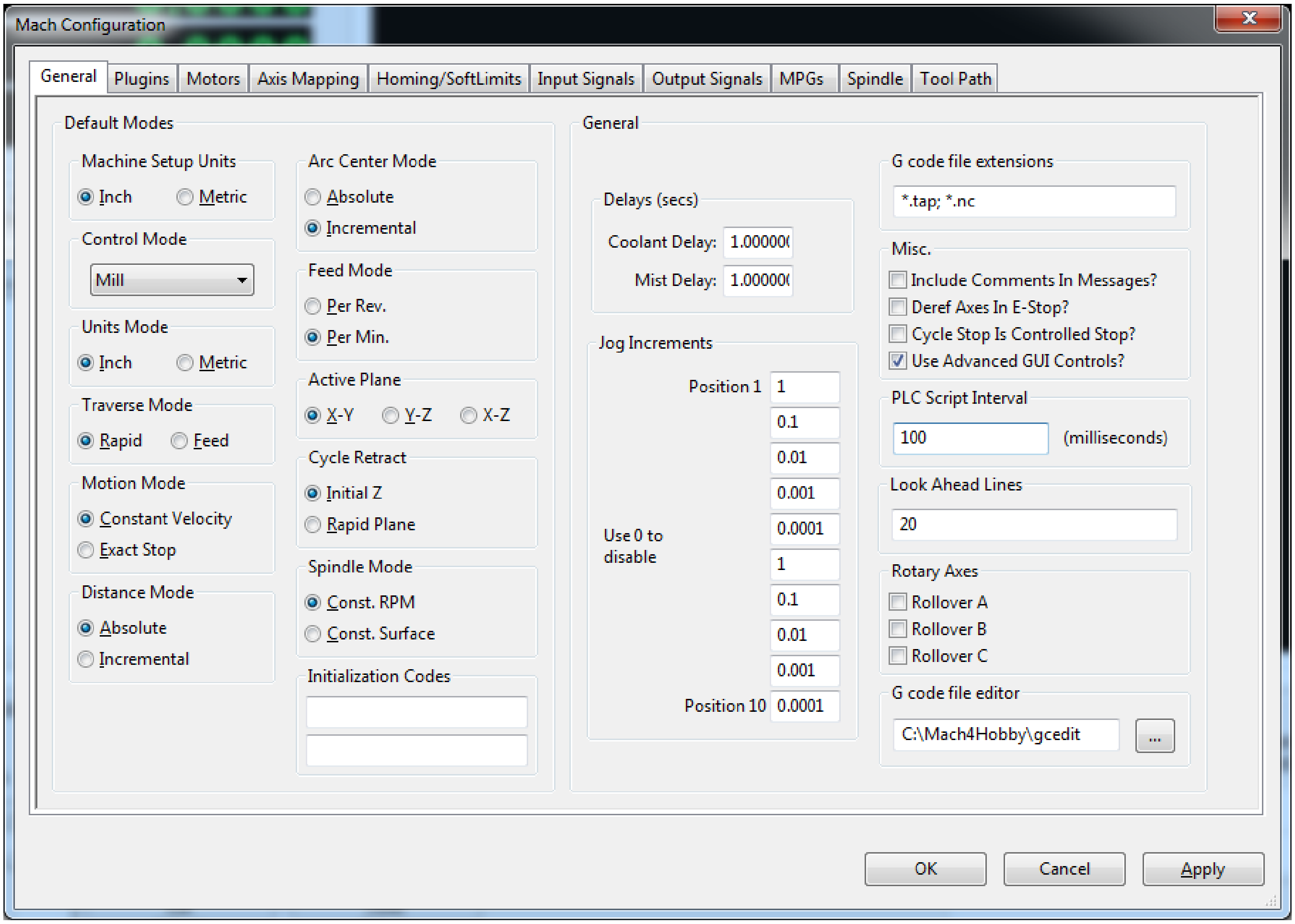
Make sure the DSPMC is powered up and properly connected to the Ethernet network. Mach4 will automatically search through all networks for any DSPMC and, if successful, will display a status message containing information for the currently connected DSPMC. (*See image below*).

> **NOTE**: *If you do not see the status message in the image below, then the plugin has not connected with any DSPMC on the network.*
>
> - *Refer to the "**DSPMC User Manual**" to resolve network connection issues or how to change your PC's network adapter IP address.*
> - *Refer to the **"VSI Device Manager User Manual"** to change the DSPMC IP address.*
> - *The DSPMC and PC IP addresses must be located on the same network.*


Mach 4
File View Configure Diagnostic Wizard Operator Help
Program Run MDI Tool Path Diagnostics HICON
G Code
Axis Control
Tool Path
Ref All Home
Zero X
Zero Y
Zero Z
Zero A
0.0000
0.0000
0.0000
0.0000
Goto Zero
To Go
Machine Coordinates
Soft Limits
Current File
Load Wizards
Load Last Wiz.
Limit Override
Regen ToolPath
Display Mode
Jog Follow
Control
Cycle Start
Feed Hold
Stop
Reset
Enable
File Ops Run Ops Tool Path Ops Jogging MPG
Edit G Code
Set Next Line
Recent File
Run From Here
Close G Code
Load G Code
Line: 0
Tool Information
Tool: 0
Tool Chng.
Diameter: 0.0000
Height: 0.0000
Auto Tool Zero
Remember
Return
Feed Rate
RRO%
FRO%
Spindle
SRO%
Spindle CW
G17 G90 G91.1 G94 G20 G40 G49 G50 G98 G15 G54 G64 G97
History
DSPMC Plugin: 1.04.39, Board: 7762, FW: 8.17, Serial: B0949D, FPGA: E104, IP:
Profile: DSPMC
Feed Rate: 0.00

Mach4 Configuration

Open the Mach Configuration window by going to the Mach4 main window, and on the main menu (top of the main window), click on *"Configure"*, and then *"Mach…"* from the drop-down menu. You should see the following window.

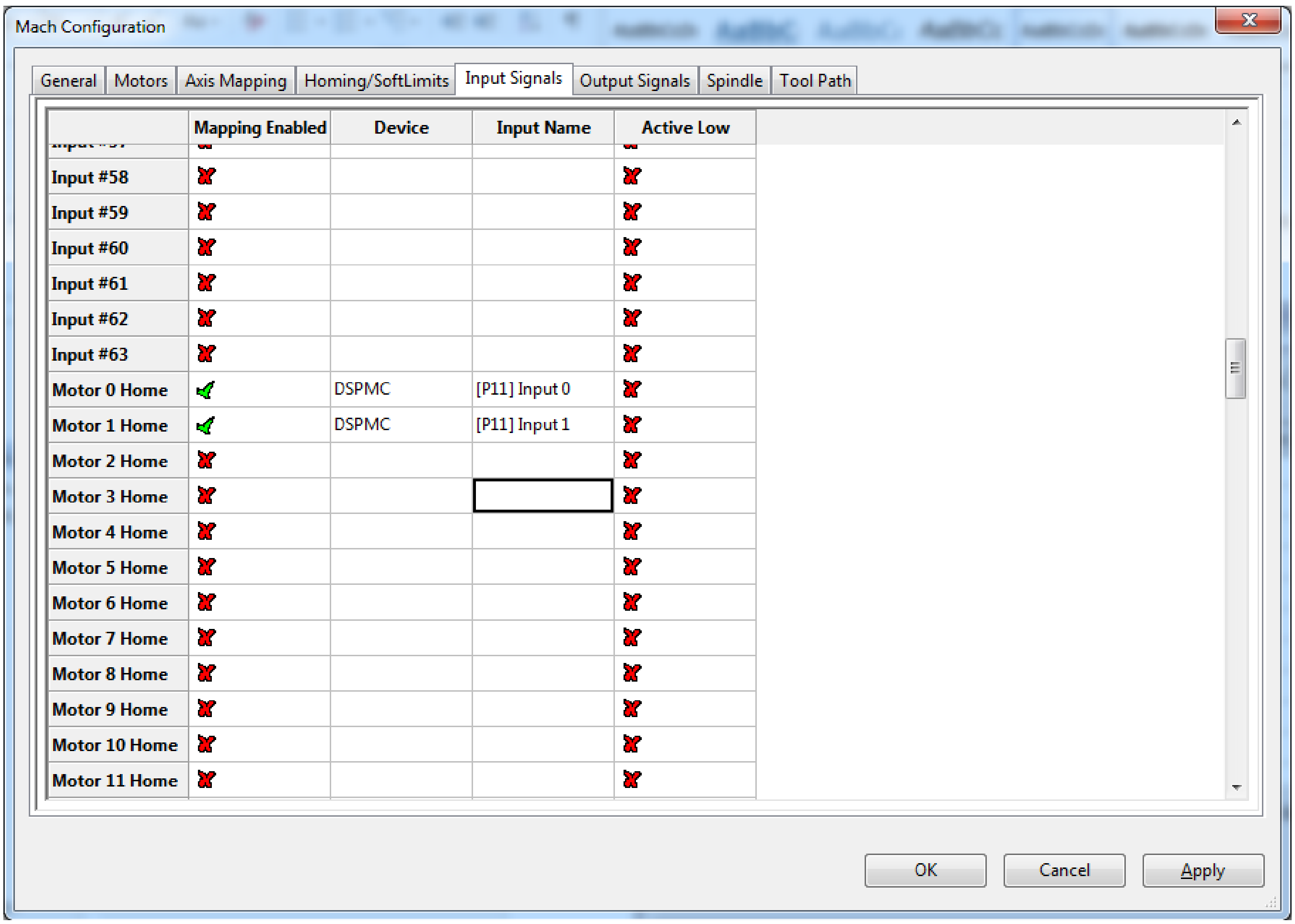


From here, click on the "Input Signals" tab, or the "Output Signals" tab to configure the Digital I/O Settings. It is imperative to setup the necessary limit and estop switches before attempting to perform motion.

Digital I/O

To map a Mach4 signal to a DSPMC Digital I/O, the selected device must be set to "***DSPMC***", and the Input/Output name set to one of the available Digital I/O from the dropdown menu. (*See image below*).

> **<u>NOTE</u>**: *It is highly recommended to first set up the ESTOP and limit switches before attempting jogging (or motion in general) on the machine.*



The following section provides a reference when mapping Mach4 Signals to the physical I/O on the DSPMC.

Mapping Mach4 Inputs to DSPMC Digital Input Pins

The following table shows the mapping from DSPMC Mach4 inputs to the actual digital input pin numbers available on the DSPMC board.

Mach4 Input	DSPMC J4 Input Pin	7535 Breakout Board Input	DSPMC BASIC Macro GetPin() Index
[P11] Input 0	**18**	0	11, 0
[P11] Input 1	**6**	1	11, 1
[P11] Input 2	**19**	2	11, 2
[P11] Input 3	**7**	3	11, 3
[P11] Input 4	**20**	4	11, 4
[P11] Input 5	**8**	5	11, 5
[P11] Input 6	**21**	6	11, 6
[P11] Input 7	**9**	7	11, 7
[P11] Input 8	**22**	8	11, 8
[P11] Input 9	**10**	9	11, 9
[P11] Input 10	**23**	10	11, 10
[P11] Input 11	**11**	11	11, 11
[P11] Input 12	**24**	12	11, 12
[P11] Input 13	**12**	13	11, 13
[P11] Input 14	**25**	14	11, 14
[P11] Input 15	**13**	15	11, 15

Mach4 Input	DSPMC J5 Input Pin	7535 Breakout Board Input	DSPMC BASIC Macro GetPin() Index
[P12] Input 0	**18**	0	12, 0
[P12] Input 1	**6**	1	12, 1
[P12] Input 2	**19**	2	12, 2
[P12] Input 3	**7**	3	12, 3
[P12] Input 4	**20**	4	12, 4
[P12] Input 5	**8**	5	12, 5
[P12] Input 6	**21**	6	12, 6
[P12] Input 7	**9**	7	12, 7
[P12] Input 8	**22**	8	12, 8
[P12] Input 9	**10**	9	12, 9
[P12] Input 10	**23**	10	12, 10
[P12] Input 11	**11**	11	12, 11
[P12] Input 12	**24**	12	12, 12
[P12] Input 13	**12**	13	12, 13
[P12] Input 14	**25**	14	12, 14
[P12] Input 15	**13**	15	12, 15

Mach4 Input	DSPMC J11 Input Pin	7535 Breakout Board Input	DSPMC BASIC Macro GetPin() Index
[P13] Input 0	**18**	0	13, 0
[P13] Input 1	**6**	1	13, 1
[P13] Input 2	**19**	2	13, 2
[P13] Input 3	**7**	3	13, 3
[P13] Input 4	**20**	4	13, 4
[P13] Input 5	**8**	5	13, 5
[P13] Input 6	**21**	6	13, 6
[P13] Input 7	**9**	7	13, 7
[P13] Input 8	**22**	8	13, 8
[P13] Input 9	**10**	9	13, 9
[P13] Input 10	**23**	10	13, 10
[P13] Input 11	**11**	11	13, 11
[P13] Input 12	**24**	12	13, 12
[P13] Input 13	**12**	13	13, 13
[P13] Input 14	**25**	14	13, 14
[P13] Input 15	**13**	15	13, 15

Mach4 Input	DSPMC J12 Input Pin	7535 Breakout Board Input	DSPMC BASIC Macro GetPin() Index
[P14] Input 0	**18**	0	14, 0
[P14] Input 1	**6**	1	14, 1
[P14] Input 2	**19**	2	14, 2
[P14] Input 3	**7**	3	14, 3
[P14] Input 4	**20**	4	14, 4
[P14] Input 5	**8**	5	14, 5
[P14]Input 6	**21**	6	14, 6
[P14] Input 7	**9**	7	14, 7
[P14] Input 8	**22**	8	14, 8
[P14] Input 9	**10**	9	14, 9
[P14] Input 10	**23**	10	14, 10
[P14] Input 11	**11**	11	14, 11
[P14] Input 12	**24**	12	14, 12
[P14] Input 13	**12**	13	14, 13
[P14] Input 14	**25**	14	14, 14
[P14] Input 15	**13**	15	14, 15

Mapping Mach4 Outputs to DSPMC Digital Output Pins

The following table shows the mapping from DSPMC Mach4 outputs to the actual digital output pin numbers available on the DSPMC board.

Mach4 Output	DSPMC J4 Output Pin	7535 Breakout Board Output	DSPMC BASIC Macro SetPin () Index
[P11] Output 0	**14**	0	11, 0
[P11] Output 1	**2**	1	11, 1
[P11] Output 2	**15**	2	11, 2
[P11] Output 3	**3**	3	11, 3
[P11] Output 4	**16**	4	11, 4
[P11] Output 5	**4**	5	11, 5
[P11] Output 6	**17**	6	11, 6
[P11] Output 7	**5**	7	11, 7

Mach4 Output	DSPMC J5 Output Pin	7535 Breakout Board Output	DSPMC BASIC Macro SetPin () Index
[P12] Output 0	**14**	0	12, 0
[P12] Output 1	**2**	1	12, 1
[P12] Output 2	**15**	2	12, 2
[P12] Output 3	**3**	3	12, 3
[P12] Output 4	**16**	4	12, 4
[P12] Output 5	**4**	5	12, 5
[P12] Output 6	**17**	6	12, 6
[P12] Output 7	**5**	7	12, 7

Mach4 Output	DSPMC J11 Output Pin	7535 Breakout Board Output	DSPMC BASIC Macro SetPin () Index
[P13] Output 0	**14**	0	13, 0
[P13] Output 1	**2**	1	13, 1
[P13] Output 2	**15**	2	13, 2
[P13] Output 3	**3**	3	13, 3
[P13] Output 4	**16**	4	13, 4
[P13] Output 5	**4**	5	13, 5
[P13] Output 6	**17**	6	13, 6
[P13] Output 7	**5**	7	13, 7

Mach4 Output	DSPMC J12 Output Pin	7535 Breakout Board Output	DSPMC BASIC Macro SetPin () Index
[P14] Output 0	**14**	0	14, 0
[P14] Output 1	**2**	1	14, 1
[P14] Output 2	**15**	2	14, 2
[P14] Output 3	**3**	3	14, 3
[P14] Output 4	**16**	4	14, 4
[P14] Output 5	**4**	5	14, 5
[P14] Output 6	**17**	6	14, 6
[P14] Output 7	**5**	7	14, 7

Motor Parameters

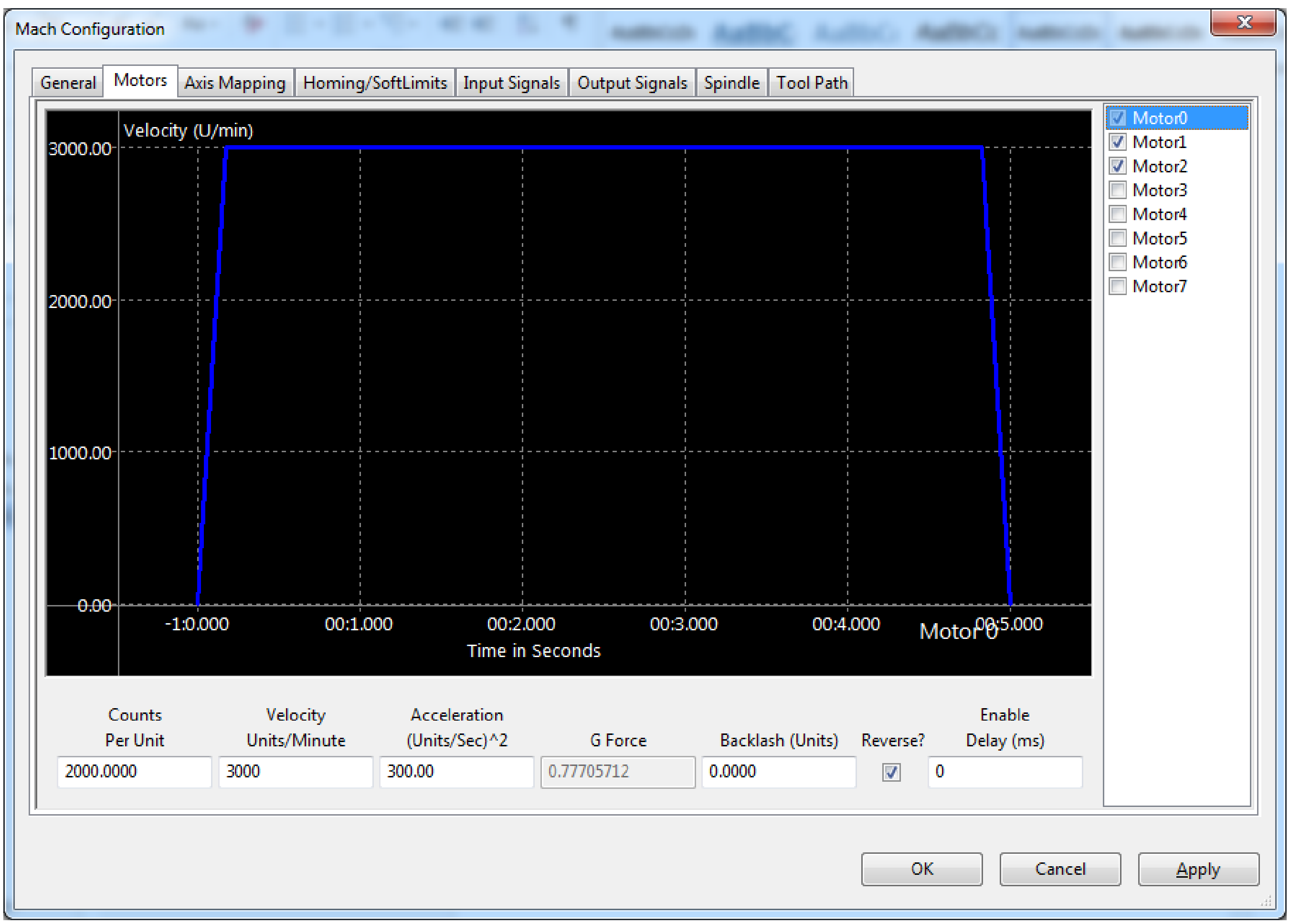
On the "Motors" tab, enable the motors that will be used by checking the checkbox of the corresponding motor to the upper-right of the window.

NOTE: *The Motor Backlash SHOULD NOT be configured from this window. As such, it should always be set to zero in this window. The motor backlash values are configured from the DSPMC Plugin Window.*

Because of the flexible nature of Mach4, the number of available motors is determined by the currently selected motion control device. For example, the DSPMC motion controller can control a maximum of 8 motors, while the HiCON motion controller can control 6 motors.

NOTE: *These motors correspond to the motor config in the DSPMC plugin config. As such, these parameters are utilized by the DSPMC plugin when generating motion.*

The DSPMC will only arm a motor if it is enabled from this window

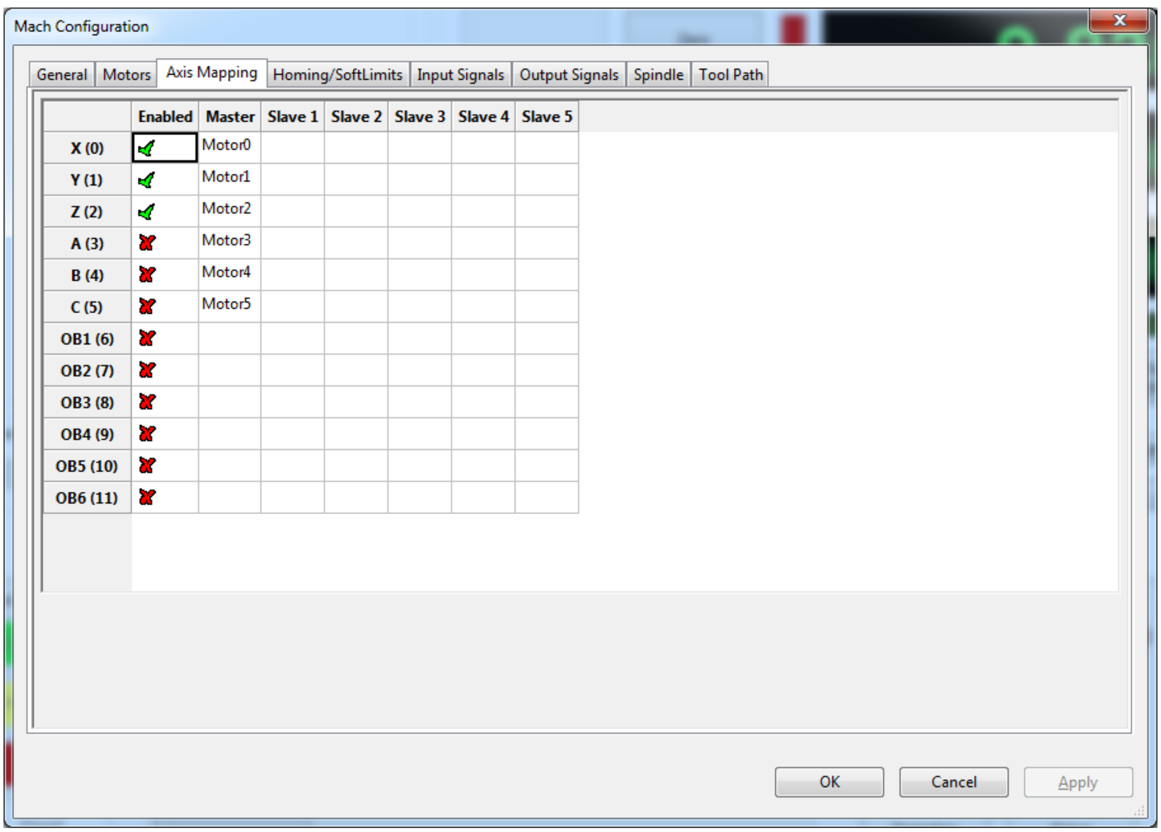


Mach4 Axis Mapping

Mach4 marks a clear distinction between the definition of an "Axis" and a "Motor". As such, it is improper terminology to interchange the two. An "Axis" represents the logical component of a motion vector, while a "Motor" represents the physical component.

> **NOTE**: *In Mach4, motion is commanded on an axis via on-screen/keyboard jogging, MPG jogging, or GCode commands through MDI or a file. An axis can have one or several motors under its control which will be moved according to the commanded motion on the axis.*

With this in mind, the available motors must be organized under the control of the Mach4 Axes. This can be done with the "Axis Mapping" tab in the Mach4 config.



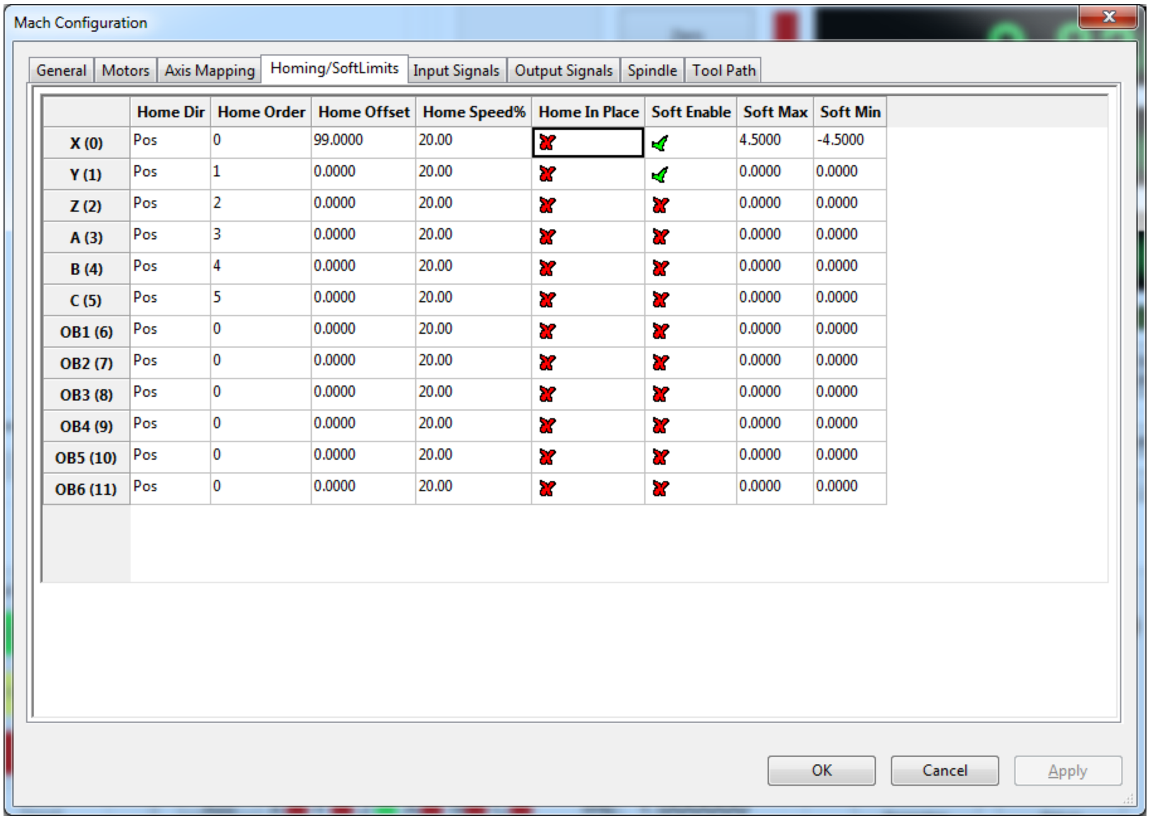
From this window, an axis can be quickly enabled/disabled, motors can be assigned as master or slave on any axis, and motors can be reassigned to operate on different axes without having to modify the motor configuration in the DSPMC plugin.

> **NOTE**: *An unmapped motor will still be armed by the DSPMC as long as it is enabled from the "Motors" tab in the Mach4 config. This special case is not typically used for most systems, but rather, it is for motors that will be controlled outside of Mach4 (e.g. through the DSPMC Macro feature).*

Axis Homing Setup and Soft Limits

Refer to the Homing Parameters in the System Tab of the DSPMC Plugin Config for additional settings for the homing procedure. Also, refer to the Mach4 manual for information regarding the fields in this window. General Mach4 knowledge applies to this setup.

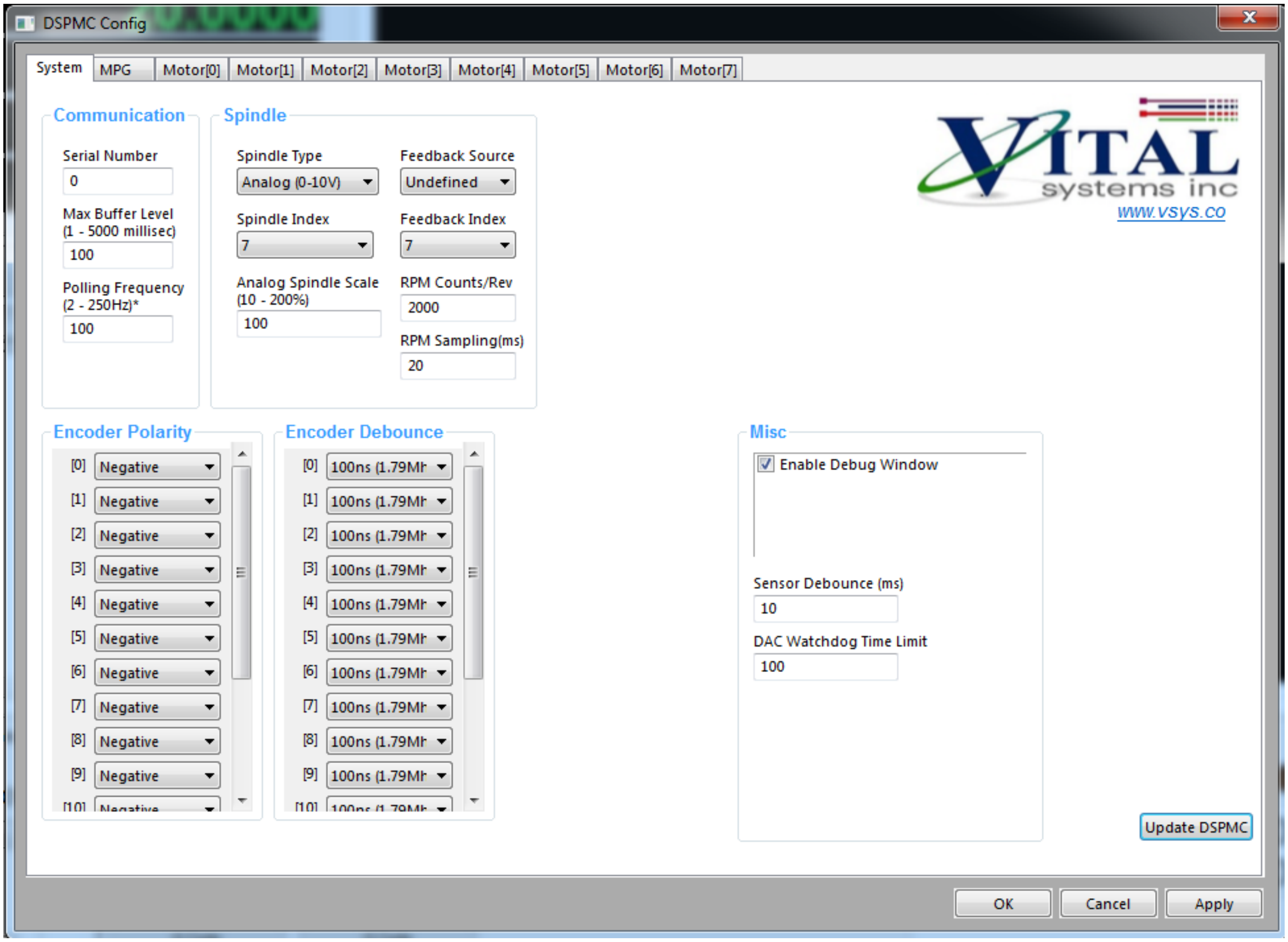
In the Mach4 Configuration Window, select the “Homing/SoftLimits” tab.



	Home Dir	Home Order	Home Offset	Home Speed%	Home In Place	Soft Enable	Soft Max	Soft Min
X (0)	Pos	0	99.0000	20.00	✗	✓	4.5000	-4.5000
Y (1)	Pos	1	0.0000	20.00	✗	✓	0.0000	0.0000
Z (2)	Pos	2	0.0000	20.00	✗	✗	0.0000	0.0000
A (3)	Pos	3	0.0000	20.00	✗	✗	0.0000	0.0000
B (4)	Pos	4	0.0000	20.00	✗	✗	0.0000	0.0000
C (5)	Pos	5	0.0000	20.00	✗	✗	0.0000	0.0000
OB1 (6)	Pos	0	0.0000	20.00	✗	✗	0.0000	0.0000
OB2 (7)	Pos	0	0.0000	20.00	✗	✗	0.0000	0.0000
OB3 (8)	Pos	0	0.0000	20.00	✗	✗	0.0000	0.0000
OB4 (9)	Pos	0	0.0000	20.00	✗	✗	0.0000	0.0000
OB5 (10)	Pos	0	0.0000	20.00	✗	✗	0.0000	0.0000
OB6 (11)	Pos	0	0.0000	20.00	✗	✗	0.0000	0.0000

NOTE: *The* DSPMC *Plugin assigns the “Home Offset” value as the current axis position after a successful Homing Sequence for the specified axis.*

DSPMC System Configuration



The "***System***" tab of the DSPMC Plugin Configuration contains general settings for the DSPMC Motion Controller. Clicking on the "***Update DSPMC***" button will transmit these settings to the DSPMC controller. Clicking "***OK***" or "***Apply***" will also transmit the data, and save this data in the selected Mach4 profile (e.g. Mach4mill, Mach4turn etc.) Clicking on "***Cancel***" will disregard all the changes made in this window.

In the following sub sections users can find detailed information about various configuration options that are provided under the system tab.

Communication

- ***Serial Number*** – This parameter allows Mach4 to selectively connect to a DSPMC (if multiple are present on the network) with the 6-character serial number written on the DSPMC. If set to zero, Mach4 will connect to the first DSPMC it finds. *A value of zero is recommended if there is only one DSPMC connected to the Mach4 PC.*

- ***Max Buffer Level –*** This parameter defines how much command position buffering will be done inside the DSPMC controller. The maximum buffering level is around 4096 motion vectors (4 seconds worth of motion). A motion vector is defined as a vector containing all axis positions (XYZABC). Motion vectors are generated in a millisecond resolution. These vectors are consumed by the DSPMC to produce motion at a frequency of 1 kHz (or every millisecond).

  Lower values make the motion more responsive to feedrate changes and feed holds, but may be more prone to jerking if the motion buffer were to prematurely become empty in the middle of motion. The ideal value is one where feedrate change responsiveness is adequate while preventing the motion buffer from going empty in the middle of motion.

  The valid rage for this parameter is 1 – 5000 milliseconds. *The recommended value is 100 – 500 at a polling frequency of 100Hz.*

- ***Polling Frequency –*** This parameter sets the update and data exchange frequency of the DSPMC plugin. A higher value will speed up the plugin processes and exchange data faster with the DSPMC, but it will also significantly increase network traffic and add more strain on the CPU.

  Valid values are 2 – 250Hz. Although Mach3 used a Frequency of 10Hz with every plugin to exchange data with devices, *the recommended value for Mach4 is 100Hz*.

Spindle Config

- ***Spindle Type*** and ***Spindle Index*** can be set to the following:

Undefined	Select this if a spindle is not used.
Analog (0-10V)	Most common for spindles driven by a VFD. Direction is determined by the Spindle Relay Outputs. *(Spindle Index should be set to 0)*
Analog (+/-10V)	Similar to the 0-10V analog signal, but the polarity of the voltage determines the direction.
GCode Axis	Sets a Mach-controlled motor to be used as a spindle. Typically used for spindles that are controlled by Step/Dir signals instead of Analog Voltage. *(Spindle Index determines which motor will be used)*

- ***Feedback Source and Index –*** These parameters define the feedback type for Spindle Speed/RPM calculation.

Undefined	Spindle RPM will be calculated based solely on the commanded RPM.
Encoder	Spindle RPM will be calculated from the selected Encoder Channel (*Determined by the Spindle Feedback Index)*. The encoder's differential A and B signals are used to calculate the RPM of the spindle. The Z (Index pulse) signal from the encoder is typically used to launch the Z-Axis at the right time in order to position tools correctly for cycles such as Threading and Rigid Tap. The RPM calculation is used to override/adjust the feedrate of the Z-Axis during these cycles.

- ***Analog Spindle Scale*** – Applies a percent scaling (10% - 200%) to the commanded Spindle Speed. This parameter is only used when an "Analog" spindle type is selected. *It is recommended to keep this value at 100% for an unmodified output ratio.*

- ***RPM Counts/Rev –*** This parameter defines the encoder resolution in terms of counts per revolution for Spindle Speed/RPM feedback. For quadrature encoders, the encoder resolution must be multiplied by 4.

- ***RPM Sampling (ms) –*** This parameter defines the timing window in milliseconds to sample the encoder counts for RPM calculation. For Threading and Rigid Tap, higher millisecond values cause the Z-Axis feedrate to become less responsive to changes in the spindle RPM, while a lower millisecond value will allow the Z-axis feedrate to react faster. *For low resolution encoders, this value should be high enough to accumulate enough pulses to calculate the Spindle RPM in a more consistent manner (approximately 100 – 200ms). Higher resolution encoders can use 10 – 50ms*. The valid range of this parameter spans 1 – 10000ms.

Misc Config

- ***Enable Debug Window –*** Selecting this option will have Mach4 open a debug window on the next startup. More "technical" debugging messages are displayed here and can be used for assistance in debugging problems in Mach4.

- ***Sensor Debounce*** – This parameter controls the debounce value (in milliseconds) for the digital inputs on the DSPMC. This value is useful if the sensor that is connected to a digital input on the DSPMC is experiencing inconsistent/premature ON and OFF states due to the presence of electrical noise. Higher values will filter out more noise and make sensor readings more consistent, but will increase the response delay for reading the input's state. This value cannot exceed 250ms. *The recommended value is typically 2- 5ms, but for systems experiencing a high level of electrical noise, a value of 5 – 10ms is more appropriate.*

- ***DAC Watchdog Time Limit*** – This setting only applies to motors controlled by a DAC Output (+/-10V). When a "runaway motor" condition is detected, this condition sets how much time (in milliseconds) to wait before performing an emergency stop in order to prevent damage to the machine. A runaway axis condition is one where the command position and actual position are not moving in the same direction. This could be a result of a reversed encoder polarity setting or reversed wiring on the encoders. *Setting this value to zero, disables the watchdog*.

Hardware Encoder Polarity

The ***Hardware Encoder Polarity*** field is used to reverse the direction of the encoder feedback signal. If the Differential A and B encoder signals are connected in reverse such that it does not match the motor control direction, the system will not be able to arm. To fix this issue, the hardware A and B signals can be reversed using this parameter.

> **NOTE**: *The Index pulse signal polarity is not affected by this setting. This encoder polarity setting only swaps the A and B signals to change the counter direction.*

Encoder Debounce

The ***Encoder debounce*** field is used to filter improper readings (due to electrical noise on a high-frequency signal) from the hardware encoder signals. *Lower debounce values filter less noise, while higher values reduce the maximum frequency of the encoder signals.*

> **NOTE**: While a recommended setting of *"100ns"* is normally sufficient, higher values may also be used if the electrical noise persists.

Update DSPMC (System Tab)

This button downloads all system configuration parameters to the DSPMC.

> **NOTE**: *System Config changes are ONLY applied after clicking this button or the "OK" button at the bottom of the config screen.*

MPG Setup

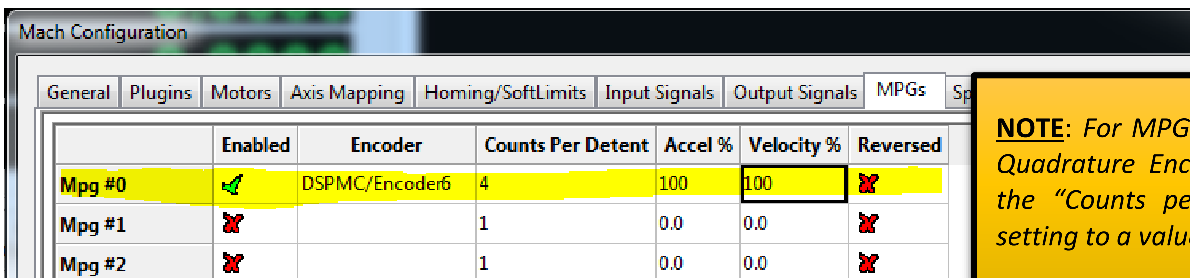
This section describes the settings for Mach4 and the DSPMC plugin if the encoder channels and digital I/O available on the DSPMC motion controller are to be used for MPG operations.

> **NOTE**: *For USB based MPGs (e.g. Shuttle Pro), this section can be disregarded, but make sure the "Enable VSI MPG I/O" setting is unchecked.*

> **NOTE**: MPG jogging is enabled anytime the Keyboard/On-screen jogging is allowed (e.g. No GCode File running, and no active motion sequences such as MDI or homing are being performed).
>
> It is always recommended to first configure the MPG ESTOP button (if available) before attempting to perform axis jogging with an MPG Pendant.

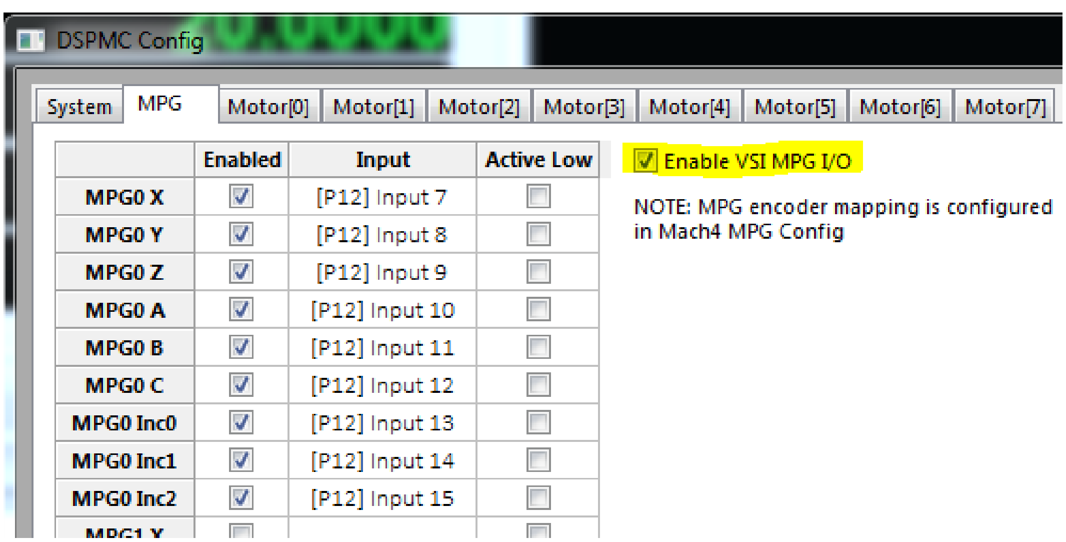
MPG Encoder Selection

To use an MPG Handwheel in Mach4, a DSPMC Encoder must be mapped to one of the Mach4 MPGs as shown in the Mach4 configuration below.



> **NOTE**: *For MPGs that use Quadrature Encoders, set the "Counts per Detent" setting to a value of 4.*

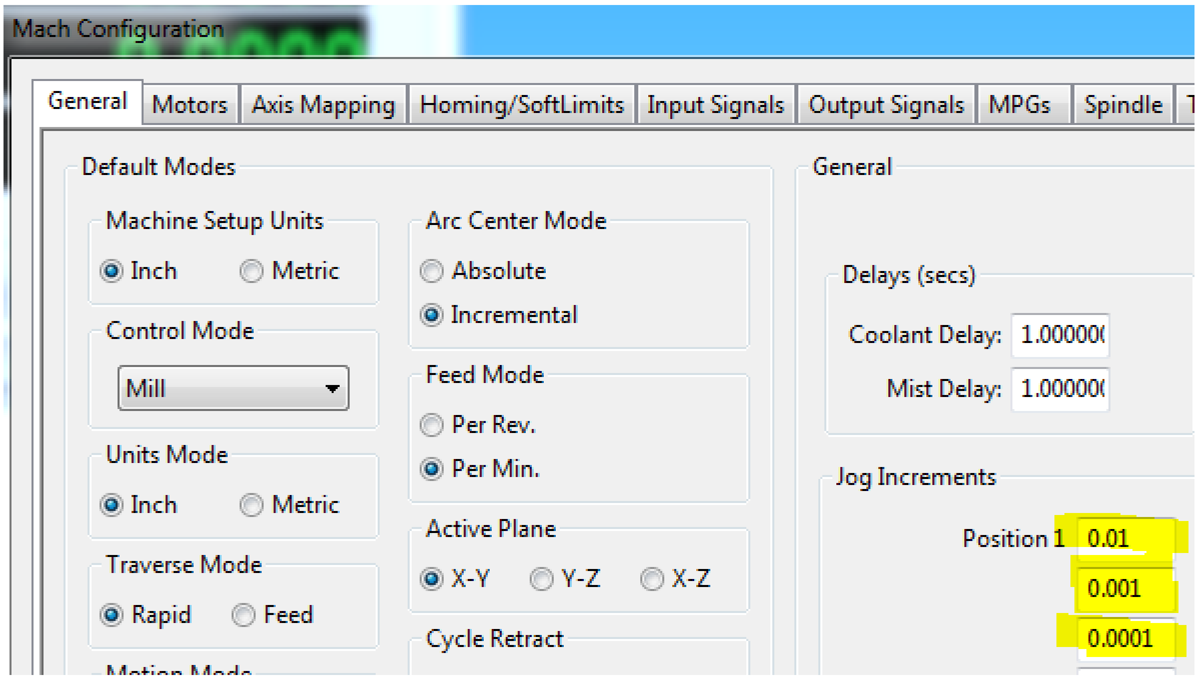
The MPG tab in the plugin config displays the available MPG actions and allows mapping digital inputs to them.



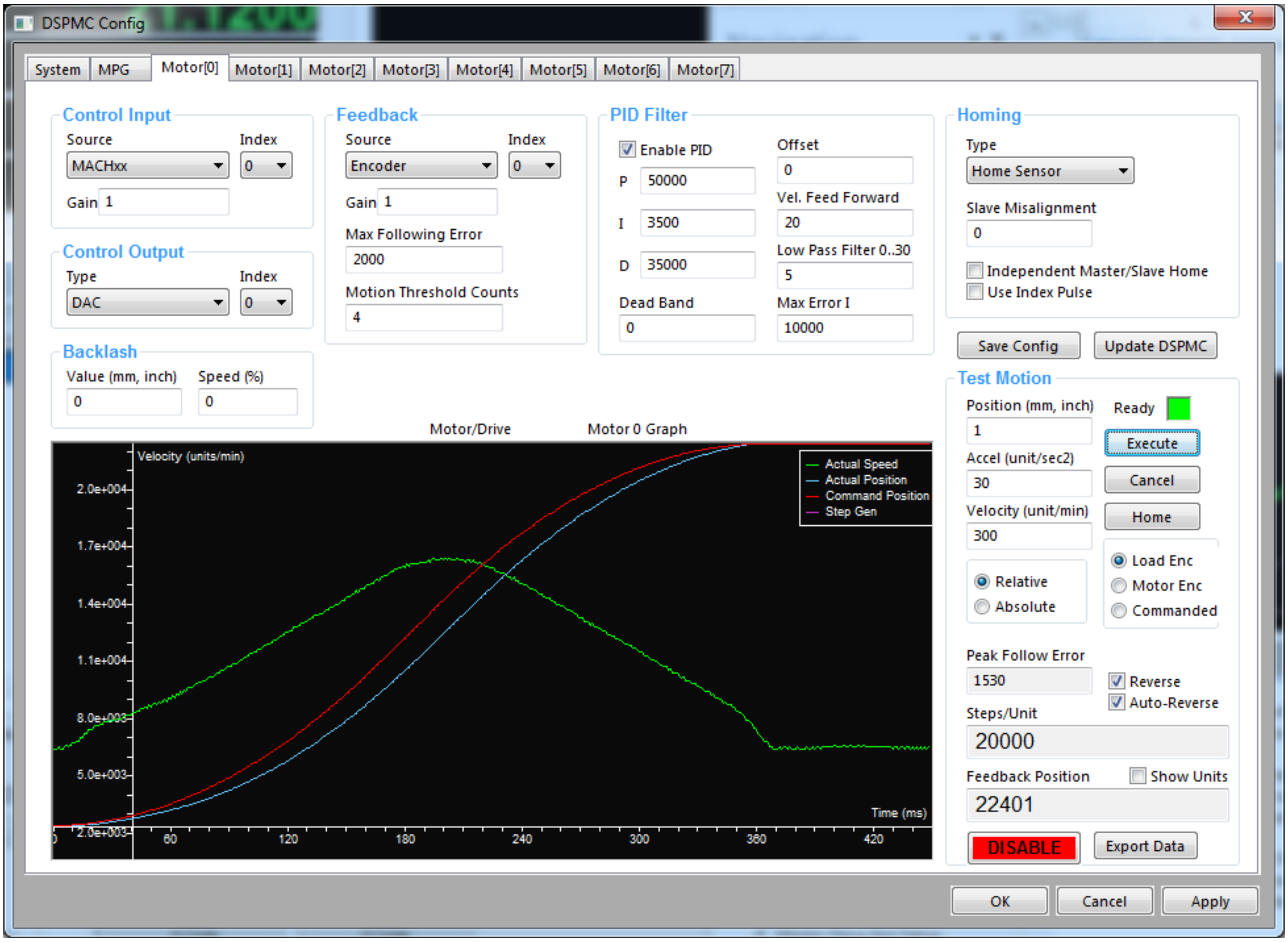
The DSPMC plugin MPG I/O behavior (axis and increment selector buttons) can be enabled/disabled via the "Enable VSI MPG I/O" checkbox.

MPG Jog Increments

The jog increment values for the increment selector switch are taken from the first 3 values of the jog increments in the Mach4 config:



DSPMC Motor Configuration



The Motor tabs provide configuration settings that are directly related to each motor. These tabs also provide motion testing features.

There are two sets of parameters, PID parameters and the Controls parameters. The Control Parameters are used to denote which command output and feedback channels are used to control the motor. The PID Parameters fine tune the output signal and are only used when a DAC Output (+/-10V motor control signal) is selected.

Motion may also be tested from this tab. The Test Motion component is the recommended starting point when attempting motion as it provides better diagnostics and a more controlled environment. It is preferred to start testing motion on this window before performing motion functions in Mach4 such as jogging, MDI, or GCode Files.

Control Input Parameters

- ***Control Input Source*** – This field defines the input type for a particular motor. *This should be set to **MACHxx** in all cases unless advised otherwise*.

> **NOTE**: *Motors can be enabled/disabled from the motor settings in the Mach4 config.*

- ***Control Input Index –*** Defines the index of the input source.

> **NOTE**: *In Mach4 and in almost all cases, it is recommended that this value **ALWAYS** be set to the motor number (e.g. Motor[0] to 0, Motor[1] to 1, Motor[2] to 2, etc.).*
>
> *Motors can be set as slaves through the "Axis Mapping" tab in the Mach4 configuration.*

- ***Control Input Gain –*** The commanded position for the motor is multiplied by this number in order to scale the outputted motion up or down. *It is normally recommended to leave this value at 1*.

Control Output Parameters

- ***Control Output Type –*** This field determines the output signal type for the selected motor. The possible values are:
  - ***DAC:*** Use one of the analog output channels as the PID control output. This setting is used for Servo Drives/Amplifiers that takes +/-10V as the command signal.

    > **NOTE**: *Some servo motors are operated by drives that can utilize either +/-10V or Step-Direction. In some cases, it may be more advisable to use "Stepper" instead as it does not require tuning the PID Parameters in the plugin which makes the setup easier.*

  - ***Stepper:*** Use one of the dedicated Step and Direction channels (0 – 7) as the output for position control (2 MHz max frequency). *This is the recommended setting*.
  - ***EtherCAT:*** Use a slaved EtherCAT Drive that supports the CIA402 Drive Profile specification. The output index determines which slave will be used in the EtherCAT chain.

    > **NOTE**: *This setting can only be used with the EtherCAT Adapter Board (pn7729) from Vital System Inc.*

  - ***Undefined:*** This setting causes the motor to be disabled to ignore the control output index.

- ***Control Output Index*** - The output channel number where the commanded motion will be issued.

Feedback Parameters

- ***Feedback Source –*** This determines the source of the feedback position for the selected motor. The possible values are:
  - ***Undefined:*** No feedback source is selected. The outputted step pulses are used instead as the feedback source.
  - ***A2D:*** Use one of the analog inputs as the PID feedback. This allows PID to be used for temperature and process control, in addition to motion control applications.
  - ***Encoder:*** Use one of the Differential Quadrature hardware encoder channels (0 – 7) as the PID feedback (4MHz max frequency).

> **NOTE**: *While some motors close the loop with the drives, routing the output encoder signals from the drive to the DSPMC provides the following notable benefits:*
> - *The physical machine position will be displayed on Mach. Mach will display any changes to the motor position even when moved manually.*
> - *Post-processed motion correction is improved.*
> - *Prevents the need to reference the machine multiple times (as the home position is preserved should the axis go out of band)*

- ***Feedback Index –*** Selects the index of the feedback source. *Not used when Feedback Source is set to Undefined.*

- ***Feedback Gain*** – This value is used to apply a gain to scale up/down the feedback reading. This is primarily used for encoders with resolutions that do not match the Step-Dir resolution.

  For example, if a command of 2000 counts caused the encoder to read back only 1000 counts, a feedback gain of 2 would be required.

- ***Max Follow Error*** – Maximum deviation allowed between command and actual position. If the controller detects a difference between the commanded and actual position that exceeds this limit, an emergency stop is triggered which will require the controller to be manually re-enabled.

> **NOTE:** *A value of 0 disables the following error check which may cause the motor to move at max uncontrolled speed (in a run-away situation),* ***which can be extremely dangerous****. As such, it is recommended to never use a zero value in this field.*

The required value for this field depends on multiple factors such as the maximum motor speed, drive tuning, mechanical characteristics of the axis etc. In general, this value should be set high enough to not accidentally trigger an ESTOP while the motor is moving at faster speeds (while still not out of band), but low enough to detect the error in a timely manner.

PID Filter

The values in this section define the co-efficient of PID filters for the selected motor. The PID filter runs at 5 KHz for each motor.

Setting the PID parameters requires a trial-and-error process of test motions with the objective of getting the commanded position as close as possible to the actual position.

To achieve the ideal tuning, do the test motions over very small distances (e.g. 3-5mm) and at the maximum velocity and acceleration that the motor will allow. This way, the PID values will be configured to accurately position the motor using the minimum amount of time.

> **NOTE:** A coarse PID tuning can be performed with the motors disengaged from the machine. After the coarse PID tuning appears satisfactory, the motors can be re-engaged to the machine and fine-tuned for the increased load.

Below is a description of the PID Parameters.

- ***P*** – Proportional Gain. Higher values increase the outputted Volts for the DAC output.
- ***I*** – Integral Gain. Higher values cause an increased response to closing the difference between the command and actual position.
- ***D*** – Derivative Gain. Higher values balance out oscillations during motor movement.
- ***Max Error I*** – Maximum Integral Error for the integral gain. This value must be greater than zero if a non-zero value was used for the "***I***" term.
- ***Dead band*** – a range around the commanded position where the PID is not active (when armed). For example, if the current command position is 1000 counts, with a Dead band of 10, the PID will be inactive between 990 and 1010 counts.
- ***Offset*** - Sets a constant bias to the PID output. This is useful to cancel any offset output voltage on the analog output channels.
- ***Velocity Feed Forward*** – This value bypasses the PID filter and directly applies an increase to the Analog Voltage output.
- ***Low Pass Filter*** – This field is used to smooth the analog output so the motors run smooth and with less noise. Because the PID loop runs at 5 KHz, ***a value of 5 is recommended*** as it will create a nice linearly increasing DAC output at 1 KHz. With a value of 10, the effective PID speed will become 500 Hz.

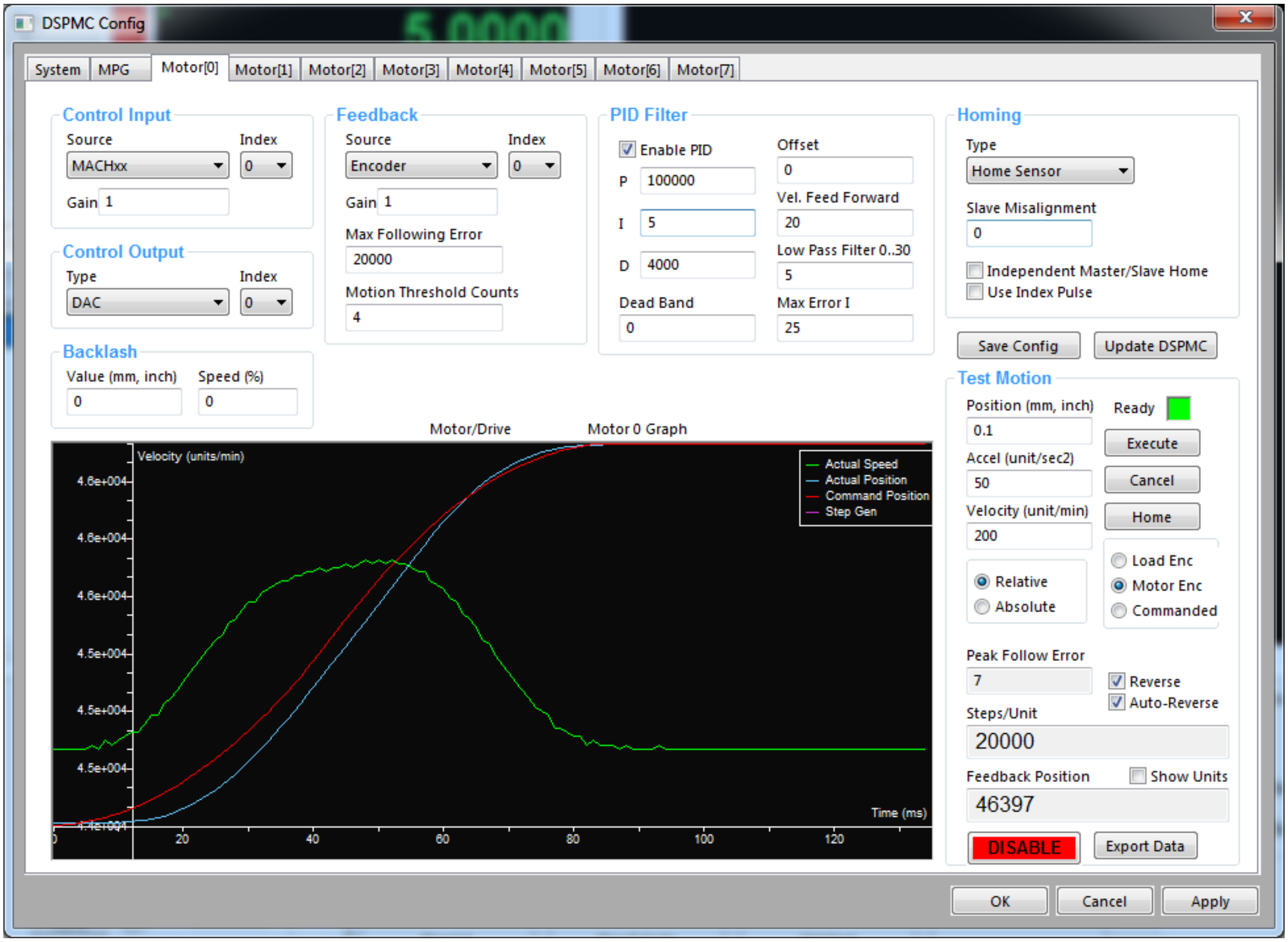
More information on other PID terms and general discussion on PID control is available at:
http://en.wikipedia.org/wiki/PID_control.

PID Tuning Steps

1. Set ***"P"*** to 100, "***Max Follow Error***" at 5000, and "***Low Pass Filter***" to 5. Set all other parameters to zero.
2. Do test motion commands with very high acceleration and velocity settings, but over a small position such as 0.1 inch or 3-5mm. *(Refer to the [Test Motion]() Section).*
3. If no motion is observed, start increasing "***P***" until you motion can be observed physically and on the motion graph. Check the Response graph for details on the motion.
4. If the actual position moves in the opposite direction of commanded, reverse the "***Encoder Polarity***" in the system tab.
5. Once a satisfactory motion graph is achieved (i.e. actual position follows commanded), the motion can be tuned to a finer degree:
   - Start gradually increasing the "***I***" value (start at 5) to close the gap faster between the commanded and actual position.
   - Set "***Max Error I***" to 100 and gradually increase to match with the "***I***" value.
   - Set "***Velocity Feed Forward***" as necessary, or if required (a value of 20 seems to work well).
   - If oscillations (jittering moves) are observed with the motion, increase the "***D***" term to balance out the oscillations (start at a tenth of the "***P***" Parameter and gradually increase as necessary).

   > **NOTE**: *When "**I**" term is greater than zero, make sure the "**Max Error I**" value is also non-zero, otherwise uncontrolled oscillations will be incurred.*

6. Ideally, the actual position (blue line) should be as close to commanded position (red line) as possible. Do the test motions for short and longer distance and repeat as necessary while tweaking the PID parameter values. Below is an image of an ideal PID tuning graph.



(Ideal PID Tuning Graph)

Homing Parameters

> **NOTE:** *Most homing parameters are configured in the Mach4 Homing Config. While the homing settings are configured for each axis in Mach4, each motor inherits the homing settings from the axis that it is mapped to. An axis can only be set as "referenced" only if all motors assigned to said axis are referenced.*

- ***Homing Type*** – Defines the homing sequence for each axis.

  The ***Home Sensor*** method uses the following sequence:

  - The motor is commanded to move in the configured homing direction (from Mach4) until the motor's home sensor (mapped from the Mach4 inputs tab) is activated.
  - It then moves in reverse at 20% of homing speed until the home sensor is deactivated.
  - If the motor is **NOT** set to ***Use Index Pulse,*** the home position is defined at this exact point and the home offset is assigned as the current position of the axis.
  - If the axis is set to ***Use Index Pulse***, the axis continues to move and only stops when the Encoder Index Pulse is triggered. The home position is defined at this exact point and the home offset is assigned as the current position of the axis.

> **NOTE:** *Limit sensors can also be used as home sensors by mapping the Mach4 limit and home signals to the same DSPMC Digital Input. The limit detection is only disabled during homing, and if the limit and home signals are mapped to the same digital input.*

- For ***Index Pulse Only***, the axis moves in the configured direction to locate the Encoder Index Pulse to home the motor. As soon as the index pulse is detected, the home position is defined at this exact point and the home offset is assigned as the current position of the axis.

- ***Slave Misalignment*** – This value sets the amount of additional motion a slave motor will perform after a Master/Slave Homing Sequence. This comes in handy in certain applications such as squaring a gantry after homing.

> **NOTE:** *This value is only used if the selected motor is mapped as a slave in the Mach4 Axis Mapping.*

- ***Independent Master/Slave Homing*** – This parameter enables a slave motor to home independently of the master.

> **NOTE:** *This option only applies if the selected motor is mapped as a master in the Mach4 Axis Mapping. This option also requires that a slave motor have its own home sensor mapped in the Mach4 Input Signals and wired to one of the DSPMC digital inputs.*

Test Motion

The Test Motion component can be utilized to accurately gauge the performance of the configured settings of the motors.

> **NOTE**: *It is required to set the motor parameters and the axis mapping within the Mach4 config (as well as the fields for the DSPMC Motor Configuration, before moving on with this section as some motion parameters are taken from the Mach4 config.*

> **NOTE**: *It is advised to first test the motion performance from here, in order to test the motion configuration for the motion controller, before attempting any jogging/motion from Mach4 itself. This method is also useful for testing the deviation between actual and command position when using encoder feedback for closed loop operation.*

The Ready LED shows if the DSPMC is armed and ready to accept motion commands. A motor can only execute one motion profile at a time *(ACCEL -> VELOCITY -> DECCEL -> STOP*), however, the other motors can still be commanded with test motion. A motor will only accept new motion commands if the current motion profile is cancelled, or once the current motor has stopped moving (or is still).

Once the test motion is complete, the result of how closely the motor had followed the commanded motion profile may be observed on the PID/Motor Response graph. The position, max velocity, and acceleration may be tweaked in order to verify the motor's performance at various commanded speeds and accelerations. If the "Auto Reverse" option is enabled, the axis will reverse the direction automatically at the start of the next Execute command and thus prevent the axis moving in only one direction during testing.

> **NOTE**: *The **velocity** and **acceleration** values for the Test Motion will override the velocity and acceleration values in the Mach4 Motor Settings. As such, the test motion can also be used as a quick method to gauge the performance limits of the motors.*

- ***Enable/Disable Button*** – Arm or Disarm the system.

  > **NOTE**: The system must be armed before performing motion.

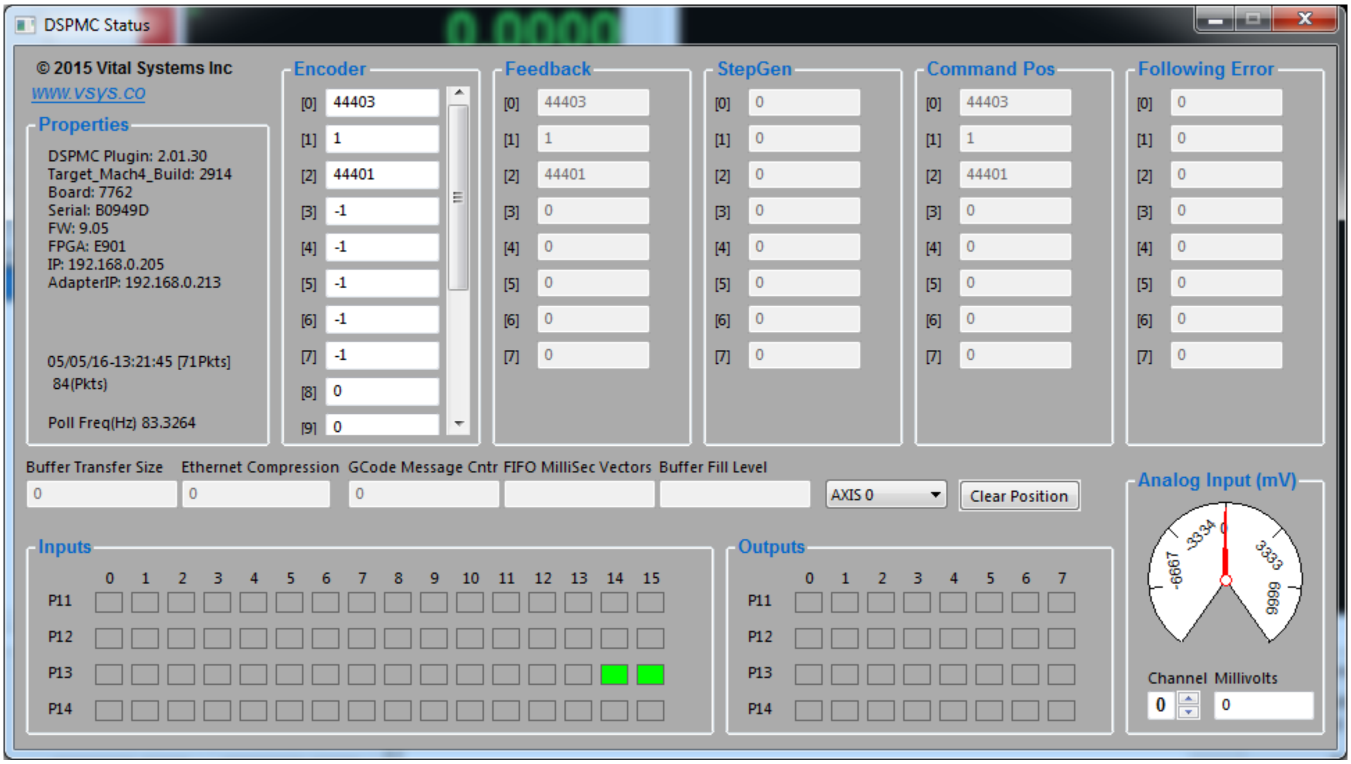
- ***Position*** – Final/Target position or displacement in terms of Position Units, (e.g. 1.5, 10.093, mm or inches etc.)
- ***Acceleration*** – Acceleration value in terms of units per second squared, (e.g. inches/second^2, mm/sec^2 etc.)
- ***Velocity*** – Motion velocity value in terms of Units per minute, e.g. inches/minute, mm/minute etc.
- ***Relative/Absolute*** – These check boxes indicate whether the value in the ***Position*** field is either the distance to travel (relative) or the final position (absolute).
- ***Execute Button*** – Issues the Execute-Motion command to the DSPMC. The "*Cancel*" button may be clicked to cancel the motion execution anytime during the machine operation. Make sure any changes to the

motor configuration are downloaded to the DSPMC by clicking "Update DSPMC" before clicking on Execute.

- ***Ready LED*** – shows if the current motion command is completed and DSPMC is ready for new motion commands. A new motion command can be launched with the "*Execute*" button when the Ready LED is Green. If the LED goes to Red after click on Execute, but no visible motion is observed, the velocity or acceleration may be too low.
- ***Home Button*** – Executes the Homing sequence based on Mach4 Homing Config, and Homing settings for the selected motor.
- ***Reverse –*** Checking this option will negate the value indicated in the ***Position*** field and thus reverse the direction of the motion.
- ***Auto Reverse –*** Enabling this setting automatically toggles the ***Reverse*** option between consecutive motion commands which makes it easier to test several motion commands while not only moging in one direction.
- ***Axis Position Display (DRO) –*** Shows the feedback position of the motor based on certain settings:
  - **Commanded position –** Displays the internal value on the motion controller for the commanded position of the selected motor.
  - **Load Encoder –** Displays the motor position derived from backlash and encoder feedback position.
  - **Motor Encoder –** Displays the motor position from encoder feedback. This value represents the motor's physical position on the machine.
  - **Show units –** Motor position will be displayed in units (mm, inches etc.). If not selected, the position will be displayed in raw counts.

DSPMC Status Window

To access the DSPMC Status Window, go to the Mach4 main window, and on the main menu (top of the main window), click on *"Diagnostic"*, and then *"DSPMC..."* from the drop-down menu. You should see the following window.



This window contains information about the current state of the DSPMC such as:

- ***Properties*** – Displays the device information of the DSPMC and activated features.
- ***Encoder*** – Displays the current encoder position. These fields are helpful for indicating if the encoders are properly connected.

> **NOTE**: *The fields can be clicked to clear the corresponding encoder position (must be disarmed)*.

- ***Feedback*** – Displays the current axis feedback. The value of this field depends on the currently selected feedback for the given motor/axis (e.g. Encoder counts in closed loop, or outputted step pulses in open loop).
- ***StepGen*** – Displays the counter for outputted step pulses. When using encoder feedback, this value should ideally be equal to the "***Encoder***" counts, although factory policy dictates a maximum difference of 4 counts between the Stepper and Encoder counts is allowed.

- ***Command Pos*** – Displays the command position. This value is either the generated position from Mach, or the generated position from internal motion commands such as those generated from the Macro Feature.
- ***Following Error*** – Displays the current following error. This value is the difference between the "***Command Pos"*** and the "***Feedback***". If this value increases beyond the configured "***Max Follow Error***", then the DSPMC immediately triggers an Estop which disarms the whole system.
- ***Analog Inputs*** – Displays the voltage reading (in millivolts) on an analog input.

The DSPMC Status window also displays the current Digital I/O states. It is recommended to consult this display for several reasons, some of which include the following:

- When verifying if sensors are properly wired and functioning (sensor state toggles as intended, and on the correct digital input).
- When diagnosing dysfunctional Mach4 signal states (e.g. ESTOP signal always active, limit switches are always active, etc.)
- To determine if a digital input/output is turning ON/OFF correctly.

NOTE: *Digital Output states can be manually toggled by clicking on the displayed LED.*

Additional Setup

Master-Slave Axis Setup

Motors can be set as masters or slaves in the Mach4 Axis Mapping

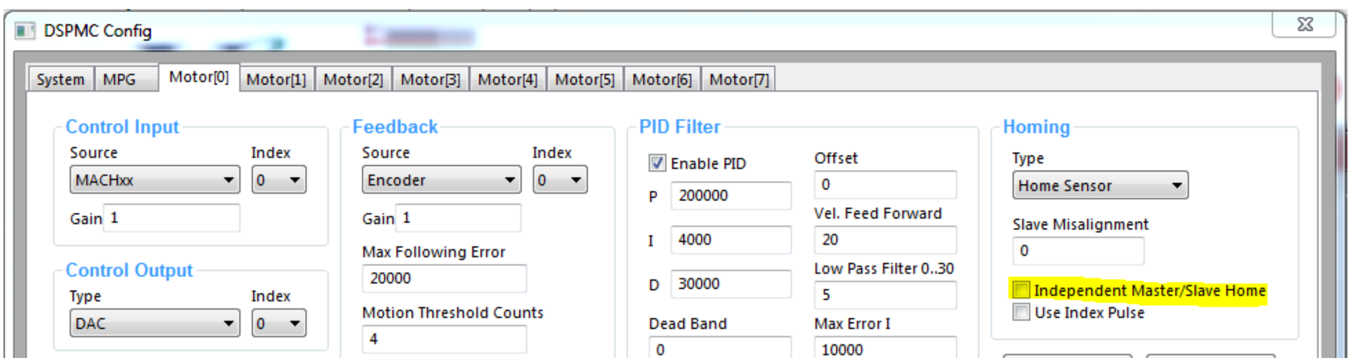
	Enabled	Master	Slave 1	Slave 2	Slave 3	Slave 4	Slave 5
X (0)	✔	Motor0	Motor1				
Y (1)	✘						
Z (2)	✔	Motor2					

In the above setup, Motor1 is set as a slave of Motor0 in the X-axis. Motor1 will follow every move of Motor0 perfectly. Multiple slaves can be assigned to any axis, but a master Motor must always be present.

NOTE: *The axis DRO will show the actual position of the master motor only.*

Dependent Homing

In a Master-Slave setup, Dependent Homing is a setting where the axis is homed using ONLY the master motor's home sensor. The slave motors will follow the master blindly during the whole homing sequence and will stop when the master motor stops. Dependent homing can be enabled by unchecking the "Independent Master/Slave Home" option in the master's plugin motor config (as shown below).



Independent Homing

Independent Homing mode allows the master motor, and all associated slave motors to home (or reference) using their own home sensors (e.g. a home sensor must be mapped from the Mach4 input signals for each motor in the axis). If one or more motors doesn't have a home sensor mapped, then the homing sequence fails and the system is disarmed. Ensure that the "Independent Master" option is checked in the master's plugin motor config.

The following picture shows motor home sensors mapping:

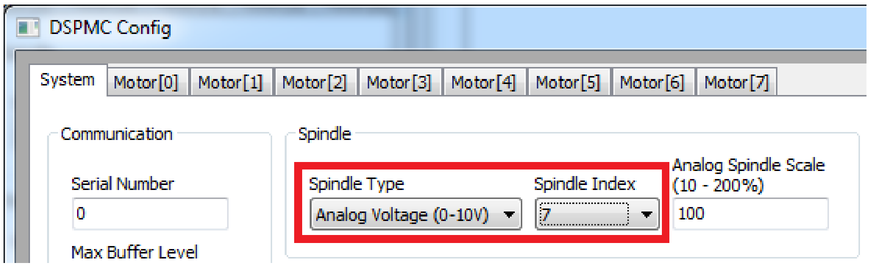
Motor 0 Home	✔	DSPMC	[P11] Input 3	✘
Motor 1 Home	✔	DSPMC	[P11] Input 4	✘
Motor 2 Home	✔	DSPMC	[P11] Input 5	✘

Spindle Setup

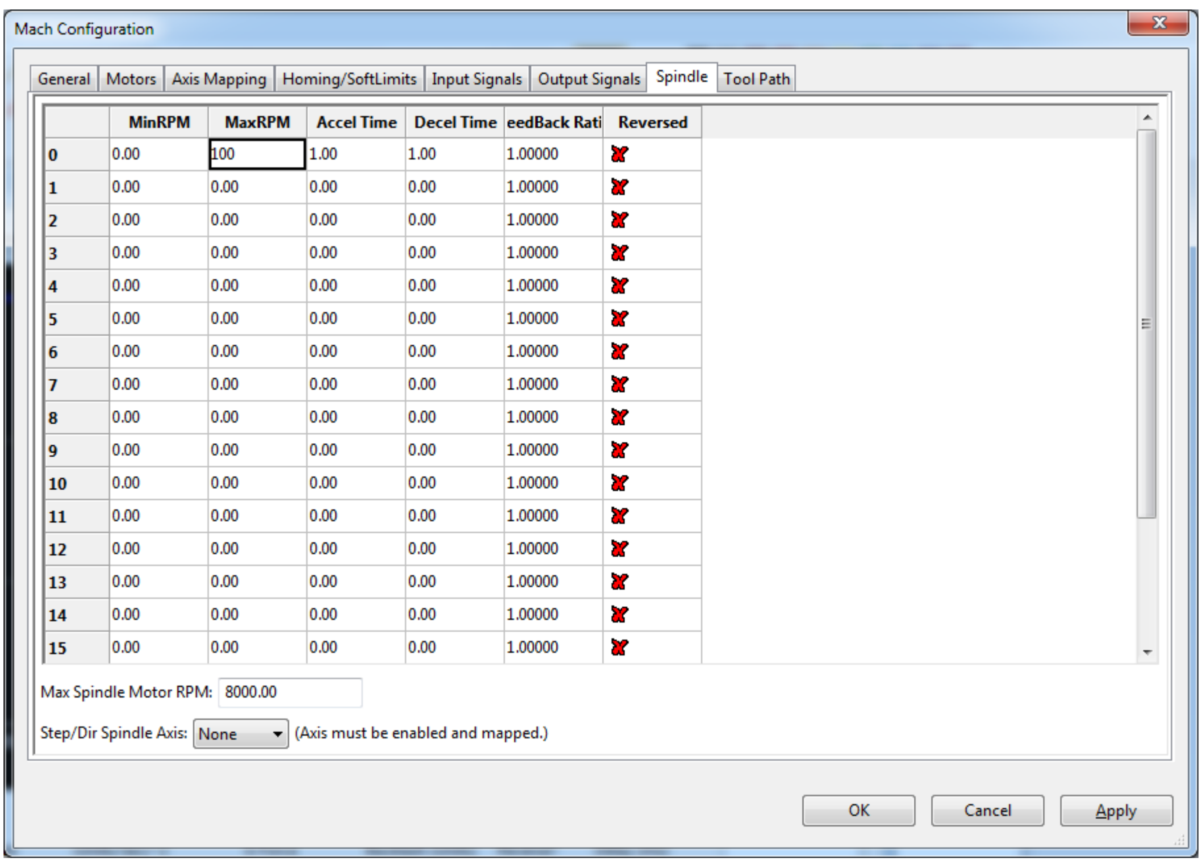
Analog Voltage Spindle

When using a VFD or other motor controlling device that uses 0–10V or ±10V control, the following steps are needed.

1. Set the Spindle Type to "***Analog Voltage 0–10V***" or "***Analog Voltage ±10V***".



2. On the Mach4 Config window, go to the "***Spindle***" Tab. For this example, set "***MinRPM***" to 0 and "***MaxRPM***" to 100. This will output a value of 0V to the spindle at S0 (min RPM) and 10V at S100 (max RPM).



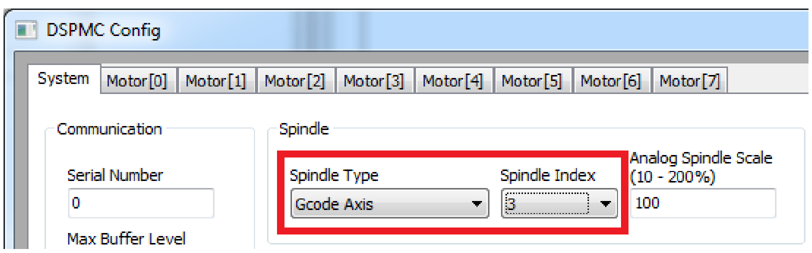
NOTE: *Make sure that the "Spindle On", "Spindle Fwd", and "Spindle Rev" are also mapped in the Output Signals Tab and wired correctly to the DSPMC outputs.*

NOTE: *Without the VFD/Drive hooked up to the DSPMC, the output Volts may be tested with a digital volt meter to ensure that 10V are outputted at max RPM, and 0V at zero RPM.*

After testing, the "MinRPM" can be set to 0, and "MaxRPM" to the spindle motor's max RPM. This allows S commands to be used for the actual motor RPM in GCode.

Spindle Axis

The "GCode Axis" spindle type allows any "position-controlled" axis motor to simultaneously function as a spindle. The "Spindle Index" is used to determine which motor will be selected.



Once the spindle axis has been set, GCode commands can be issued (e.g. "*G00 A10*") for spindle position control, as well as Spindle speed commands (e.g. *"S500 M3"*) to control the spindle speed and direction.

NOTE: *Like all axes, the motor motion parameters (Counts/ unit, Max Speed, and Acceleration) of the Spindle Axis are read from the Mach4 Motor Configuration.*

MPG Setup

See MPG Config in the System Tab for details.

Mach4 DSPMC Registers

The DSPMC plugin defines several Mach4 registers that are primarily used in communicating information for custom user functionality. These registers are included in the DSPMC plugin mostly for developer purposes.

Most motion control setups can skip this section.

> **NOTE**: *VDRO and VLED registers are used as a means of transmitting and receiving data from the DSPMC BASIC macro program.*

VENC (0 – 15) – These ***read-only*** display the current encoder counter position for each encoder channel.

VADC (0 – 7) – These ***read-only*** registers display the voltage readings for each analog input channel.

VDRO (2000 – 2049) – These ***writable*** registers are general-purpose numerical values that are sent to the DSPMC and used in the BASIC Macro Program *(i.e. DROs 2000 – 2049).*

VDRO (2050 – 2099) – These ***read-only*** registers are general-purpose numerical values that are received from the DSPMC and used in the BASIC Macro Program *(i.e. DROs 2050 – 2099).*

VLED (2000 – 2031) – These ***writable*** registers are general-purpose bit values that are sent to the DSPMC and used in the DSPMC BASIC Macro Program *(i.e. LEDs 2000 – 2032).*

VLED (2050 – 2081) – These ***read-only*** registers are general-purpose bit values that are received from the DSPMC and used in the DSPMC BASIC Macro Program *(i.e. DROs 2050 – 2099).*