DSPMC Mach4
Software Integration

Ethernet Motion Controller
Data Acquisition System
PID Controller

Document Revision 1.0
(Updated Sept. 9, 2014)

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 www.machsupport.com

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.

This document assumes that user is familiar with the usage of Mach4 software. This chapter describes the mapping of Mach4 internal software signals to the DSPMC connector.

The general Mach4 software operation remains mostly the same when using DSPMC plugin.

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

Mach4 Configuration

DSPMC Mach4 Plugin Setup

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

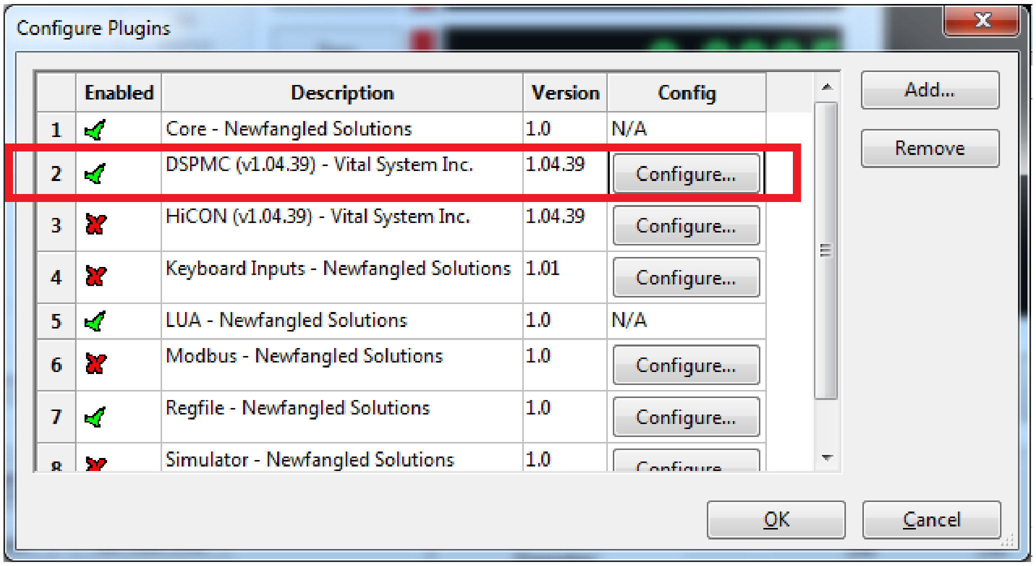
1. Download the Mach4 DSPMC plugin here.

2. Open the zip file and the plugin file “*M4DSPMC.m4pw”* can be found within.

3. To use the DSPMC plugin for Mach4, extract or copy the “*M4DSPMC.m4pw”* file to the Plugins folder in the Mach4 install directory (*usually C:/Mach4/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 screens 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.



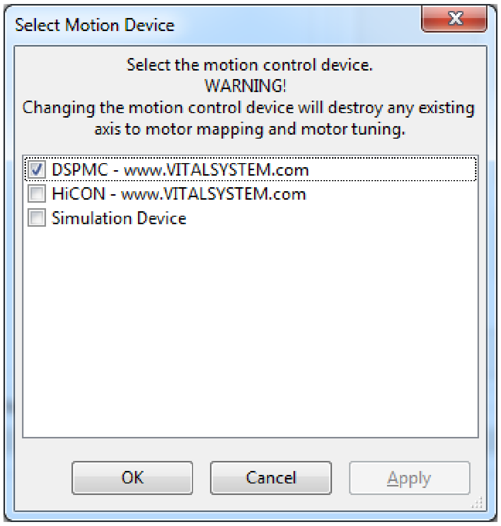
	Enabled	Description	Version	Config
1	✓	Core - Newfangled Solutions	1.0	N/A
2	✓	DSPMC (v1.04.39) - Vital System Inc.	1.04.39	Configure...
3	✗	HiCON (v1.04.39) - Vital System Inc.	1.04.39	Configure...
4	✗	Keyboard Inputs - Newfangled Solutions	1.01	Configure...
5	✓	LUA - Newfangled Solutions	1.0	N/A
6	✗	Modbus - Newfangled Solutions	1.0	Configure...
7	✓	Regfile - Newfangled Solutions	1.0	Configure...
8	✗	Simulator - Newfangled Solutions	1.0	Configure

6. If an “X” is displayed on the “Enabled” column, click it to enable the DSPMC plugin as shown in the image above.

> **NOTE**: *The “Configure” button is used to access the plugin configuration. Take note of this as a large amount of operational parameters are configured from the DSPMC plugin screen.*

7. You may have to restart Mach4 if the DSPMC plugin was disabled.

Starting Mach4 with DSPMC



Select Motion Device

Select the motion control device.
WARNING!
Changing the motion control device will destroy any existing axis to motor mapping and motor tuning.

☑ DSPMC - www.VITALSYSTEM.com
□ HiCON - www.VITALSYSTEM.com
□ Simulation Device

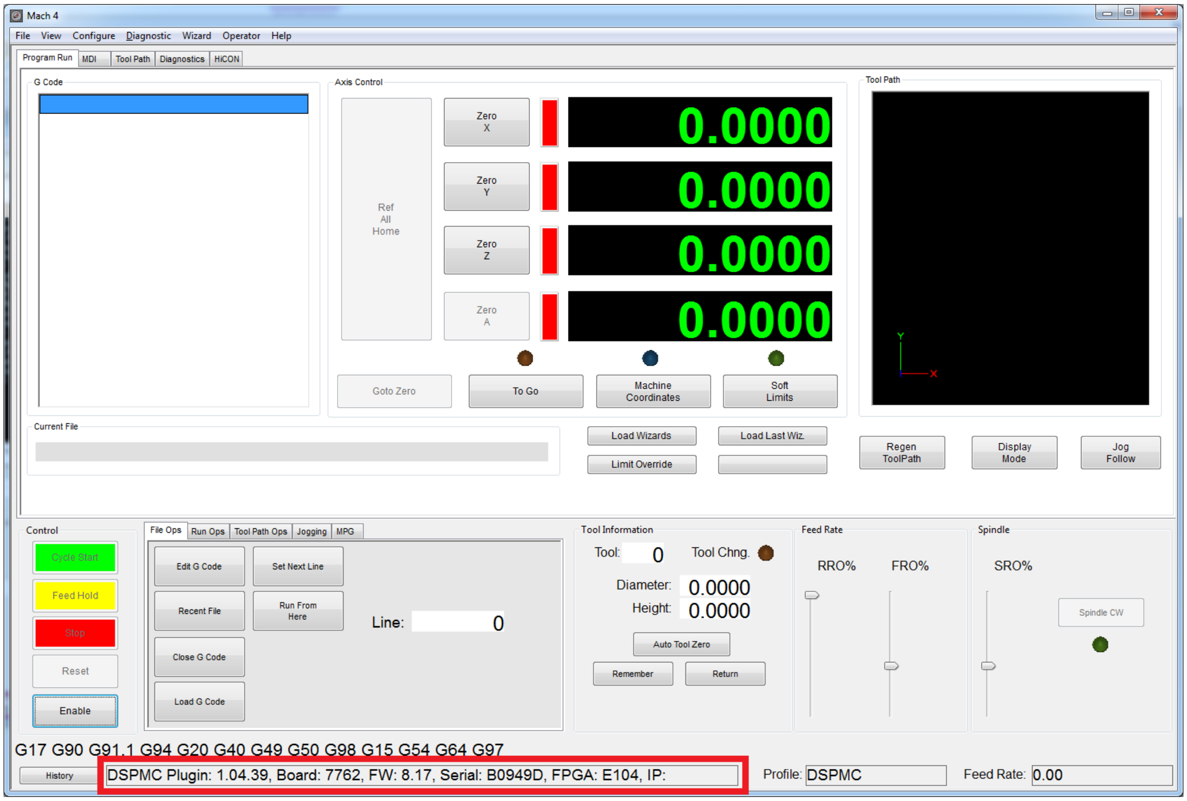
If the steps in the plugin setup were followed correctly, you should be provided with the dialog box to select the motion device with the DSPMC as an option on Mach4 startup. Make sure this plugin is selected and click 'OK'. (*See image to the right*)

> **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.*



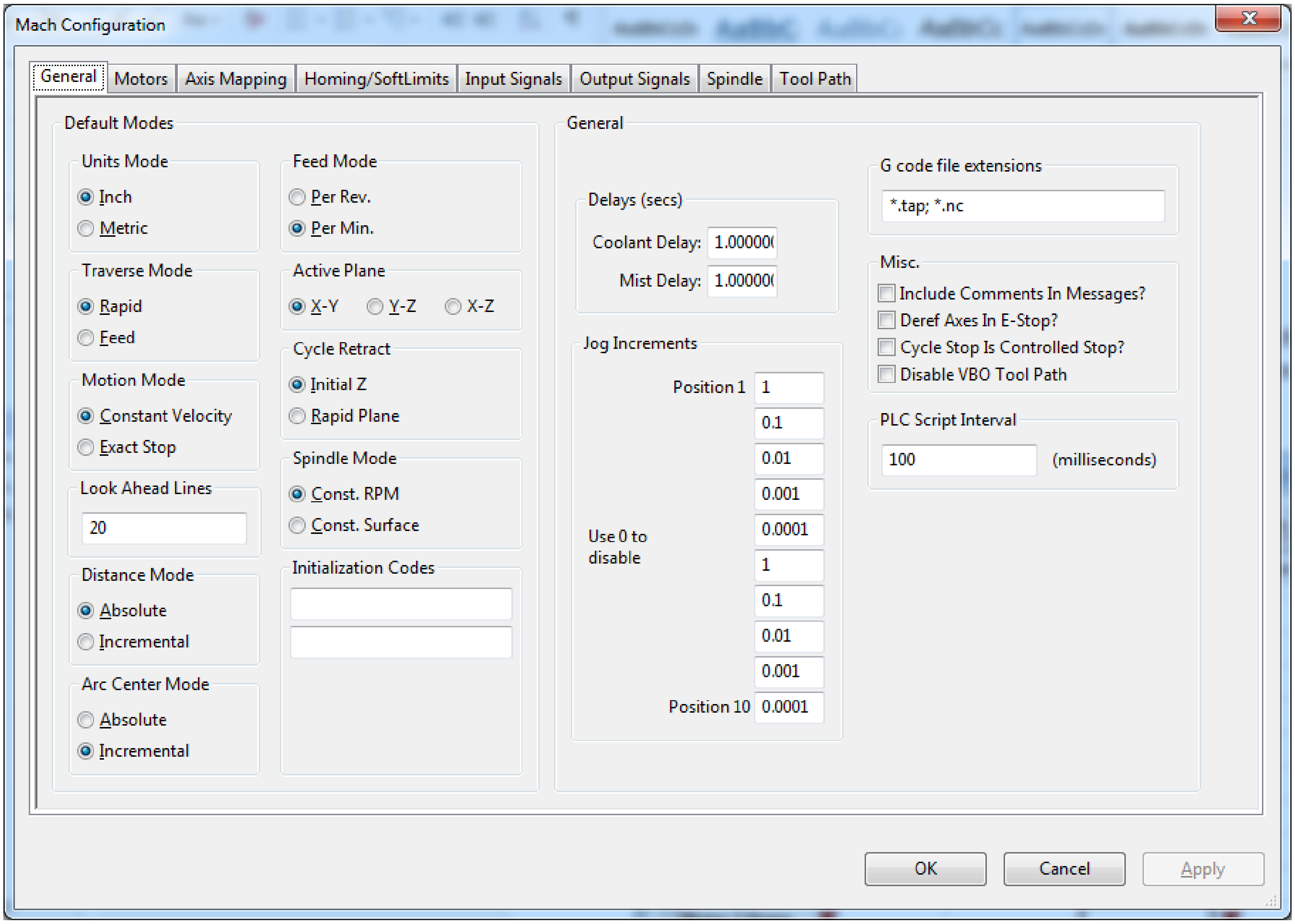
G17 G90 G91.1 G94 G20 G40 G49 G50 G98 G15 G54 G64 G97

DSPMC Plugin: 1.04.39, Board: 7762, FW: 8.17, Serial: B0949D, FPGA: E104, IP:

Profile: DSPMC

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.



Default Modes

Units Mode: ● Inch ○ Metric

Feed Mode: ○ Per Rev. ● Per Min.

Traverse Mode: ● Rapid ○ Feed

Active Plane: ● X-Y ○ Y-Z ○ X-Z

Motion Mode: ● Constant Velocity ○ Exact Stop

Cycle Retract: ● Initial Z ○ Rapid Plane

Look Ahead Lines: 20

Spindle Mode: ● Const. RPM ○ Const. Surface

Distance Mode: ● Absolute ○ Incremental

Initialization Codes:

Arc Center Mode: ○ Absolute ● Incremental

General

Delays (secs)

Coolant Delay: 1.00000(

Mist Delay: 1.00000(

Jog Increments

Position 1	1
	0.1
	0.01
	0.001
Use 0 to disable	0.0001
	1
	0.1
	0.01
	0.001
Position 10	0.0001

G code file extensions: *.tap; *.nc

Misc.

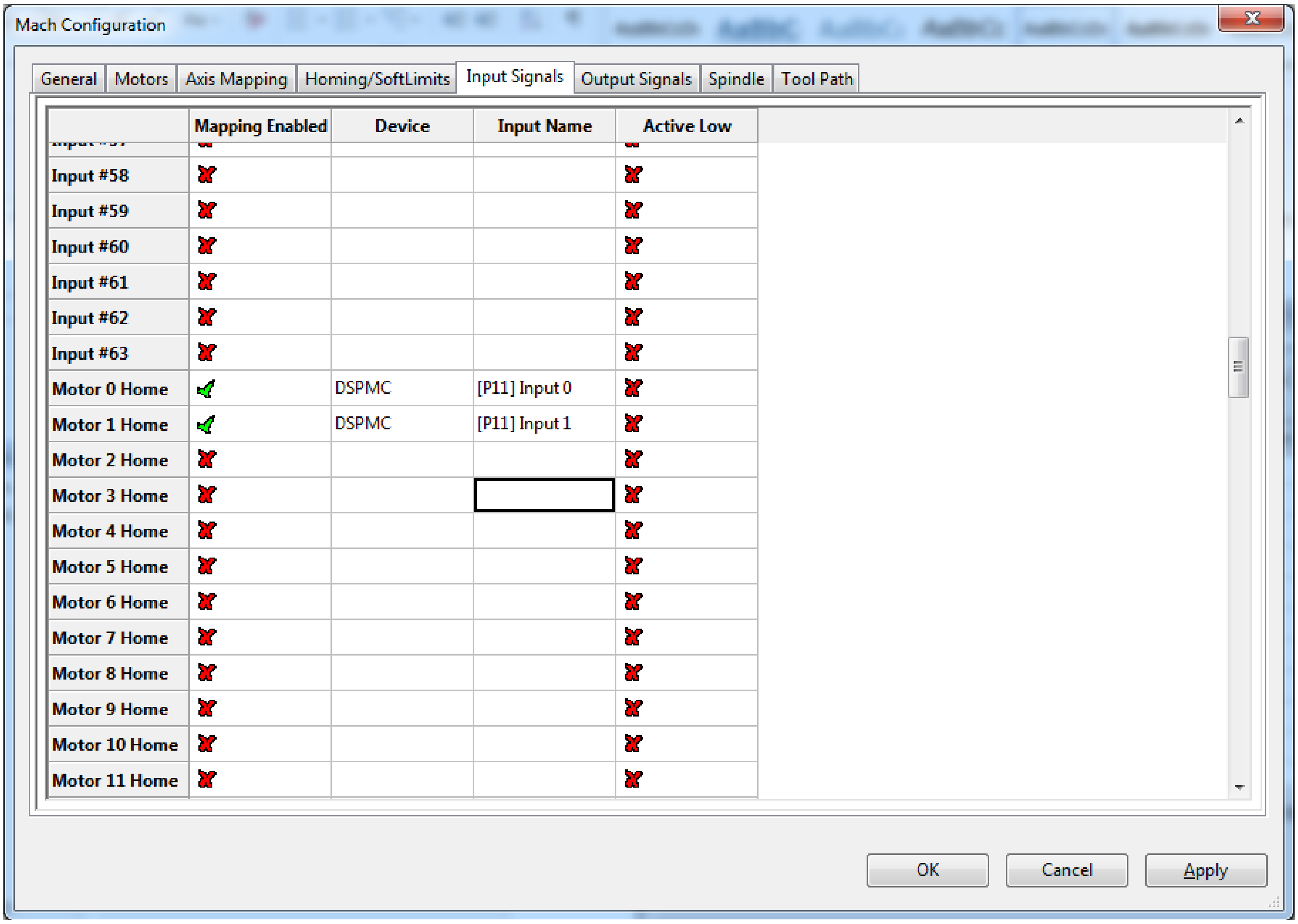
□ Include Comments In Messages?
□ Deref Axes In E-Stop?
□ Cycle Stop Is Controlled Stop?
□ Disable VBO Tool Path

PLC Script Interval: 100 (milliseconds)

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. (*See image below*).



	Mapping Enabled	Device	Input Name	Active Low
Input #57	✗			✗
Input #58	✗			✗
Input #59	✗			✗
Input #60	✗			✗
Input #61	✗			✗
Input #62	✗			✗
Input #63	✗			✗
Motor 0 Home	✓	DSPMC	[P11] Input 0	✗
Motor 1 Home	✓	DSPMC	[P11] Input 1	✗
Motor 2 Home	✗			✗
Motor 3 Home	✗			✗
Motor 4 Home	✗			✗
Motor 5 Home	✗			✗
Motor 6 Home	✗			✗
Motor 7 Home	✗			✗
Motor 8 Home	✗			✗
Motor 9 Home	✗			✗
Motor 10 Home	✗			✗
Motor 11 Home	✗			✗

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 GetDSPData() Index
[P11] Input 0	**18**	0	0
[P11] Input 1	**6**	1	1
[P11] Input 2	**19**	2	2
[P11] Input 3	**7**	3	3
[P11] Input 4	**20**	4	4
[P11] Input 5	**8**	5	5
[P11] Input 6	**21**	6	6
[P11] Input 7	**9**	7	7
[P11] Input 8	**22**	8	8
[P11] Input 9	**10**	9	9
[P11] Input 10	**23**	10	10
[P11] Input 11	**11**	11	11
[P11] Input 12	**24**	12	12
[P11] Input 13	**12**	13	13
[P11] Input 14	**25**	14	14
[P11] Input 15	**13**	15	15

Mach4 Input	DSPMC J5 Input Pin	7535 Breakout Board Input	DSPMC BASIC Macro GetDSPData() Index
[P12] Input 0	**18**	0	16
[P12] Input 1	**6**	1	17
[P12] Input 2	**19**	2	18
[P12] Input 3	**7**	3	19
[P12] Input 4	**20**	4	20
[P12] Input 5	**8**	5	21
[P12] Input 6	**21**	6	22
[P12] Input 7	**9**	7	23
[P12] Input 8	**22**	8	24
[P12] Input 9	**10**	9	25
[P12] Input 10	**23**	10	26
[P12] Input 11	**11**	11	27
[P12] Input 12	**24**	12	28
[P12] Input 13	**12**	13	29
[P12] Input 14	**25**	14	30
[P12] Input 15	**13**	15	31

Mach4 Input	DSPMC J11 Input Pin	7535 Breakout Board Input	DSPMC BASIC Macro GetDSPData() Index
[P13] Input 0	**18**	0	140
[P13] Input 1	**6**	1	141
[P13] Input 2	**19**	2	142
[P13] Input 3	**7**	3	143
[P13] Input 4	**20**	4	144
[P13] Input 5	**8**	5	145
[P13] Input 6	**21**	6	146
[P13] Input 7	**9**	7	147
[P13] Input 8	**22**	8	148
[P13] Input 9	**10**	9	149
[P13] Input 10	**23**	10	150
[P13] Input 11	**11**	11	151
[P13] Input 12	**24**	12	152
[P13] Input 13	**12**	13	153
[P13] Input 14	**25**	14	154
[P13] Input 15	**13**	15	155

Mach4 Input	DSPMC J11 Input Pin	7535 Breakout Board Input	DSPMC BASIC Macro GetDSPData() Index
[P14] Input 0	**18**	0	156
[P14] Input 1	**6**	1	157
[P14] Input 2	**19**	2	158
[P14] Input 3	**7**	3	159
[P14] Input 4	**20**	4	160
[P14] Input 5	**8**	5	161
[P14]Input 6	**21**	6	162
[P14] Input 7	**9**	7	163
[P14] Input 8	**22**	8	164
[P14] Input 9	**10**	9	165
[P14] Input 10	**23**	10	166
[P14] Input 11	**11**	11	167
[P14] Input 12	**24**	12	168
[P14] Input 13	**12**	13	169
[P14] Input 14	**25**	14	170
[P14] Input 15	**13**	15	171

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 GetDSPData() Index
[P11] Output 0	**14**	0	40
[P11] Output 1	**2**	1	41
[P11] Output 2	**15**	2	42
[P11] Output 3	**3**	3	43
[P11] Output 4	**16**	4	44
[P11] Output 5	**4**	5	45
[P11] Output 6	**17**	6	46
[P11] Output 7	**5**	7	47

Mach4 Output	DSPMC J5 Output Pin	7535 Breakout Board Output	DSPMC BASIC Macro GetDSPData() Index
[P12] Output 0	**14**	0	48
[P12] Output 1	**2**	1	49
[P12] Output 2	**15**	2	50
[P12] Output 3	**3**	3	51
[P12] Output 4	**16**	4	52
[P12] Output 5	**4**	5	53
[P12] Output 6	**17**	6	54
[P12] Output 7	**5**	7	55

Mach4 Output	DSPMC J11 Output Pin	7535 Breakout Board Output	DSPMC BASIC Macro GetDSPData() Index
[P13] Output 0	**14**	0	56
[P13] Output 1	**2**	1	57
[P13] Output 2	**15**	2	58
[P13] Output 3	**3**	3	59
[P13] Output 4	**16**	4	60
[P13] Output 5	**4**	5	61
[P13] Output 6	**17**	6	62
[P13] Output 7	**5**	7	63

Mach4 Output	DSPMC J12 Output Pin	7535 Breakout Board Output	DSPMC BASIC Macro GetDSPData() Index
[P14] Output 0	**14**	0	64
[P14] Output 1	**2**	1	65
[P14] Output 2	**15**	2	66
[P14] Output 3	**3**	3	67
[P14] Output 4	**16**	4	68
[P14] Output 5	**4**	5	69
[P14] Output 6	**17**	6	70
[P14] Output 7	**5**	7	71

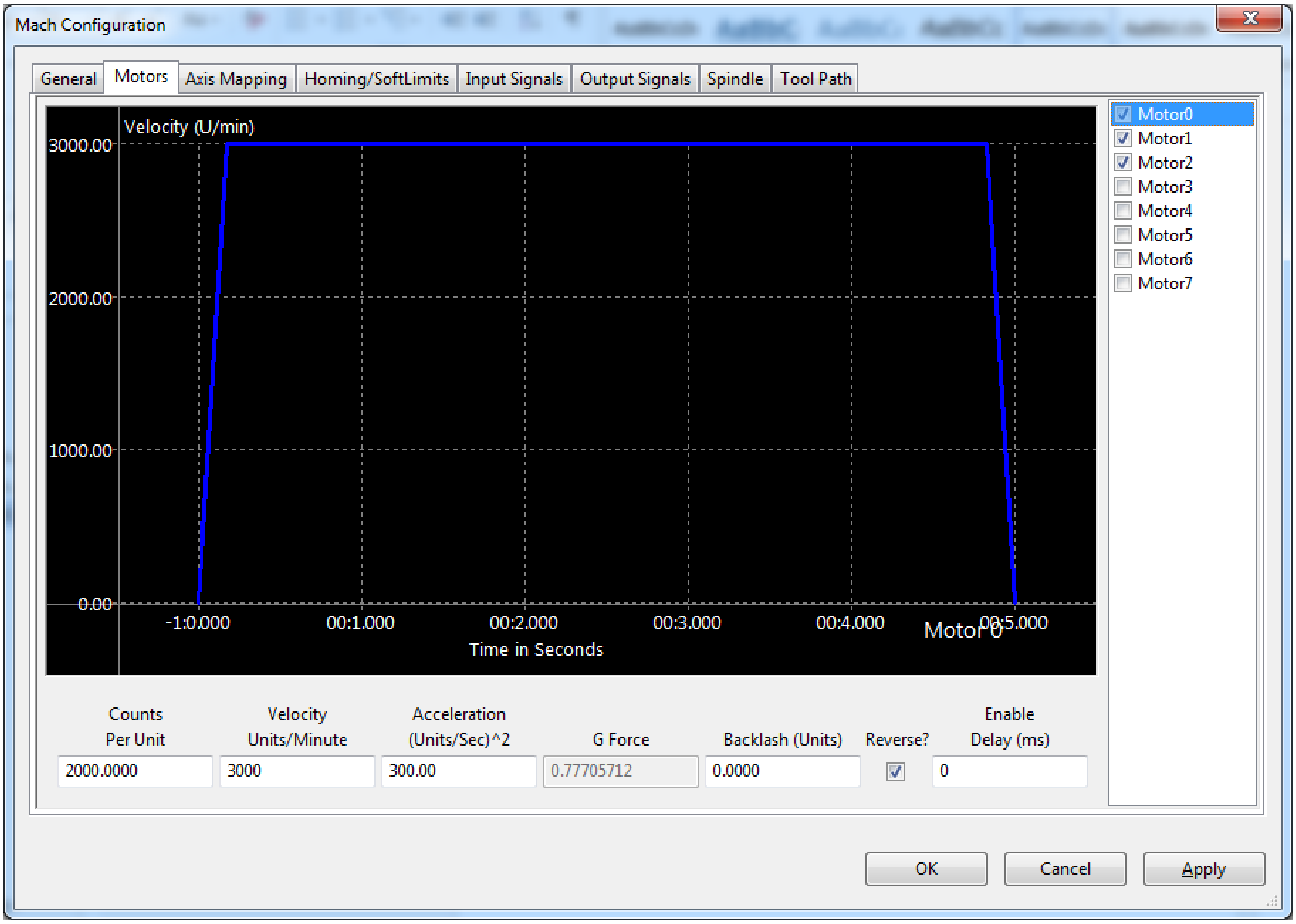
Motor Parameters

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 and Axis Direction are also configured from this 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*



- ☑ Motor0
- ☑ Motor1
- ☑ Motor2
- ☐ Motor3
- ☐ Motor4
- ☐ Motor5
- ☐ Motor6
- ☐ Motor7

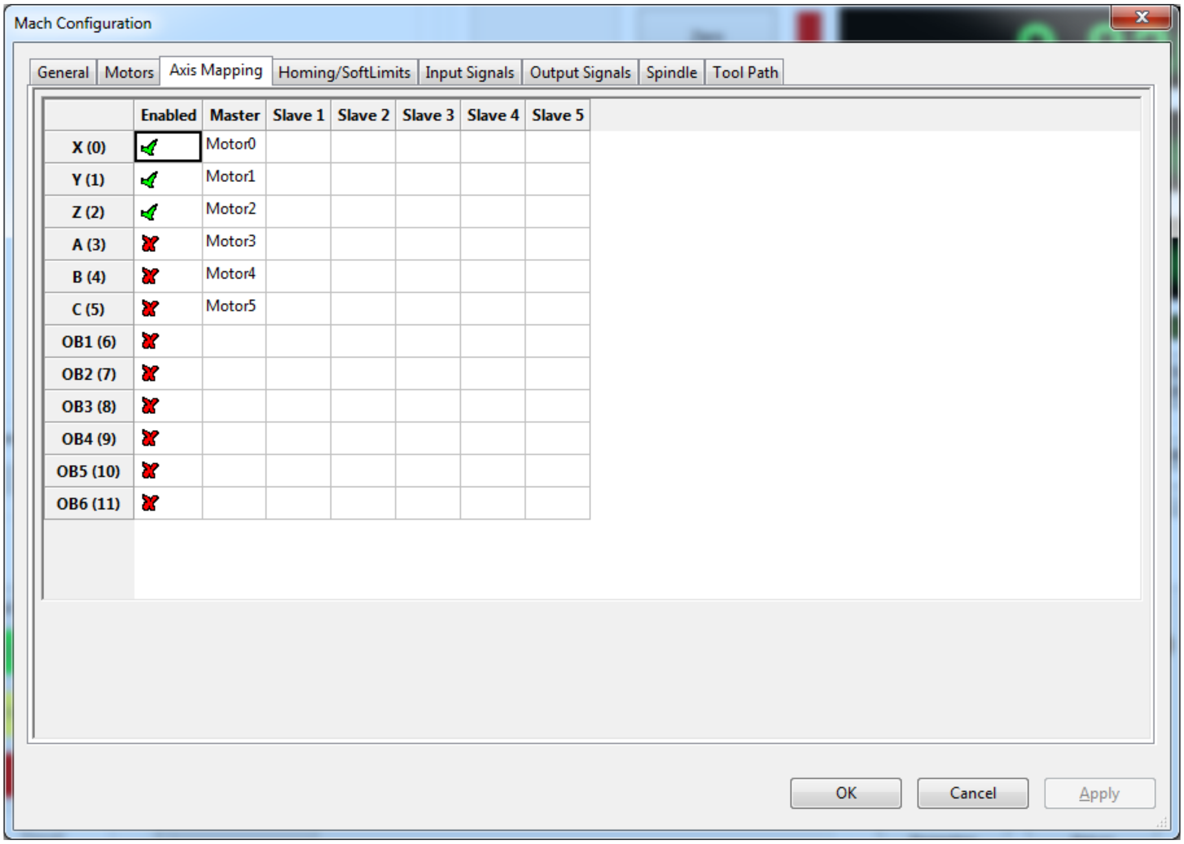
Counts Per Unit	Velocity Units/Minute	Acceleration (Units/Sec)^2	G Force	Backlash (Units)	Reverse?	Enable Delay (ms)
2000.0000	3000	300.00	0.77705712	0.0000	☑	0

Mach4 Axis Mapping

Mach4 marks a clear separation between the definition of an "Axis" and a "Motor". As such, it is improper terminology to interchange the two.

> **<u>NOTE</u>**: *In Mach4, motion is commanded on an axis via GCode commands, MDI, or jogging. An axis can have one or several motors under its control which will be moved according to the commanded motion on the axis.*
>
> *Therefore, an axis is a logical component of the motion vector while a motor is the physical component.*

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.

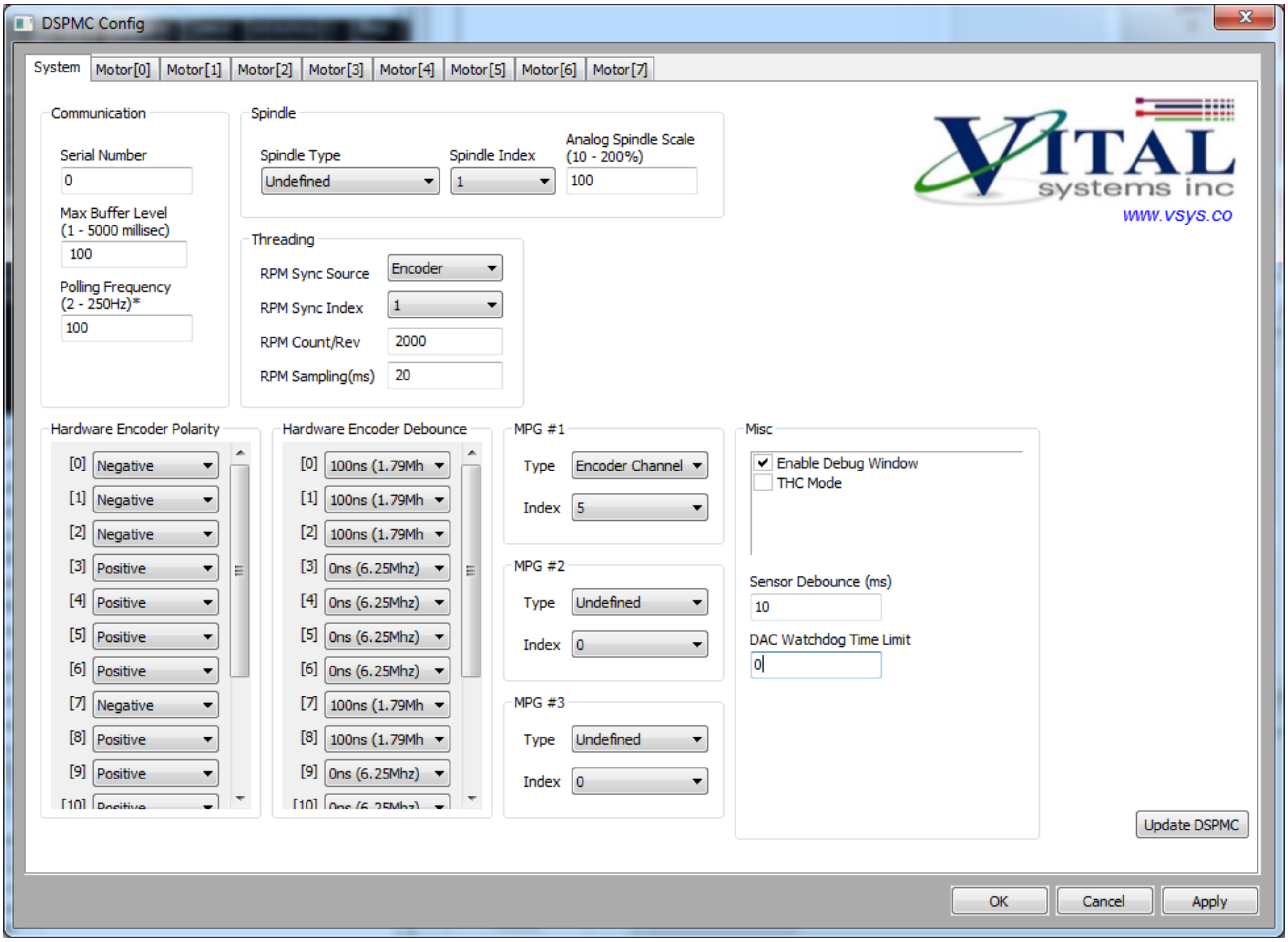


	Enabled	Master	Slave 1	Slave 2	Slave 3	Slave 4	Slave 5
X (0)	✓	Motor0					
Y (1)	✓	Motor1					
Z (2)	✓	Motor2					
A (3)	✗	Motor3					
B (4)	✗	Motor4					
C (5)	✗	Motor5					
OB1 (6)	✗						
OB2 (7)	✗						
OB3 (8)	✗						
OB4 (9)	✗						
OB5 (10)	✗						
OB6 (11)	✗						

From this window, you can quickly enable/disable an axis, set a master and slave motor, and reassign motors to different axes without having to modify the motor configuration in the DSPMC plugin config.

> **<u>NOTE</u>**: *An <u>unmapped motor</u> or a motor mapped to a disabled axis is regarded by the DSPMC as a <u>disabled motor</u>. That being said, the DSPMC will not arm an unmapped motor.*

DSPMC System Configuration



Communication
Serial Number: 0
Max Buffer Level (1 - 5000 millisec): 100
Polling Frequency (2 - 250Hz)*: 100

Spindle
Spindle Type: Undefined
Spindle Index: 1
Analog Spindle Scale (10 - 200%): 100

Threading
RPM Sync Source: Encoder
RPM Sync Index: 1
RPM Count/Rev: 2000
RPM Sampling(ms): 20

Hardware Encoder Polarity		Hardware Encoder Debounce	
[0]	Negative	[0]	100ns (1.79Mh
[1]	Negative	[1]	100ns (1.79Mh
[2]	Negative	[2]	100ns (1.79Mh
[3]	Positive	[3]	0ns (6.25Mhz)
[4]	Positive	[4]	0ns (6.25Mhz)
[5]	Positive	[5]	0ns (6.25Mhz)
[6]	Positive	[6]	0ns (6.25Mhz)
[7]	Negative	[7]	100ns (1.79Mh
[8]	Positive	[8]	100ns (1.79Mh
[9]	Positive	[9]	0ns (6.25Mhz)
[10]	Positive	[10]	0ns (6.25Mhz)

MPG #1
Type: Encoder Channel
Index: 5

MPG #2
Type: Undefined
Index: 0

MPG #3
Type: Undefined
Index: 0

Misc
☑ Enable Debug Window
□ THC Mode
Sensor Debounce (ms): 10
DAC Watchdog Time Limit: 0

In the system tab, you can set a number of configurations. Clicking on the Update DSPMC will transmit these settings to the DSPMC controller. Clicking OK will also transmit the data, and also save this data in the selected Mach4 profile (e.g. Mach4mill, Mach4turn etc.)

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 preferred 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 total size of the buffer is 4096 motion vectors (a motion vector is a vector of all axes positions in a millisecond resolution). These vectors are consumed by the DSPMC at 1 KHZ.

  Lower values make the motion more responsive to feedrate changes or feedholds, but more prone to jerking due to premature emptying of the motion buffer. 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. **Recommended is 100 – 500 and 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 Mach 3 used a Frequency of 10Hz to exchange data, the recommended value for Mach4 is 100Hz.

Spindle Type and Index

The ***Spindle Type*** configuration can be set to the following:

- ***Undefined*** – Select this if a spindle is not used.

- ***Analog Voltage 0-10V*** – Most common for spindles driven by a VFD. Uses Spindle Relay outputs for direction *(index indicates which DAC output channel to use).*

- ***Analog Voltage ±10V*** – Bi-directional analog voltage *(index indicates which DAC output channel to use)*.

- ***GCode Axis*** – Sets Mach-controlled motor to be used as a spindle. Typically used for spindles that are controlled by Step/Dir signals instead of Analog Voltage (*index determines the motor number to use*)

NOTE: For spindles using Analog Voltage, make sure the selected DAC output number is not used in any other axis' Control Output Index.

Threading

- ***Threading RPM Synch Source –*** This parameter defines the encoder type for Spindle speed calculation and starting the threading cycle. The Index pulse from the encoder is used to launch the Z-Axis at the right time in order to position the tool correctly for Threading in every cycle. The RPM calculation is used to override the feedrate of the Z-Axis during the threading cycle.

  Two possible values for **RPM Sync Source** parameter are: ***HardEncoder*** or ***Undefined***.

  - **Undefined** – When this option is selected DSMPC will not enable threading and value of **RPM Synch Index** will be ignored.

- **HardEncoder** – The spindle feedback encoder must be connected to one of the encoder inputs on J3, J6, J7 and J8 connectors. The encoder's differential A and B signals are used to calculate the RPM of the spindle, and Index pulse is used to trigger the threading cycle.

- ***Threading RPM Synch Index –*** This parameter defines the encoder index for Spindle speed feedback. Below is the range for this index:
  HardEncoder: index range is 0...7.

- ***Threading RPM Count/Rev –*** This parameter defines the encoder resolution in terms of count per revolution for Spindle speed feedback. For ***HardEncoder*** type encoder, the encoder resolution must is multiplied by 4. No multiplication is done when ***DigitalInput*** is selected.

- ***Threading RPM Sampling (ms) –*** This parameter defines the timing window in milliseconds to add the encoder counts for RPM calculation. For slow pulse train (eg only few ticks per rev), this value should be high enough to accumulate enough counts to calculate RPM consistently. If the window time is too long, the system reaction time (regulation of Z-Axis feedrate) to changing RPM will be slow. A higher count/rev encoder will allow this window time to be very small, which will allow the system to react fast (regulate Z-Axis feedrate) if RPM changes. The range of this field is from 1 thru 10000 milliseconds.

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 signals that are triggered internally on the DSPMC (e.g. Estop, Limit, Home switches). A higher value will cancel out more noise. This value cannot exceed 1000ms.

- ***DAC Watchdog Time Limit*** – When a "runaway axis" 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 config or reversed wiring on the encoders. *Setting this value to zero, disables the watchdog*.

Manual Pulse Generation (MPG)

- ***MPG Source Type –*** This section defines MPG (Manual Pulse Generation) Quadrature encoder source. Both Differential and Single Ended Encoder types are supported.

  - **Hard Encoder** – This type corresponds to the differential encoders hooked up to any of the six encoder channels on connectors J3, J6, J7 and J8.
  - **Soft Encoder** – Single Ended encoders that can be hooked up to the Digital Inputs available on Connector J5.
  - **Undefined –** This option causes DSMPC to ignore MPG Source values.

- ***MPG Source Index –*** If Hard Encoder is selected in MPG Source Type, MPG Index denotes the hardware encoder index. If Soft Encoder is selected, MPG Index denotes available digital input options.

The Soft Encoder pin assignments are as follows:

Soft Encoder 0	A on J4 Pin 24 B on J4 Pin 12
Soft Encoder 1	A on J4 Pin 25 B on J4 Pin 13
Soft Encoder 2	A on J5 Pin 18 B on J5 Pin 6

Hardware Encoder Polarity

The ***Hardware Encoder Polarity*** field is used to reverse the direction of the encoder counters. If the A/B encoder signals are connected in reverse such that it does not match the PID/Axis 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: *This encoder polarity setting only swaps the A and B signals to change the counter direction. The Index pulse signal polarity is not affected by this setting.*

Encoder Debounce

The ***Encoder debounce*** field is used to filter noise from the hardware encoder signals. Lower debounce values filter less noise, while higher values reduce the maximum frequency of the encoder signals. While a recommended setting of *"100ns"* is normally sufficient, higher values may also be used if the noise persists.

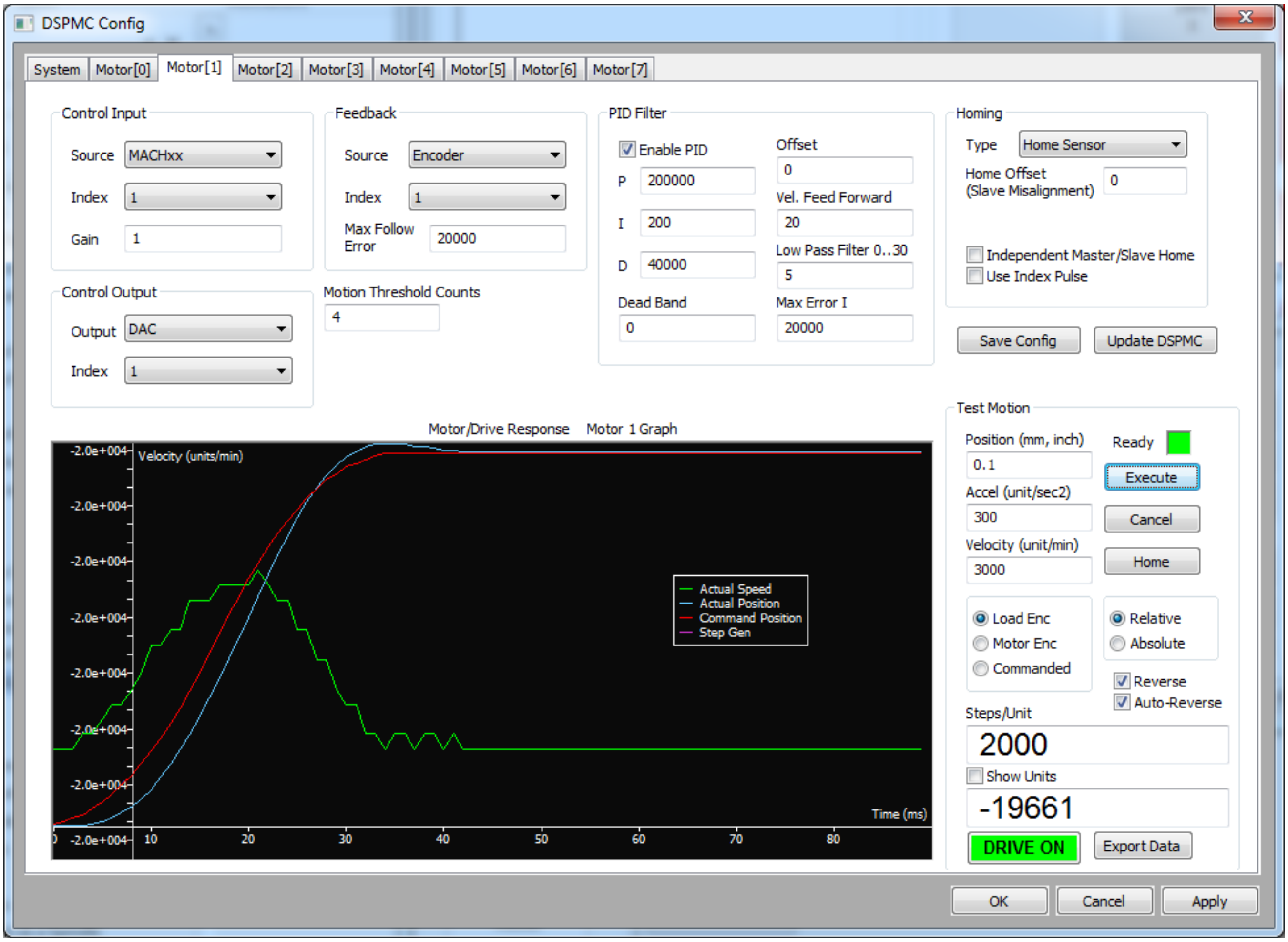
NOTE: *This setting does not apply for Soft Encoders.*

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.*

DSPMC Motor Configuration



Control Input

Source MACHxx

Index 1

Gain 1

Control Output

Output DAC

Index 1

Feedback

Source Encoder

Index 1

Max Follow Error 20000

Motion Threshold Counts 4

PID Filter

☑ Enable PID

P 200000

I 200

D 40000

Dead Band 0

Offset 0

Vel. Feed Forward 20

Low Pass Filter 0..30 5

Max Error I 20000

Homing

Type Home Sensor

Home Offset (Slave Misalignment) 0

□ Independent Master/Slave Home

□ Use Index Pulse

Test Motion

Position (mm, inch) 0.1

Accel (unit/sec2) 300

Velocity (unit/min) 3000

Ready 🟩

● Load Enc

○ Motor Enc

○ Commanded

● Relative

○ Absolute

☑ Reverse

☑ Auto-Reverse

Steps/Unit 2000

□ Show Units

-19661

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 (Analog voltage output) 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*** - Control Input Source defines the input type (or set-point) for the PID filter for a particular axis. This should be set to **MACHxx**. If the axis is not used, it must be disabled by selecting **undefined**.

- ***Control Input Index -*** Defines the index of the PID input source. Normally this is equal to the axis number. For slave axis, it should be set to the number of the master axis.

- ***Control Input Gain –*** The control input (Commanded) is multiplied by this number before applying to PID filter.

Control Output Parameters

- ***Control Output Source -*** Control Output Type defines the output for the PID filter for a particular axis. The possible values are:

  - ***DAC:*** Use one of the analog outputs as the PID control output. This setting is used to drive a Servo amplifier that takes +/-10volt reference inputs.
  - ***Stepper:*** Use one of the dedicated digital output pairs for the Step and Direction signals used in stepper drives.
  - ***Undefined:*** This setting is used to disable the axis and to ignore the control output index. If the axis is not used, the Control Output Source must be set to **undefined**.

- ***Control Output Index*** - Defines the output channel where the commanded motion will be sent to.

Feedback Parameters

- ***Feedback Source -*** PID Feedback Source defines the feedback type for the PID filter for the selected axis. The possible values are:

  - ***Encoder:*** Use one of the differential hardware encoder 0…7 as the PID feedback.

  - ***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.

- ***Feedback Index -*** Selects the index of the feedback source.

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

NOTE: *Make sure not to use 0 in the Max-Following-Error (MFE) field. Always use a non-zero positive number. If it is set to 0, the motor can move at max uncontrolled speed (in a run-away situation), which can be extremely dangerous.*

This field can be back calculated from the maximum velocity of the axis. Below is an example that uses 600,000 counts/sec for max speed.

- The DSPMC checks the max following error every millisecond.
- A max speed of 600,000 counts/sec is equivalent to 600 counts/ms.
- Therefore, to allow a 600K count/sec speed, the max following error should be 600 or more.

NOTE: *The required Max Follow Error value may be lot more than the result of the above calculation based on how tight the PID tuning is, and the mechanical characteristics of the axis.*

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.

Below is a description of the PID Parameters.

- ***P*** – Proportional Gain
- ***I*** – Integral Gain
- ***D*** – Derivative Gain
- ***Max Error I*** – Maximum Integral Error for the integral gain. This must be greater than 0 if you put any non-zero value in the “I” term.
- ***Dead band*** – a range of position 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 DSPMC analog output channels (J2, DAC 0 – 7).

- ***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 less noisy. As the PID runs at 5 KHz, a value of 5 will create a nice linearly increasing DAC output at 1 KHz. If you put a value of 10, effective PID speed will become 500 Hz.

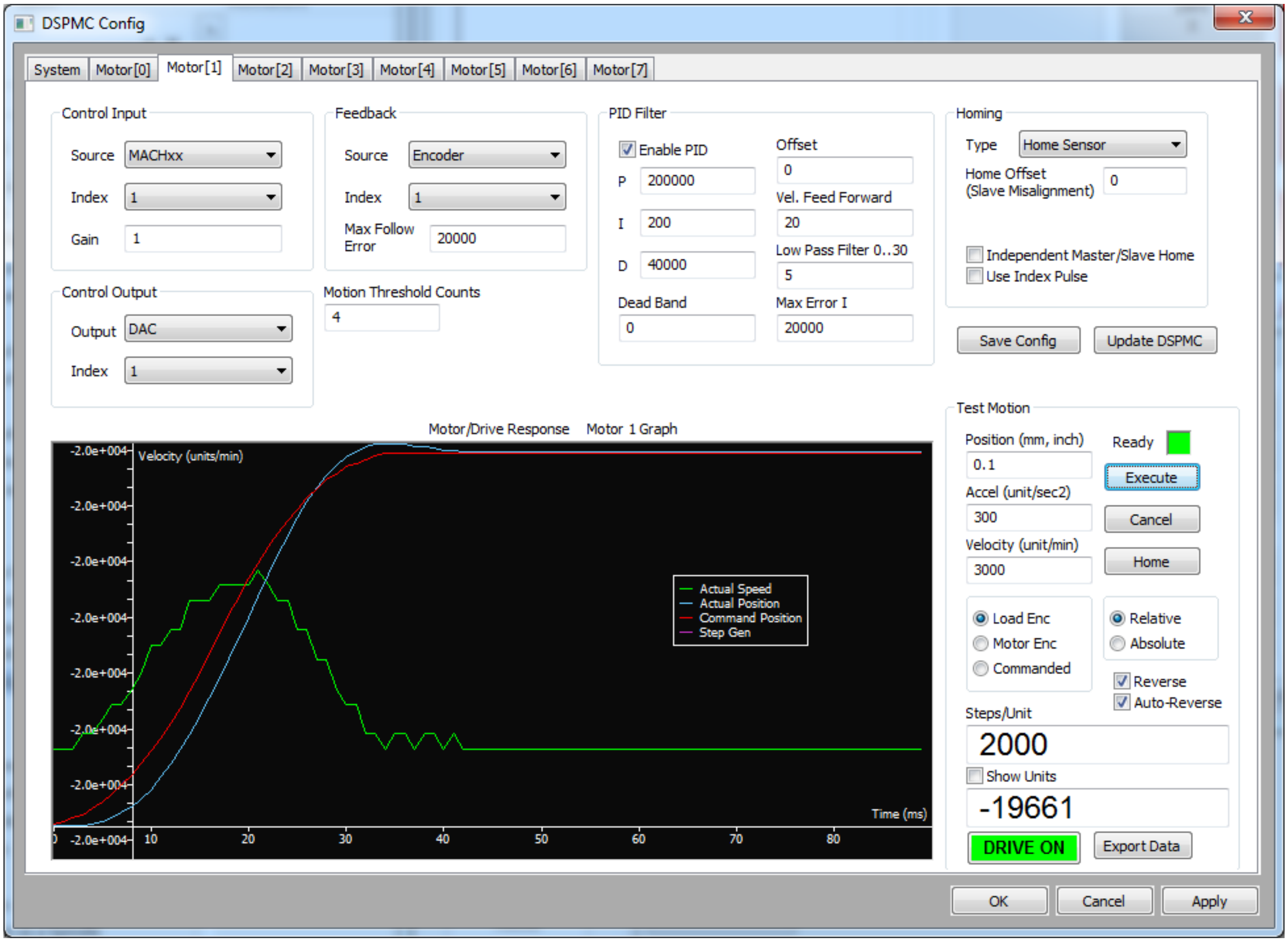
More information on other PID terms and general discussion on PID control is available at: http://en.wikipedia.org/wiki/PID_control.

PID Tuning Steps

1. Set "P" to 100, "Max Follow Error" at 5000, low pass at 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 units. *(Refer to the Test Motion Section)*
3. If no motion is observed, start increasing "P" until you do see motion. Check the Response graph for details on the motion.
4. If the actual position moves is in the opposite direction of commanded, change the encoder polarity in the system tab.
5. Once you get a decent graph, (i.e. actual position follows commanded), start adding *"I"*, *"Max Error I"*, *"D"* and *"Velocity Feed Forward"* terms to fine tune the graph.

> **NOTE**: *When you set the "I" term greater than 0, make sure "Max Error I" is also non-zero, otherwise you will get uncontrolled oscillations, (e.g. start "I" with 5 and "MaxError I" as 100, and gradually increase or decrease).*

6. Ideally, the actual graph line should be as close to commanded as possible. Do the test motions for small and large moves, i.e. few mm to 100 of mms.



Control Input
Source: MACHxx
Index: 1
Gain: 1

Feedback
Source: Encoder
Index: 1
Max Follow Error: 20000

PID Filter
☑ Enable PID
P: 200000
I: 200
D: 40000
Dead Band: 0
Offset: 0
Vel. Feed Forward: 20
Low Pass Filter 0..30: 5
Max Error I: 20000

Homing
Type: Home Sensor
Home Offset (Slave Misalignment): 0
☐ Independent Master/Slave Home
☐ Use Index Pulse

Control Output
Output: DAC
Index: 1

Motion Threshold Counts: 4

Test Motion
Position (mm, inch): 0.1
Ready 🟩
Accel (unit/sec2): 300
Velocity (unit/min): 3000
● Load Enc
○ Motor Enc
○ Commanded
● Relative
○ Absolute
☑ Reverse
☑ Auto-Reverse
Steps/Unit: 2000
☐ Show Units
-19661

(Ideal PID Tuning Graph)

Test Motion

Test Motion component can be utilized to accurately tune the PID Filter Parameters and configure the Control parameters.

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 once the current motion profile has completed or has been stopped

Once the test motion is complete, you can see how closely the axis followed the commanded motion profile on the PID Response graph. The PID parameters may be tweaked and execute the test motion to verify the behavior. If the AutoReverse 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.

- ***Position*** – Test motion final position or displacement in terms of Position Units, (e.g. 1.5, 10.093, mm or inches etc.)

- ***Acceleration*** – Test motion acceleration value in terms of Units per second squared, (e.g. inches/second^2, mm/sec^2 etc.)

- ***Velocity*** – Test motion velocity value in terms of Units per minute, e.g. inches/minute, mm/minute etc.

- ***Relative* and *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*** – Transmits Execute-Motion command to DSPMC. In addition, it also downloads PID Filter parameters before starting the motion. User can press 'Cancel' button to cancel the motion execution anytime during the machine operation. Make sure you have downloaded the axis controls setting by clicking "Update DSPMC" before clicking on Execute.

- The Ready LED shows if the current motion command is completed and DSPMC is ready for new motion command. New motion command can be launched by Execute button when the Ready LED is Green. If the LED goes to Red after click on Execute, but you do not observe any motion, the velocity or acceleration may be too low.

- ***PID Arm Button*** – By clicking this button, the Plugin download's PID Filter parameters and arm or disarm the PID. If PID is armed, the LED below this button will turn to GREEN**,** otherwise it will be RED**.**

- ***Home Button*** – Executes the Homing sequence based on selected Homing settings. Review section 7.12.3 Control Parameters to configure Mach4 Homing options for each axis before executing Homing.

- ***Reverse* -** Checking this option will multiply parameter in the position box with -1 and thus direction of motion will be reversed.

- ***Auto Reverse* -** Checking auto reverse option will toggle "reverse" option between two consecutive motion commands, thus the user do not have to manually reverse the direction of motion every time.

- ***Axis Position Display (DRO) –*** Shows the position of the axis based on certain settings as described below:

  - **Show units -** When this option is selected, the data shown will be converted and shown in units (mm, inches etc), otherwise data will be displayed in raw encoder counts.
  - **Commanded position -** Display shows the value of the internal variable for the commanded position for the selected axis.
  - **Load Encoder -** Display shows the axis position derived from backlash count and selected feedback encoder.
  - **Motor Encoder -** Display shows the current value of the axis position derived only from the encoder feedback.

> **NOTE**: *The actual position may slightly deviate from the command position when PID is enabled.*

Homing Parameters

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

  With the ***Home Sensor*** method, the axis is commanded to move in the configured homing direction until the motor's home sensor is activated. It then moves in reverse at 20% of initial speed until the home sensor is deactivated. One of the following steps is performed depending if the ***Use Index Pulse*** option is selected or not.

  - If the axis 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 switches can also be used as home sensors.*

  For ***IndexPulseOnly***, the axis moves in the configured direction to locate the Encoder Index Pulse to home the axis. 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.

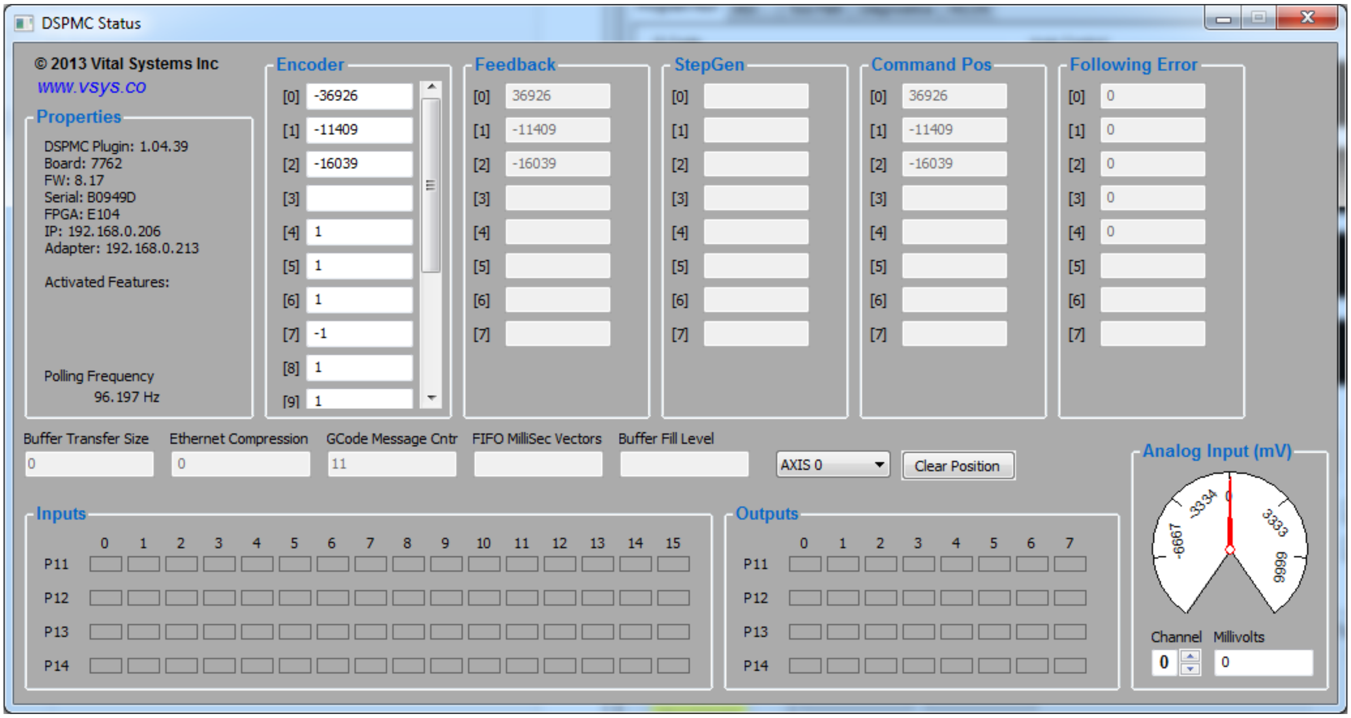
  NOTE: *This value is only used if the selected motor is mapped as a slave in the Mach4 Axis Mapping.*

- ***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 inputs.*

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.



DSPMC Status

© 2013 Vital Systems Inc
www.vsys.co

Properties

DSPMC Plugin: 1.04.39
Board: 7762
FW: 8.17
Serial: B0949D
FPGA: E104
IP: 192.168.0.206
Adapter: 192.168.0.213

Activated Features:

Polling Frequency
96.197 Hz

	Encoder	Feedback	StepGen	Command Pos	Following Error
[0]	-36926	36926		36926	0
[1]	-11409	-11409		-11409	0
[2]	-16039	-16039		-16039	0
[3]					0
[4]	1				0
[5]	1				
[6]	1				
[7]	-1				
[8]	1				
[9]	1				

Buffer Transfer Size	Ethernet Compression	GCode Message Cntr	FIFO MilliSec Vectors	Buffer Fill Level
0	0	11		

AXIS 0
Clear Position

Analog Input (mV)
Channel 0
Millivolts 0

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 I/O states. This information must be consulted when diagnosing dysfunctional Mach4 signal states (e.g. an input isn’t turning on correctly, or estop signal always active), and to determine if an input/output is turning ON/OFF correctly.

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

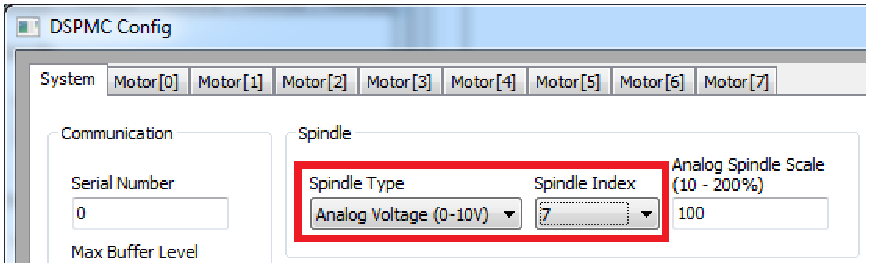
Additional Setup

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*".



Serial Number: 0

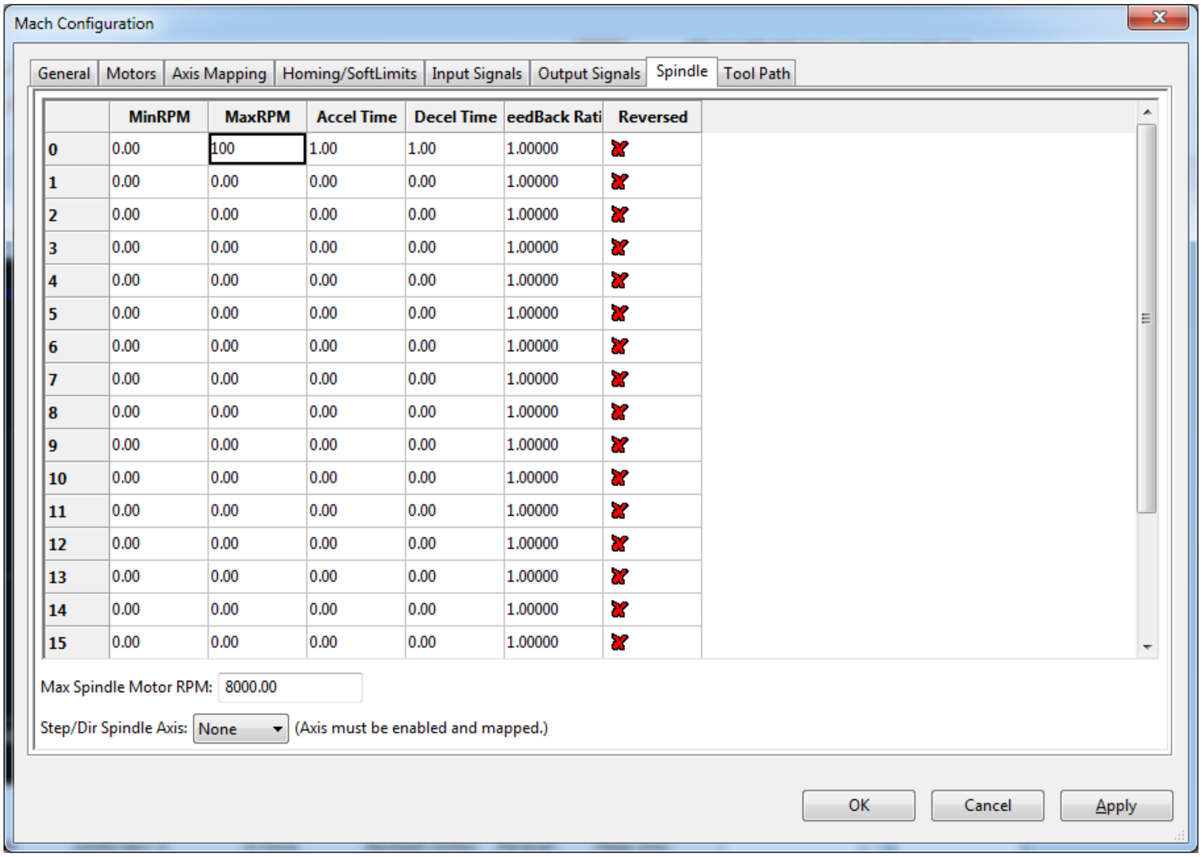
Max Buffer Level

Spindle Type: Analog Voltage (0-10V)

Spindle Index: 7

Analog Spindle Scale (10 - 200%): 100

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



	MinRPM	MaxRPM	Accel Time	Decel Time	eedBack Rati	Reversed
0	0.00	100	1.00	1.00	1.00000	✗
1	0.00	0.00	0.00	0.00	1.00000	✗
2	0.00	0.00	0.00	0.00	1.00000	✗
3	0.00	0.00	0.00	0.00	1.00000	✗
4	0.00	0.00	0.00	0.00	1.00000	✗
5	0.00	0.00	0.00	0.00	1.00000	✗
6	0.00	0.00	0.00	0.00	1.00000	✗
7	0.00	0.00	0.00	0.00	1.00000	✗
8	0.00	0.00	0.00	0.00	1.00000	✗
9	0.00	0.00	0.00	0.00	1.00000	✗
10	0.00	0.00	0.00	0.00	1.00000	✗
11	0.00	0.00	0.00	0.00	1.00000	✗
12	0.00	0.00	0.00	0.00	1.00000	✗
13	0.00	0.00	0.00	0.00	1.00000	✗
14	0.00	0.00	0.00	0.00	1.00000	✗
15	0.00	0.00	0.00	0.00	1.00000	✗

Max Spindle Motor RPM: 8000.00

Step/Dir Spindle Axis: None (Axis must be enabled and mapped.)

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.*

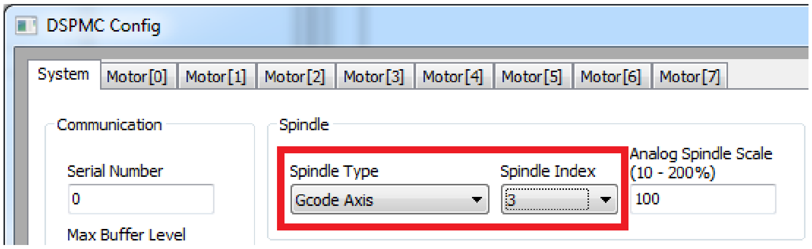
NOTE: *Without the VFD/Drive hooked up to the DSPMC, you can test your output with a digital volt meter to make sure you are getting 0V when the spindle is turned off or set to zero RPM, and 10V for max RPM.*

After testing, you can then set the "MinRPM" to 0, and "MaxRPM" to your spindle motor's max RPM. This will allow you to program S commands for actual RPM in the G-code.

On the Plugin Config System tab, you can configure the spindle to use ±10V or 0-10V by selecting the Spindle Type.

Spindle Axis

You may configure any Mach4 axis as a spindle axis by setting the Spindle Type as a "GCode Axis" and the "Spindle Index" as the axis number.



Serial Number: 0

Max Buffer Level

Spindle Type: Gcode Axis

Spindle Index: 3

Analog Spindle Scale (10 - 200%): 100

Once the spindle axis has been set, you may then issue any GCode command (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 using the closed loop axis PID control.

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

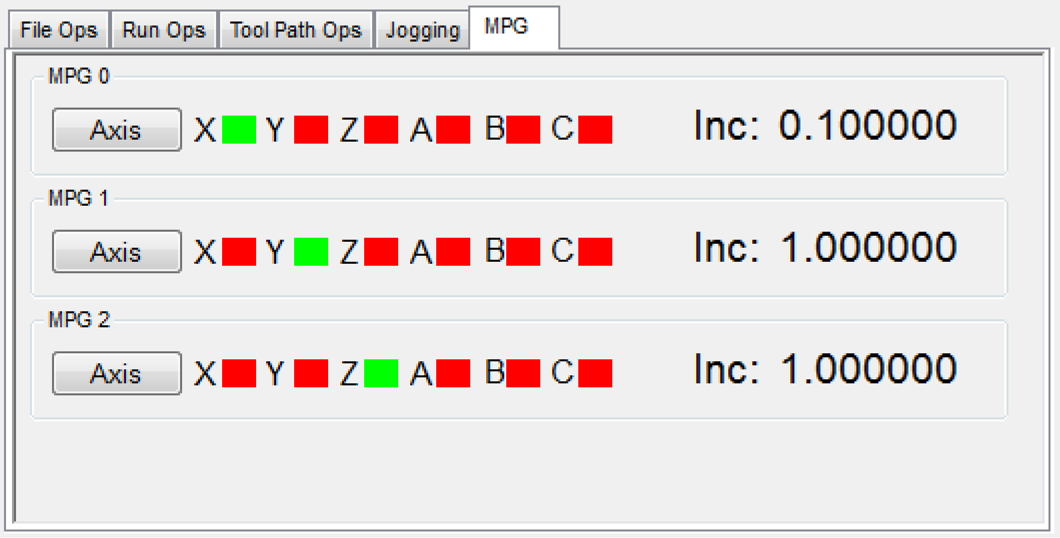
MPG Setup

See MPG Config in the System Tab for details.

In Mach4, jogging an axis with a Handwheel/Pendant only requires the MPG parameters in the System Tab of the DSPMC Plugin are configured.

MPG jogging is enabled anytime the Keyboard/On-screen jogging is allowed (e.g. GCode File Running, no active motion sequences such as MDI or homing are being performed).

> **NOTE**: *An interface for selecting the jog axis and jog step size must be supported by your current screen. The sample Mach4 screen provided on the Vital System Inc. website has such an interface.*



MPG 0

X 🟩 Y 🟥 Z 🟥 A 🟥 B 🟥 C 🟥 Inc: 0.100000

MPG 1

X 🟥 Y 🟩 Z 🟥 A 🟥 B 🟥 C 🟥 Inc: 1.000000

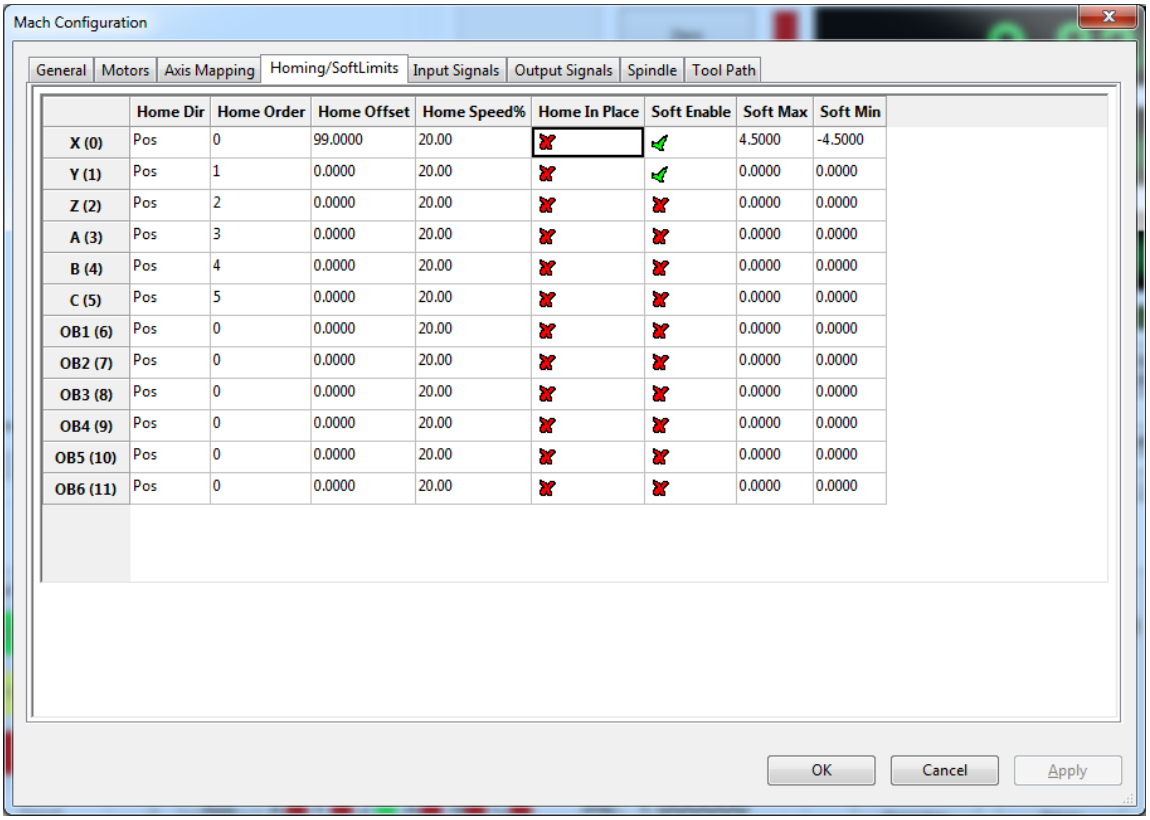
MPG 2

X 🟥 Y 🟥 Z 🟩 A 🟥 B 🟥 C 🟥 Inc: 1.000000

Axis Homing Setup and Soft Limits

Refer to the Homing Parameters in the System Tab of the DSPMC Plugin Config.

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



	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

Refer to the Mach4 manual for information regarding the fields in this window. General Mach4 knowledge applies to this setup.

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

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).*