DSPMC/IP

Ethernet Motion Controller
Data Acquisition System
PID Controller

User Guide

Version 2.18
(Updated January 19, 2010)

Phoenix, AZ USA

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

1. Overview

1.1 Introduction

The DSPMC/IP is an Ethernet based controller for motion control, data acquisition, and general PID control system applications. Utilizing the latest Digital Signal Processing (DSP) technology, the DSPMC/IP offers the highest performance in its class.

DSPMC/IP board is the perfect solution for a variety of applications involving PC based Motion Control, Storage and Retrieval Systems and Milling / Lathe CNC Machines. Equipped with a rich set of hardware interfaces, it can also be used for wide variety of applications involving PID control, e.g., speed, oven temperature control and so on.

DSPMC/IP comes with 48 digital I/Os, 8 channel analog inputs of 14-bit resolution, 6 encoder inputs of 32-bit resolution, and 8 channel analog output of 14-bit resolution with +/-10V range. The board has a wide input power supply range, from 10V to 40V DC. It has a RS-232 serial port and TCP/IP based 10/100 Ethernet connectivity.

DSPMC/IP board comes with GUI software tools to test the hardware, setup PID controller, run motion control commands, and upgrade new firmware. Following gives a brief description of the software tool set:

- DSPMC AxisWorks – A GUI based software tool to exercise the user's hardware installation. After configuring the I/Os and PID settings, this program can be used to execute motion related commands to verify the installation is setup properly

- DSPMC Firmware Upgrade – A GUI based software tool to update the program stored on the dspMC board. New versions of this program can be obtained from the factory.

- Windows driver DLLs

Extremely Important Reminder

When operating machines, take extreme precautions. The machines can have enormous power even with a small motor. Never come inside a machine path while powered. Operating machines without necessary precautions can result in lost of limbs or even death.

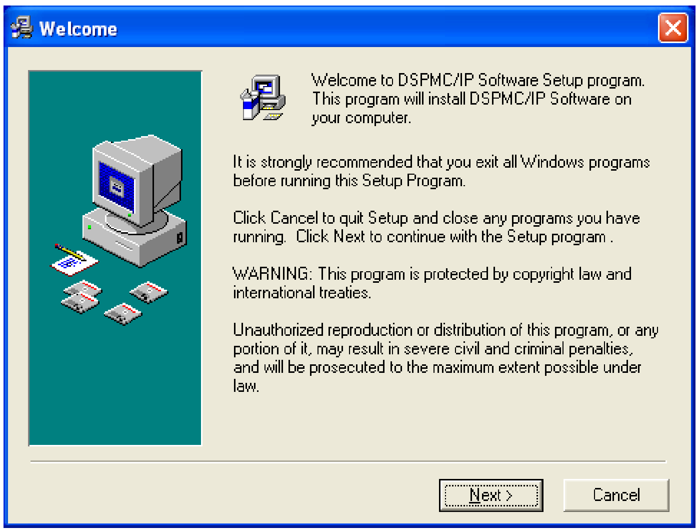
2. Software setup

2.1 Installing the software tools

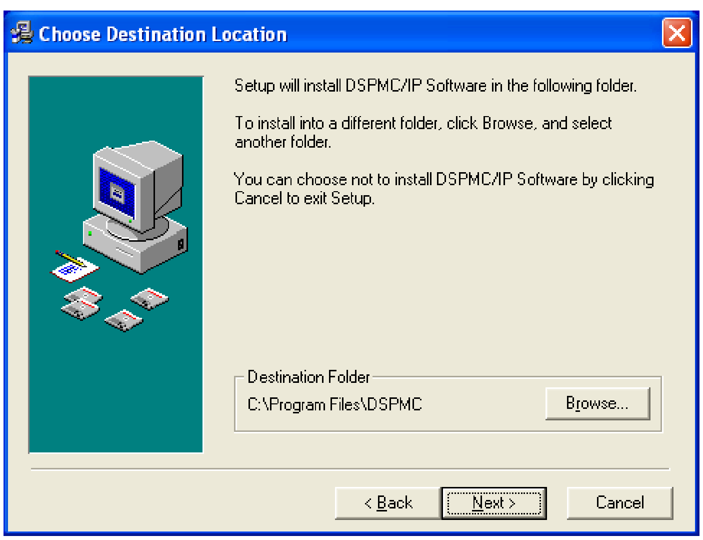
Following are the steps to download and install the setup program.

Download the setup software from http://www.vitalsystem.com/dspmc, click on the link Setup Software. Save the dspMCsetup.zip file to an appropriate directory. Unzip this file and double click on the setup.exe file.

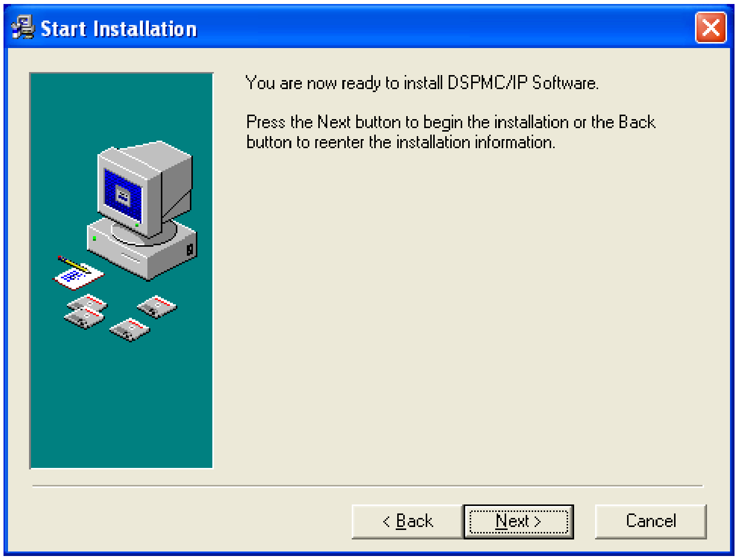
The following Welcome screen appears. Click on next button.

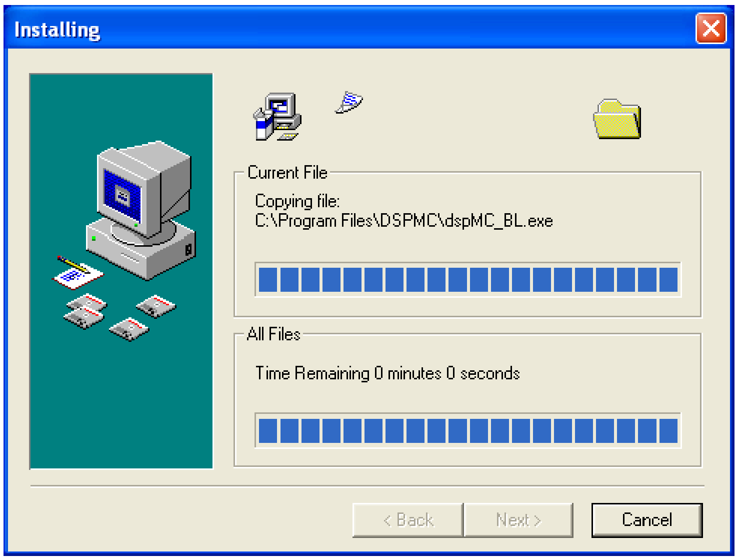


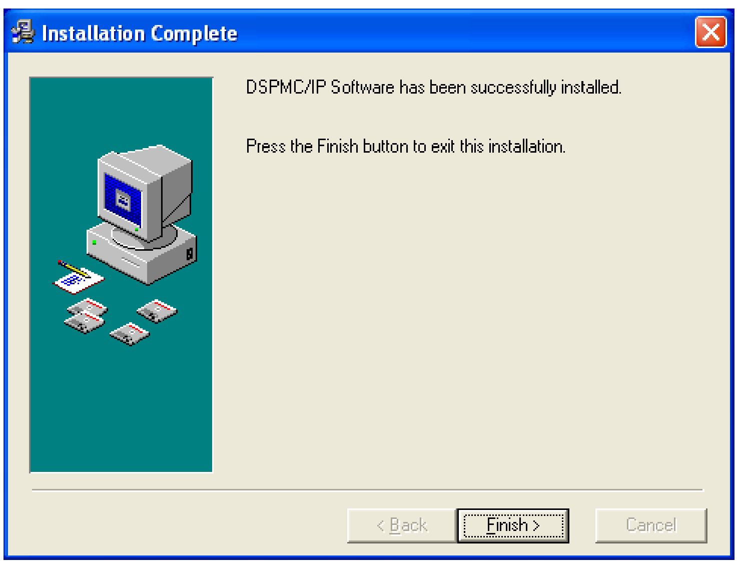
Select the destination of the programs to be installed.



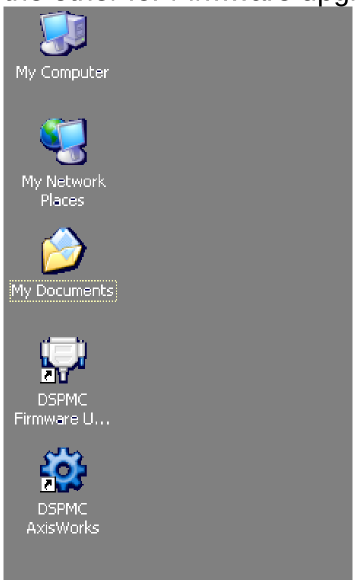
Click on the Next Button to start the installation.


Start Installation
You are now ready to install DSPMC/IP Software.
Press the Next button to begin the installation or the Back button to reenter the installation information.
< Back
Next >
Cancel


Installing
Current File
Copying file:
C:\Program Files\DSPMC\dspMC_BL.exe
All Files
Time Remaining 0 minutes 0 seconds
< Back
Next >
Cancel


Installation Complete
DSPMC/IP Software has been successfully installed.
Press the Finish button to exit this installation.
< Back
Finish >
Cancel

The installation is complete if you see the above window. Click on ‘Finish’ to exit the setup program. This installation creates two shortcut icons on the desktop, one for the AxisWorks testing program and the other for Firmware upgrade software.



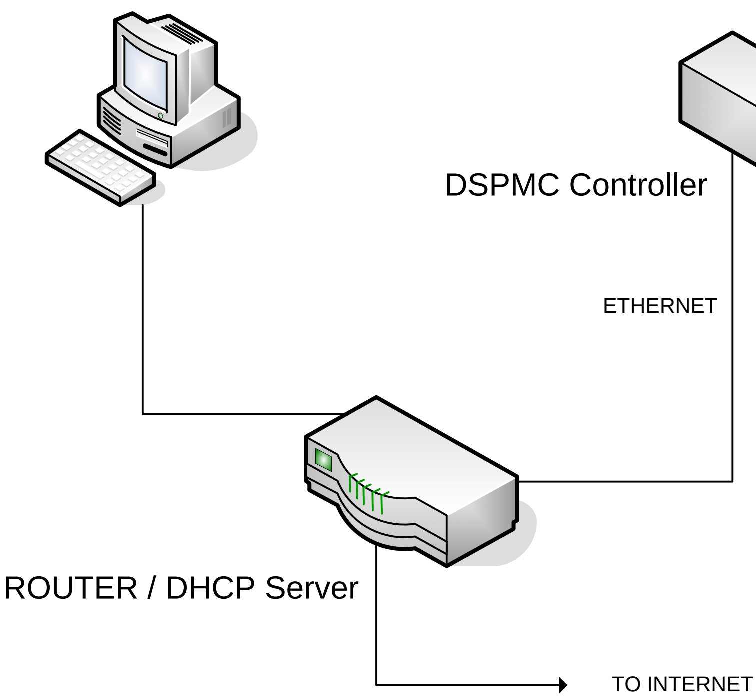
3. Network Connection Setup

You can connect the dspMC board directly to your PC or connect via an Ethernet switch or router. The dspMC board can use the firmware pre-assigned IP address, ie, 192.168.0.50, or it can get a unique IP address from an external DHCP server on your network. In the latter case, the firmware pre-assigned IP address is ignored.

There are two ways to setup the IP addresses of your PC and the dspMC board.

1. Using a Router with DHCP Server
2. Manually assigning an IP Address to your PC

3.1 Setup IP address using a Router with DHCP Server

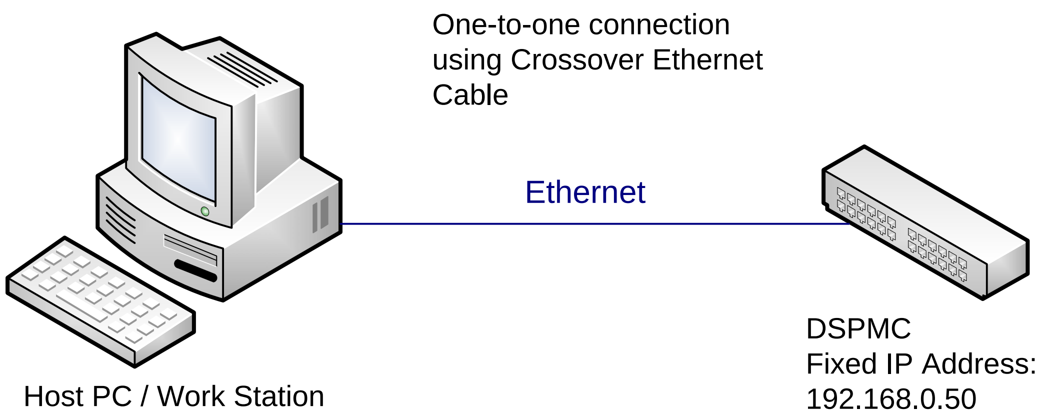


The figure above shows a basic setup using a router on your network. Connect the Ethernet cable from the J1- Ethernet port of the dspMC board to the DHCP server/Router. Connect another Ethernet cable from the DHCP Server/Router to the PC. The DHCP server dynamically assigns IP address both to the PC as well as to the dspMC board, and therefore completes the network setup without requiring any intervention from the user.

3.2 Manually assigning an IP Address to the PC

When connecting the PC directly to the DSPMC board, you will need to manually assign an IP address to your PC. The DSPMC board will use its firmware pre-assigned IP address, i.e., 192.168.0.50, so there is no need to reassign IP address to the board.

The Ethernet cable is connected from the J1-ethernet port of the DSPMC board to the PC as shown below:

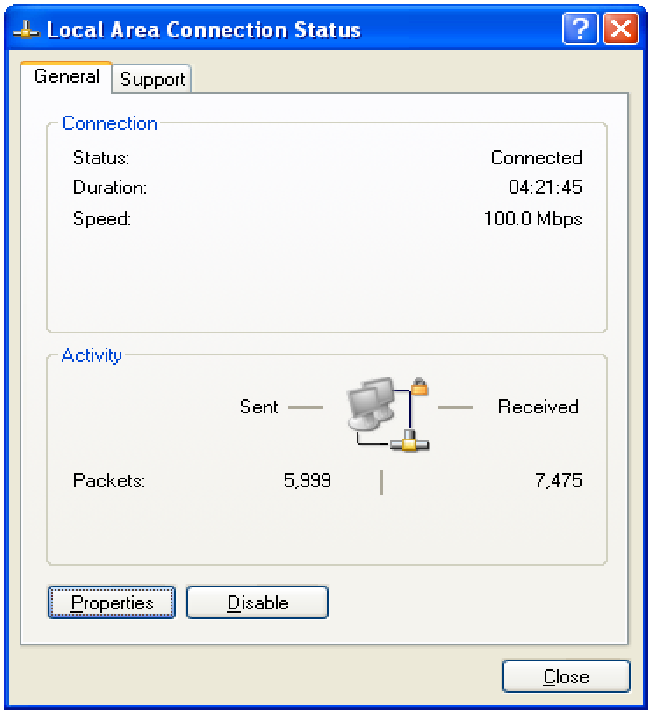


Note: If your network card has Auto-MDIX feature, a crossover cable is not be required.
This setup works best when you are using the DSPMC to control a machine tool. It requires less hardware and wiring.

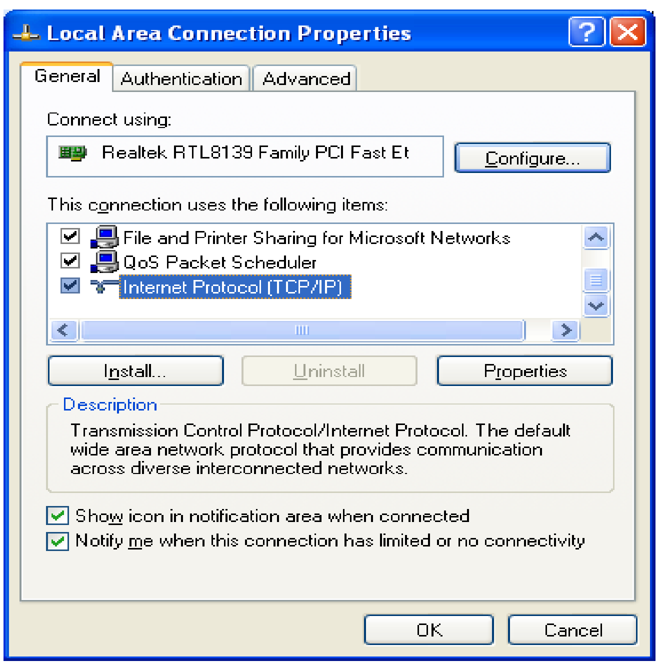
The PC IP Address can be configured manually in windows XP as follows. For other operating systems, please consult the respective user guides for changing the IP address.

1. Double click on the ‘My Network Places’ icon in Windows XP and open the ‘available network connections’.

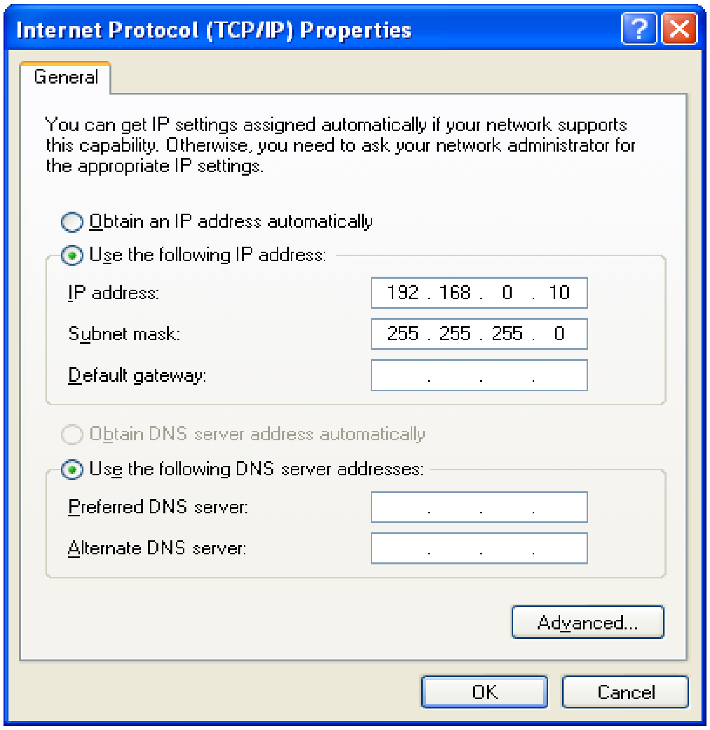
2. Double click on the corresponding LAN Connection over which the device will be setup. The following window appears.



3. Click on the Properties and select the Internet Protocol (TCP/IP) Connection in ‘General’ Tab

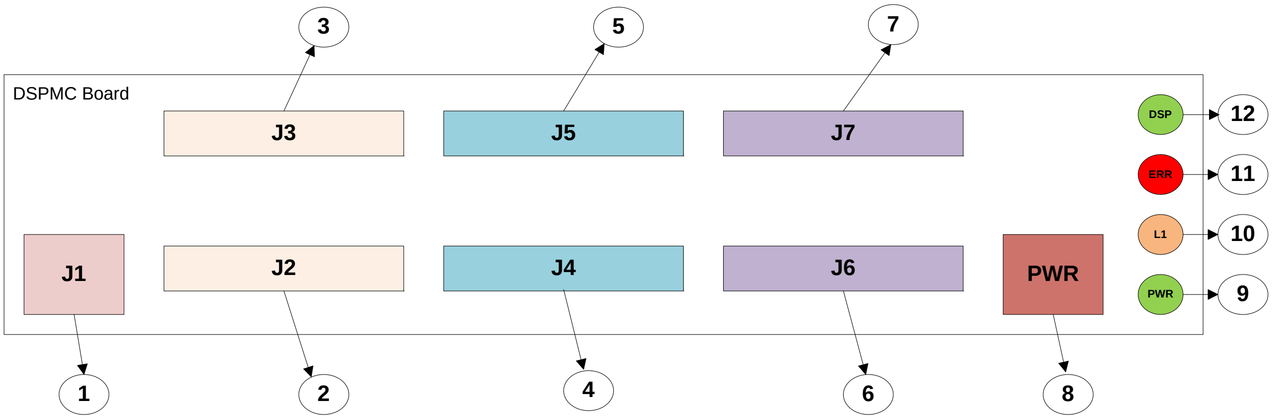


4. Click on the ‘Properties’ button and make the settings in your PC similar to the one shown in the figure below. After settings are done, click ‘OK’ button to finish the setup


Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address:
192 . 168 . 0 . 10
Subnet mask:
255 . 255 . 255 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Advanced...
OK
Cancel

4. Hardware Interface description

The DSPMC board has several interface ports and indicator LEDs. Figure below shows a side horizontal view of the dspMC board with interface ports and other components



1 – J1 – Ethernet port connected to PC
2 – J2 – Provides analog input and DAC output
3 – J3 – Port for RS 232, DeviceNet and Stepper Motor outputs
4 – J4 – Digital I/Os, provide 16 inputs (from 0 to 15) and 8 outputs (from 0-7)
5 – J5 – Digital I/Os, provide 16 inputs (from 16 to 31) and 8 outputs (from 8-15)
6 – J6 – Encoder inputs (for Encoders 0, 1 and 2).
7 – J7 – Encoder inputs (for Encoders 3, 4 and 5)
8 – Power Connector
9 – PWR LED – Green colored LED for Power indication; it glows steadily when Power is on
10 – L1 LED – Orange colored LED for PIDs in-control; it glows steadily when PID is armed
11 – ERR LED – Red colored LED for error indication
12 – DSP LED– Green colored LED indicating DSP processor operation; blinks constantly during normal operation.

4.1 J1 - Ethernet

Connect to PC directly or via an Ethernet Hub or a switch. The dspMC board supports 10 MBit and 100 Mbit network speeds. TCP/IP network protocol in UDP mode is used for PC communications.

4.2 J2 - Analog I/O Port

- **Analog Inputs**
  - Input voltage Range: **0 to 5 Volts**.
  - Input impedance: 10M Ohm.
  - Binary Resolution: 14 bits
  - Conversion Rate: up to 20KHz

- **Analog Outputs**
  - Analog Output range: +/-10 Volts.
  - Analog Output Resolution: 14 Bits
  - Maximum Output Current Per Output: 20mA

J2 Pin Assignments:

Pin#	Function	Pin#	Function
1	+12V, 100mA max	20	Analog Input 0
14	+5V, 500mA max	8	Analog Input 1
2	-12V, 50mA max	21	Analog Input 2
15	Analog Output 0	9	Analog Input 3
3	Analog Output 1	22	Ground (return)
16	Analog Output 2	10	Analog Input 4
4	Analog Output 3	23	Analog Input 5
17	Ground (return)	11	Analog Input 6
5	Analog Output 4	24	Analog Input 7
18	Analog Output 5	12	Ground (return)
6	Analog Output 6	25	Ground (return)
19	Analog Output 7	13	RESERVED
7	Ground (return)		

4.3 J3 –Serial RS232 / CAN

J3 Pin Assignments:

Pin#	Function	Pin#	Function
1	Reserved	20	Reserved
14	Reserved	8	Reserved
2	Reserved	21	Ground (Return)
15	Reserved	9	+5V, 500mA max
3	Reserved	22	RS 232 Rx
16	Reserved	10	RS 232 Tx
4	Reserved	23	Ground (Return)
17	Reserved	11	CAN Shield
5	Reserved	24	CAN V- (0v)
18	Reserved	12	CAN V+ (24v)
6	Reserved	25	CAN Lo
19	Reserved	13	CAN Hi
7	Reserved		

4.4 J4, J5 - Digital I/O Ports

On the DSPMC board, there are two Digital I/O connectors, providing total of 32 digital inputs and 16 digital outputs. Each DB-25 connector provides sixteen inputs and eight outputs.

These I/Os are not optically isolated. To get optical isolation, you can use the digital I/O breakout board ***pn 7535*** with dspMC board. Please visit ***www.vitalsystem.com*** for more info on 7535. These boards connect directly to J4 and J5 and provide detachable screw terminals for easy wiring and maintenance.

The digital inputs are also used for emulated quadrature encoder for low speed applications, eg, MPG wheel for CNC control. For more details, see section 4.6 Single-Ended Encoder Inputs.

The DSPMC board has the following electrical limits on its Digital I/O pins:

Digital Inputs	Digital Outputs
Input Voltage Range: 0 … 5.5 Volts High Level Threshold: 3.0 Volts Low Level Threshold: 0.8 Volts	Output Voltage Range: 0 … 5 Volts Max Output Current Per Output: 8mA

J4 Pin Assignments:

Pin#	Function	Pin#	Function
1	Ground (Return)	20	Input 4
14	Output 0	8	Input 5
2	Output 1	21	Input 6
15	Output 2	9	Input 7
3	Output 3	22	Input 8
16	Output 4	10	Input 9
4	Output 5	23	Input 10
17	Output 6	11	Input 11
5	Output 7	24	Input 12
18	Input 0	12	Input 13
6	Input 1	25	Input 14
19	Input 2	13	Input 15
7	Input 3		

J5 Pin Assignments:

Pin #	Function	Pin#	Function
1	Ground (Return)	20	Input 20
14	Output 8	8	Input 21
2	Output 9	21	Input 22
15	Output 10	9	Input 23
3	Output 11	22	Input 24
16	Output 12	10	Input 25
4	Output 13	23	Input 26
17	Output 14	11	Input 27
5	Output 15	24	Input 28
18	Input 16	12	Input 29
6	Input 17	25	Input 30
19	Input 18	13	Input 31
7	Input 19		

4.5 J6, J7 – Differential Quadrature Encoders

J6 Pin Assignments:

Pin#	Function	Pin#	Function
1	Ch 0 A+	8	+5V 500mA
14	Ch 0 A-	21	Ground (Return)
2	Ch 0 B+	9	Ch 2 A+
15	Ch 0 B-	22	Ch 2 A-
3	Ch 0 Z+	10	Ch 2 B+
16	Ch 0 Z-	23	Ch 2 B-
4	+5V 500mA	11	Ch 2 Z+
17	Ground (Return)	24	Ch 2 Z-
5	Ch 1 A+	12	+5V 500mA
18	Ch 1 A-	25	Ground (Return)
6	Ch 1 B+	13	Reserved
19	Ch 1 B-		
7	Ch 1 Z+		
20	Ch 1 Z-		

J7 Pin Assignments:

Pin#	Function	Pin#	Function
1	Ch 3 A+	8	+5V 500mA
14	Ch 3 A-	21	Ground (Return)
2	Ch 3 B+	9	Ch 5 A+
15	Ch 3 B-	22	Ch 5 A-
3	Ch 3 Z+	10	Ch 5 B+
16	Ch 3 Z-	23	Ch 5 B-
4	+5V 500mA	11	Ch 5 Z+
17	Ground (Return)	24	Ch 5 Z-
5	Ch 4 A+	12	+5V 500mA
18	Ch 4 A-	25	Ground (Return)
6	Ch 4 B+	13	Reserved
19	Ch 4 B-		
7	Ch 4 Z+		
20	Ch 4 Z-		

4.6 Single-Ended Encoder Inputs

In addition to dedicated hardware encoder inputs, DSPMC board also provide three Single Ended encoder inputs in Digital Input J4 and J5 connector. These are simulated encoder inputs, therefore called, *SoftEncoder*, and are used for low speed applications like MPG. The following table lists the Digital I/O pins assigned to *SoftEncoders*. (Requires Firmware Rev 63 or newer):

SoftEncoder 0 : A+ On J4 Pin 24,
B+ On J4 Pin 12

SoftEncoder 1 : A+ On J4 Pin 25
B+ On J4 Pin 13

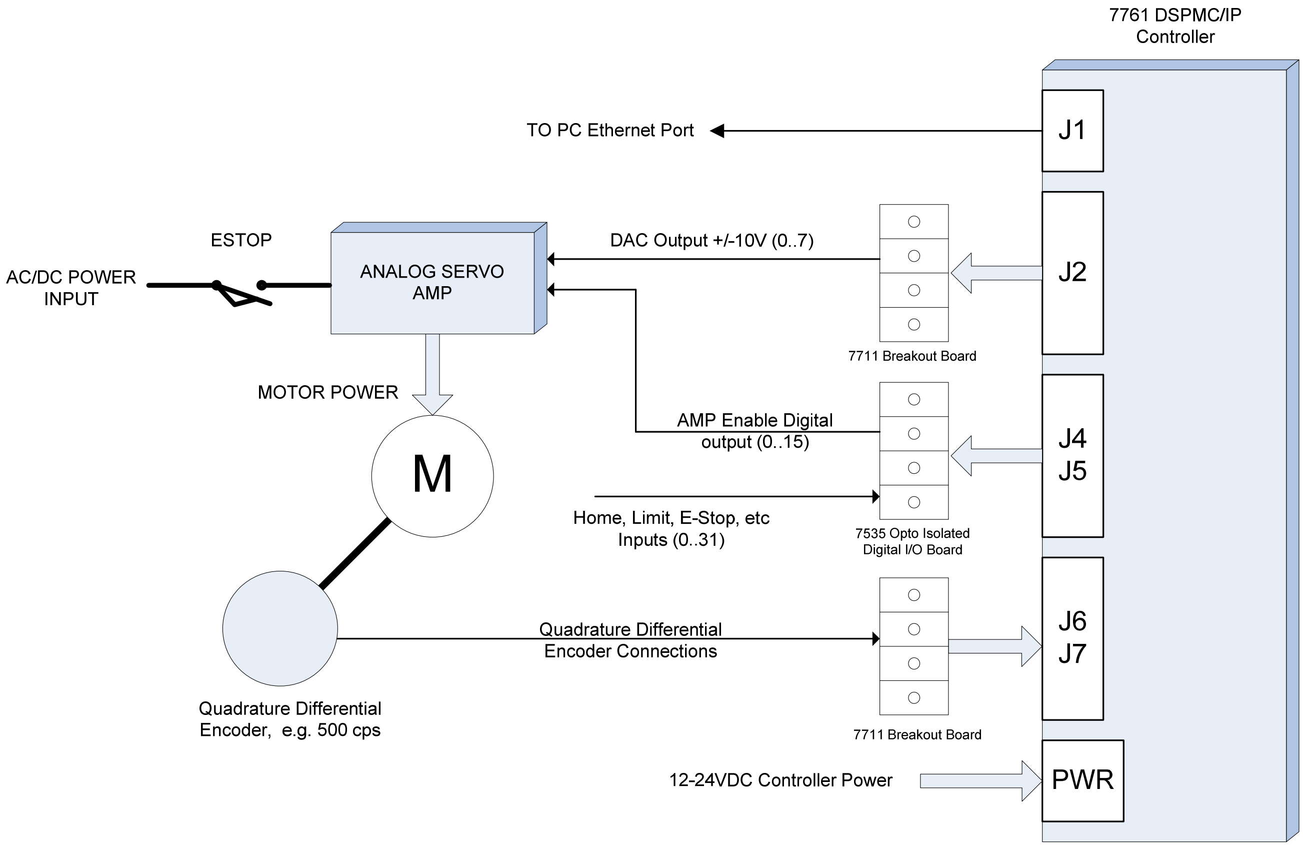
SoftEncoder 2 : A+ On J5 Pin 18
B+ On J5 Pin 6

SoftEncoders are normally used as MPG source. See Section A.16 MPG Settings – MpgSource1 for more information.

In addition, *SoftEncoders* are designed to avoid Jerks when MPG scale is changed by the PC software. To accomplish this, these encoder counters are implemented using floating point numbers (instead of whole integer numbers). They can be assigned decimal values (eg 3.092, 5.001, 64000.5 etc) which the hardware encoder counters are not capable of (they can only take whole integers).

5. Hardware Connections

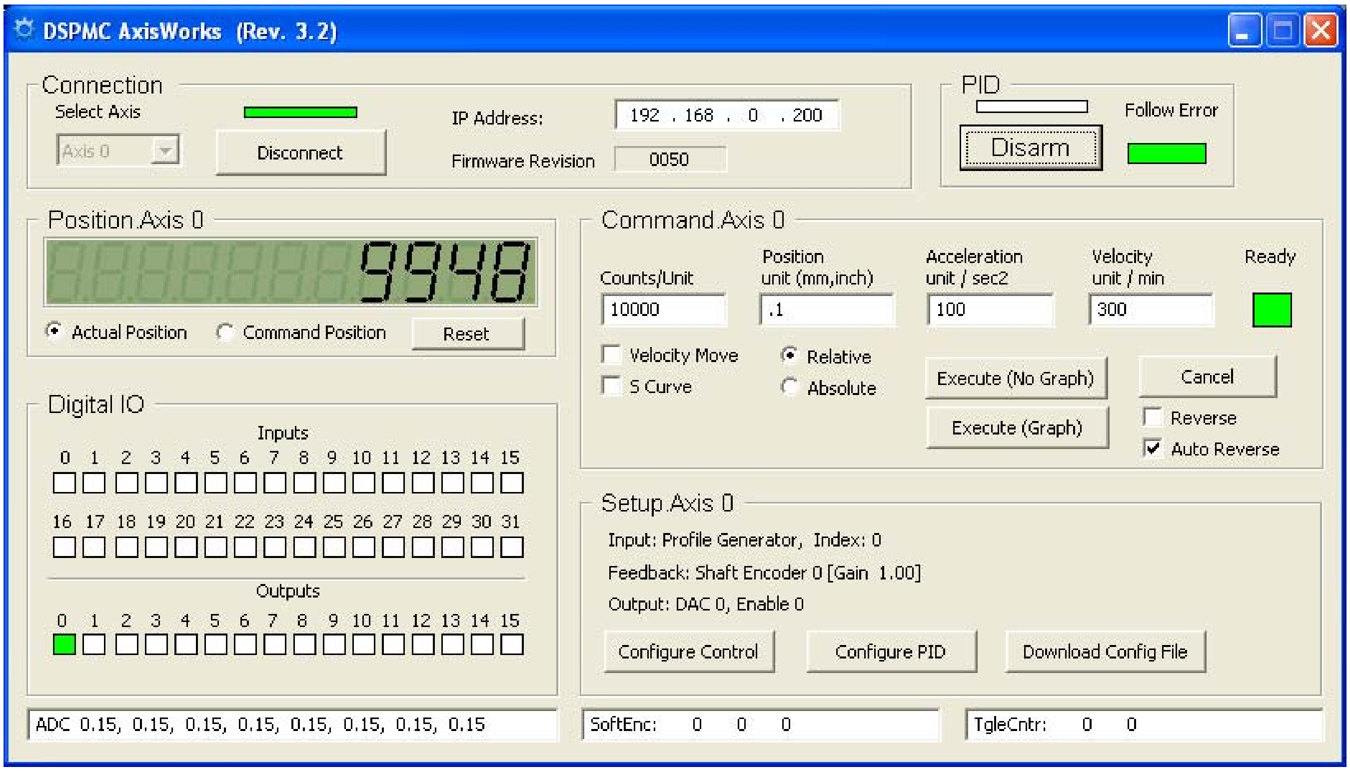
The figure below shows a typical axis setup using Analog Servo amplifier and quadrature encoder feedback. The connectors on dspMC board are all DB25 connectors. The user can wire up the system using the 7711 (or 7721) and 7535 breakout boards as shown in the figure.



6. AxisWorks Software tool

The AxisWorks software tool allows the user to test the dspMC installation. The user can exercise motion and I/O controls easily, without doing any custom programming. The tool lets the user configure the I/O and PID settings. I/O control configuration includes configuring the servo loop Input source and gain, Feedback source, output control and amp-enable output control etc. Once configured, the user can run the motion commands and can also tune the PID controls. PID configuration includes configuring each of the P, I, and D gains as well as set the error limits if any.

Upon double-clicking the DSPMC AxisWorks icon on the desktop, the initial window would look similar to the following. Each screen item is explained in detail as follows



6.1 User Interface

1 – Connection – Provides details for connecting AxisWorks to the dspMC board

Before clicking on the 'Connect dspMC' button, check whether the board has been powered on and network setup has been done as explained in *Network Connection Setup*.

Note that the IP Address and Firmware revision fields (which specify the dspMC board's IP address and current loaded firmware revision) are left blank before the connection to the board is made.

A 'Red' bar above the 'Connect dspMC' button indicates that the connection is not yet made. In order to initiate the connection, select an Axis (out of the 8 Axes from the drop down list). Then click on 'Connect dspMC' button, this performs the actual connection to the board. , If connection is successful, the 'Red' bar will turn into 'Green' color.

2 – PID – Provides control for Arming (enable) and Disarming (disable) of PID loops globally. Also a color bar in red shows if a Following Error is tripped. If green, no error is detected.

3 – Position Axis – Displays actual position and command position for the selected Axis in raw encoder counts.

By selecting 'Command position', the display shows the value of the internal variable for the commanded position for the selected axis.

By selecting 'Actual position', the display shows the current value of the encoder counter for the selected axis.

Note that the actual position usually slightly deviates from the Command position when PID is enabled.

4 – Command Axis - Executes Move command based on User inputs for selected Axis.

Counts/Unit – Number of encoder counts in one Position Unit (e.g. mm, inches, etc). For example, if your preferred unit of measurement is inches, and if 40000 encoder counts move the axis 1 inch, you will put 40000 in this field. If you are going to work directly in encoder counts, then set this value to 1.

For CNC applications, this is calculated by encoder counts as follows:
Encoder Counts / Inch = Quadrature counts/rev x Gear ratio x lead-screw revs/inch.

For example
Encoder: 2000 Counts / Rev (Actual 500 CPR)
Gear Ratios: 2
Lead Screw: 5 Revs / Inch

Encoder Counts / Inch = 2000 x 2 x 5 = 20000

Position – Final position or displacement in terms of Position Units, e.g. 1.5, 10.093, etc.

Acceleration – Acceleration value in terms of Position Units per second squared, e.g. inches/second^2, mm/sec^2 etc.

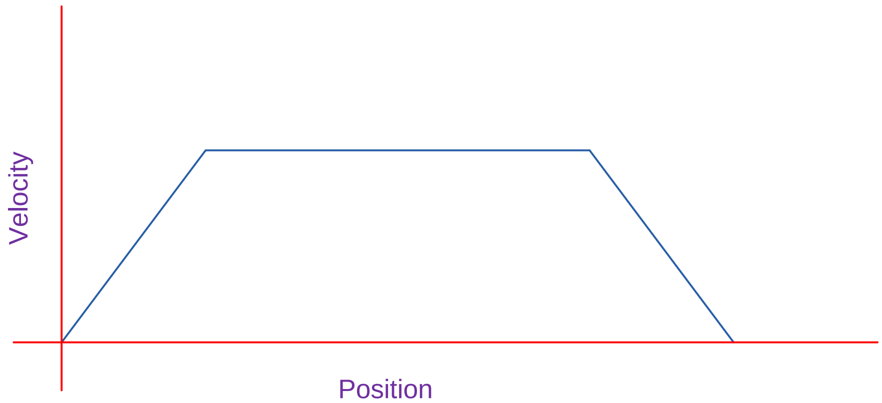
Velocity – Velocity value in terms of Position Units per minute, e.g. inches/minute, mm/mintue etc.

Note:
Be careful not to use Acceleration and Velocity values greater than the machine is capable of, or you will cause a "Following Error". Start slow and work your way up. When you get an error that you can't tune out, back your settings down to 10% or so.

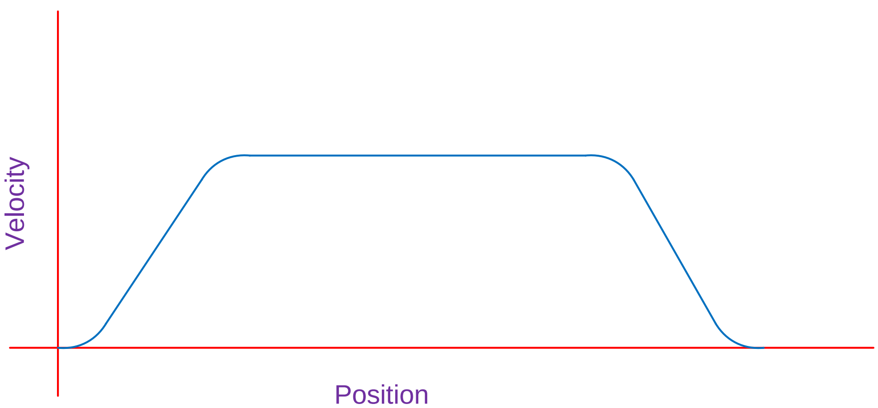
Relative and *Absolute* – These buttons indicate whether the value in the Position field is either the total distance to travel (relative) or the final position (absolute).

Velocity Move – This indicates that the move will not use the position data provided. This is useful when you need to run the axis for long time without worrying about the final position. To stop the axis, click on the cancel button.

S-curve – This indicates that the motion profile will transition smoothly (no sudden change in acceleration). The motion profiles with or without S-Curve option are shown below:

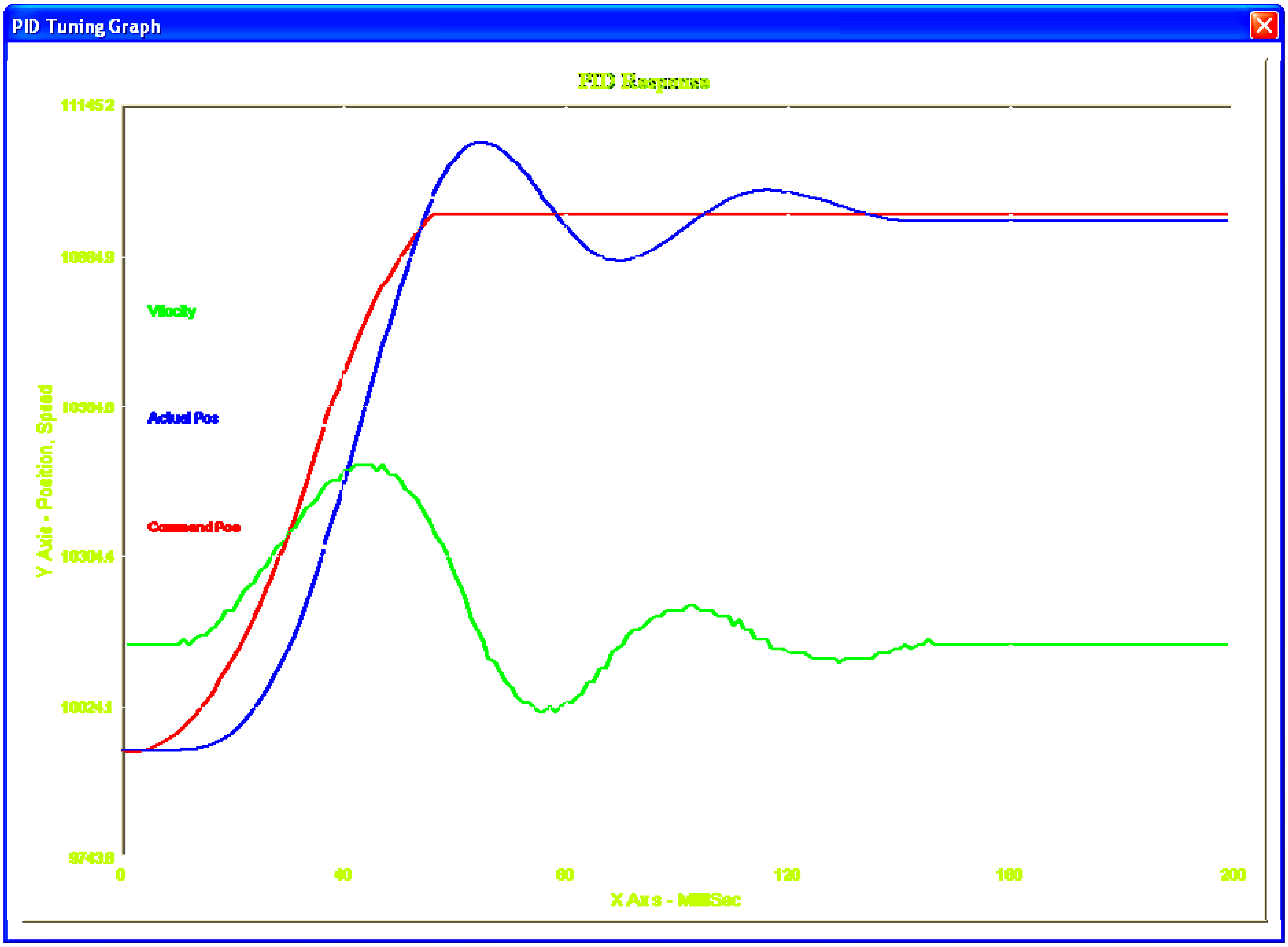


Trapezoidal Motion Profile



S-Curve Motion Profile

Pressing the 'Execute' button (with or without graph) starts the motion. User can press 'Cancel' button to cancel the motion execution anytime during the machine operation. Make sure you have setup the axis properly before moving the axis, as described in the following sections. If Execute with Graph is selected, the Graph window will appear, similar to shown below. The red line shows the commanded position, Blue line shows the actual position, while the Green line shows the actual velocity profile.



5 – Digital I/O – Displays the current states of digital Input and Output pins.

6 – Download Config File – Downloads the user specified configuration file (e.g. dspMC.xml) to the board. [Appendix A](#) shows the format of this xml file. You can manually edit this file and re-download the configuration without going thru different windows.

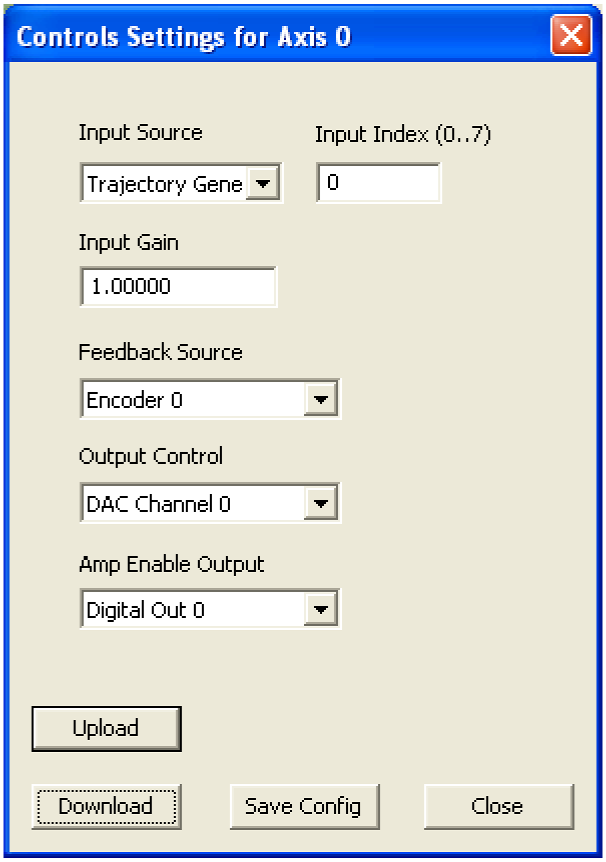
7 – Setup Axis – Setup axis controls (input, feedback, output etc) and PID configuration for the selected axis.

8 – Motion Profile Graphs – Displays actual versus programmed axis motion

The Current configuration values for 'Input', 'Feedback' and 'Output' are displayed for the selected axis. 'Configure Control' and 'Configure PID' provides dialog windows to the user to add/modify the configuration and PID settings.

6.2 Controls Configuration

The ‘**Configure Control**’ button displays following dialog window:



Input source is the command input to the PID controller for the axis. This can be any one of the following:

- PC Host Software (this configuration is required by some PC software, e.g. Mach3)
- Trajectory Generator
- Any Encoder input
- Any analog input

> ***Important Note:***
> You must select “Trajectory Generator” as the Input Source when using ***AxisWorks*** to execute motion on DSPMC Controller.

Input Index is the index of the command position source. This number normally match the axis number. In case for slave axis, it is equal to the master axis number.

Input Gain is the multiplier before the source data is used, e.g. if encoder 0 is the source, the counter value is multiplied by Gain and then used as the input to the PID controller for the selected axis.

Feedback source to the PID controller can be from any Encoder input or any Analog inputs (A/D channel 0-7)

Output control selects the DAC channels (DAC 0-7) that will receive the PID controller output.

Amp Enable Output selects the output pin used to enable the external servo amplifier. User can select any of the 16 digital output pins (Digital out 0-15) as the enable pin for the selected axis.

Note that an 'undefined' value is displayed for these fields when not configured.

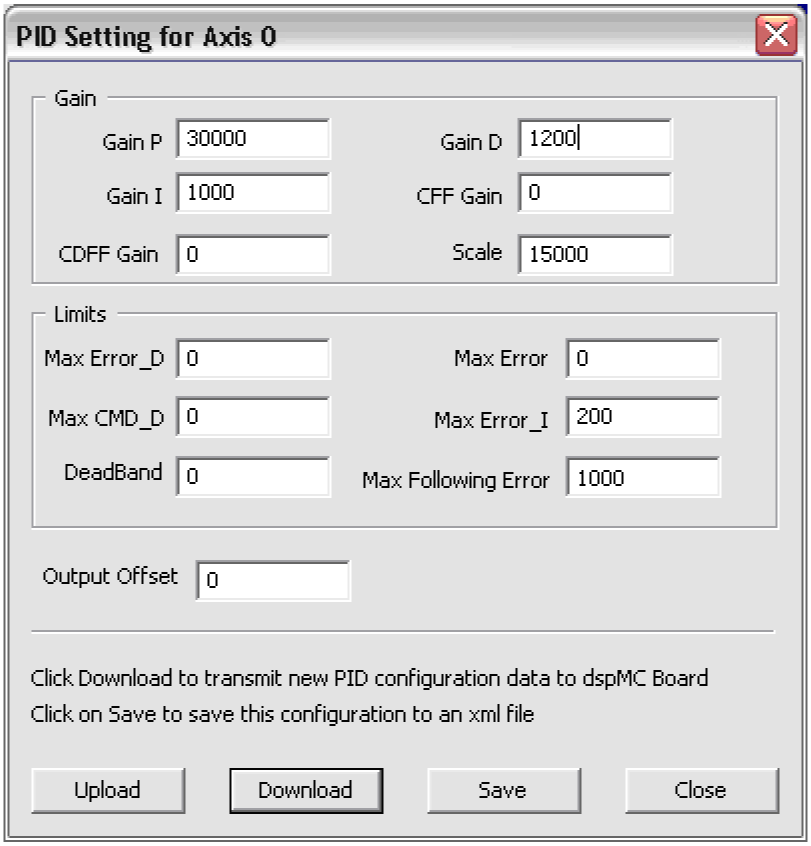
The ***Upload*** button reads the current values from the DSPMC board and displays them on this window. If you make a change and like to see the original value you can click on Upload button to get the current programmed values inside the DSPMC.

Clicking the ***'Download'*** button send the configuration to the DSPMC board. Note that these configurations are stored on volatile memory inside the board, i.e., the information will need to be re-downloaded when power is cycled on the controller. To avoid re-entering the information, you can also save the configuration to the dspMC.xml file by clicking on the '***Save Config***' button. Once saved, the file can be downloaded to the DSPMC by clicking on the '***Download Config File***' button on the main window.

> ***Note for Mach3 Users:***
> dspmc.xml is the file that you will copy and paste into the Mach3 folder when you are done setting up with ***AxisWorks*** *and ready to run with Mach3*.

6.3 PID Filter Configuration

The ‘**PID Settings**’ button displays the following dialog window:



Important Note:

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

User can input the P, I, and D coefficients as well as error limits and scale.

Scale is used as a divider for all the setting in the PID gain box. This makes it possible for the software to use smaller manageable numbers and still get the same effect. An example is if you put a gain of 4000 and a scale of 100, it is the same as a scale of 1 and a gain of 400,000.

Don’t be concerned if on the first time you don’t get any motion as the numbers are trial and error. Using P to D ratio of 25 to 1 works real well. If you have no experience, try these numbers:
P = 4000, I = 0, D = 100000, Max following error 1000, Dead-band = 3, Scale = 100 and all others to 0. If your servo drives are tuned real well, Deadband can be set at zero but start at 3 or 5, higher if your servo’s buzz at a standstill.

Command Feed Forward (CFF) gain and Command Differential Feed Forward (CDFF) gain for PID loop control can also be entered here. These are definitely for the more advanced user but will allow for more precise movement.

User can also specify following limits…
Max_error - Maximum error limit. Leave it at 0.

Max_Error_D - Maximum Derivative Error for Derivate gain,
Max_CMD_D - Maximum Command Error for CDFF gain,
Max_Error_I - Maximum Integral Error for the integral gain,
Deadband – a range of position where the PID is not active
Max Following Error – Maximum deviation allowed between command and actual, above that, the PID controller shuts down and need to be re-enabled manually. If 0, PID will never shutdown which can be extremely dangerous in a run-away motor condition. So always use a positive value in this field.

Output Offset - Sets a constant bias to the PID output. This is useful when tuning Z axis, where motor has to apply more pressure in one direction than the other.

Clicking on '***Download***' sends the PID configuration data to the dspMC board. The new PID configuration can also be saved to an XML file by clicking '***Save***' button. Once saved, there is no need to enter these values every time you run AxisWorks. You can click on the '***Download Config File***' button on the main window to send the configuration to the DSPMC controller.

If you changed any field and like to revert back or see the current PID settings inside the dspMC, click on the '***Upload***' button to read back the data from the controller.

7. Mach3 Software Integration

The Mach3 Software is an off-the-shelf Milling and Lathe machine control software. User can download the trail version of the software from www.machsupport.com

The dspMC board can be integrated with Mach3 to form a high performance machining center. The dspMC Software Tools provide the necessary drivers and configuration files to interface with Mach3 software. If you have installed the software tools as explained in the software installations section, you already have all the necessary drivers.

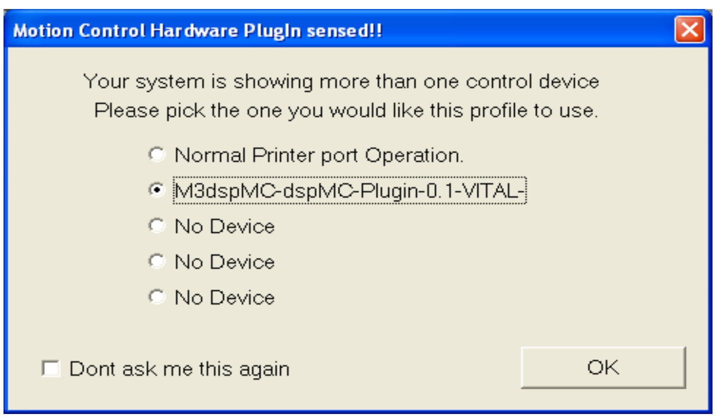
Before using dspMC board with Mach3, the user should test successful software and hardware installation by using dspMC AxisWorks testing tool as explained in *Testing the installation* chapter.

This document assumes that user is familiar with the usage of Mach3 software. This chapter describes the mapping of Mach3 internal software signals to the dspMC connectors.

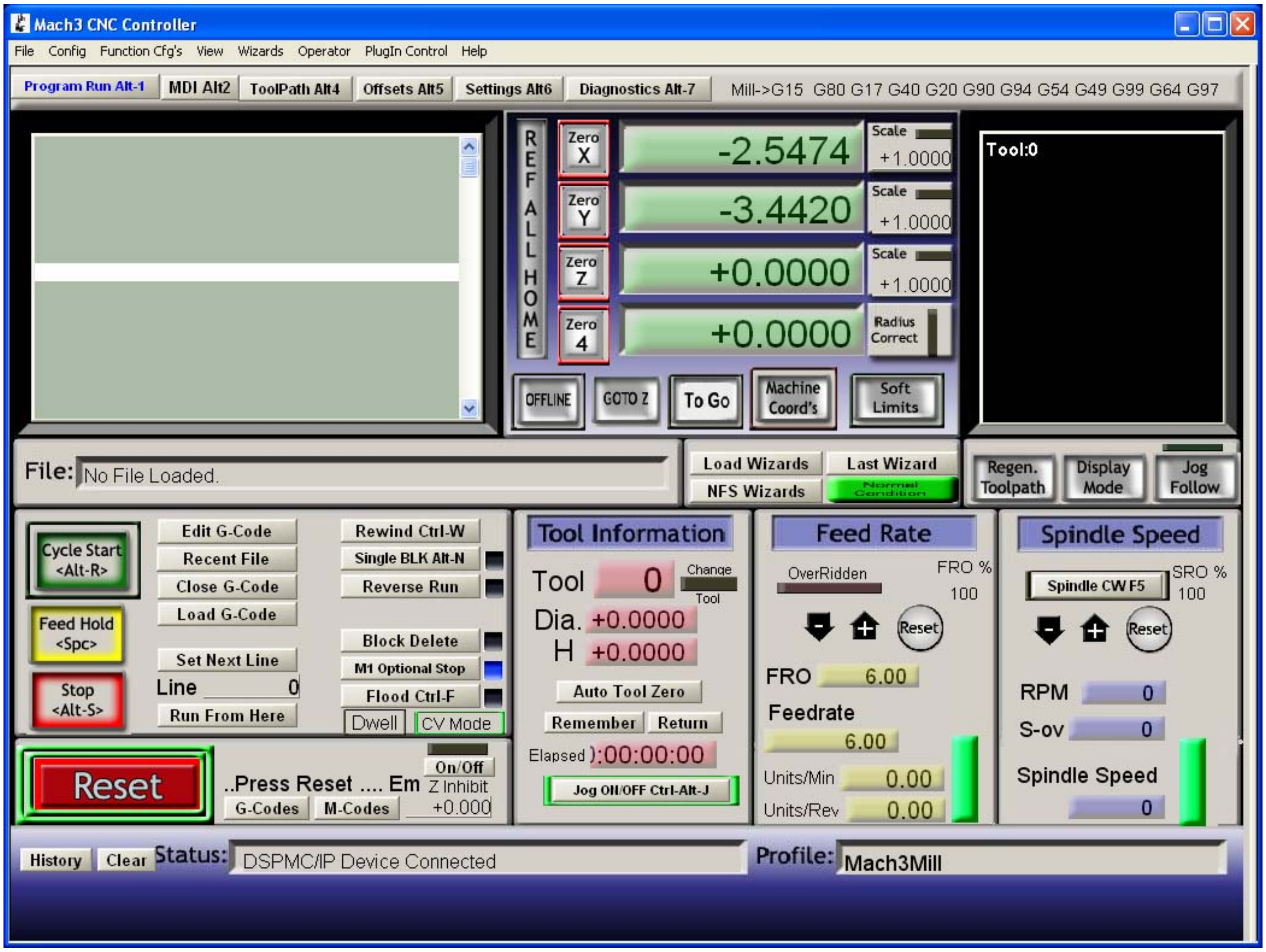
The general Mach3 software operation remains mostly the same when using dspMC plugin.

7.1 Starting Mach3 with dspMC

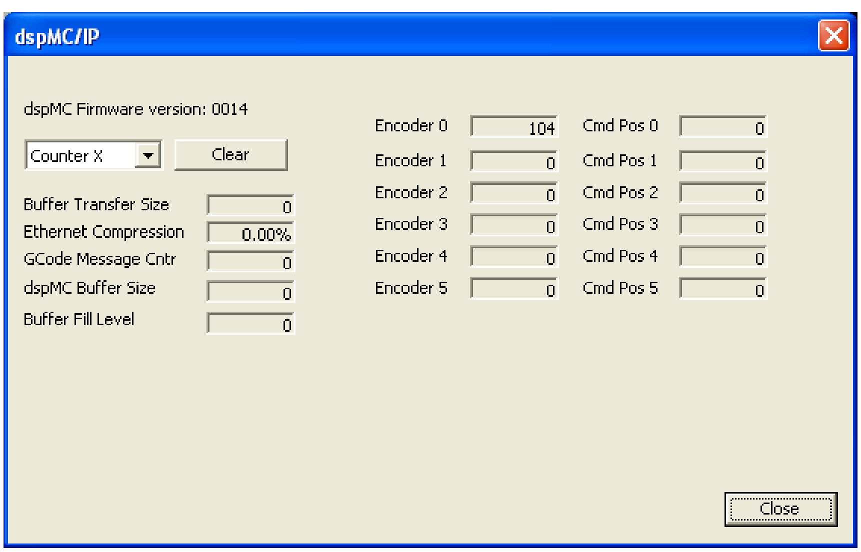
To launch Mach3 with dspMC plugin, double-click on the Mach3Mill software icon on the desktop. It shows the following dialog box with the option to select M3dspMC plugin. Make sure this plugin is selected and click 'OK'.



Make sure the dspMC is powered up and connected to the Ethernet network. The Mach3 software shows up as follows with a message 'dspMC/IP Device Connected' in the Status bar.



Click on the Menu item ‘PlugIn Control’, and then click on the item ‘VITAL dspMC/IP Window” which displays the following screen indicating that the dspMC board is connected with current states of counter and other stats. User can leave this window open while running Mach3.



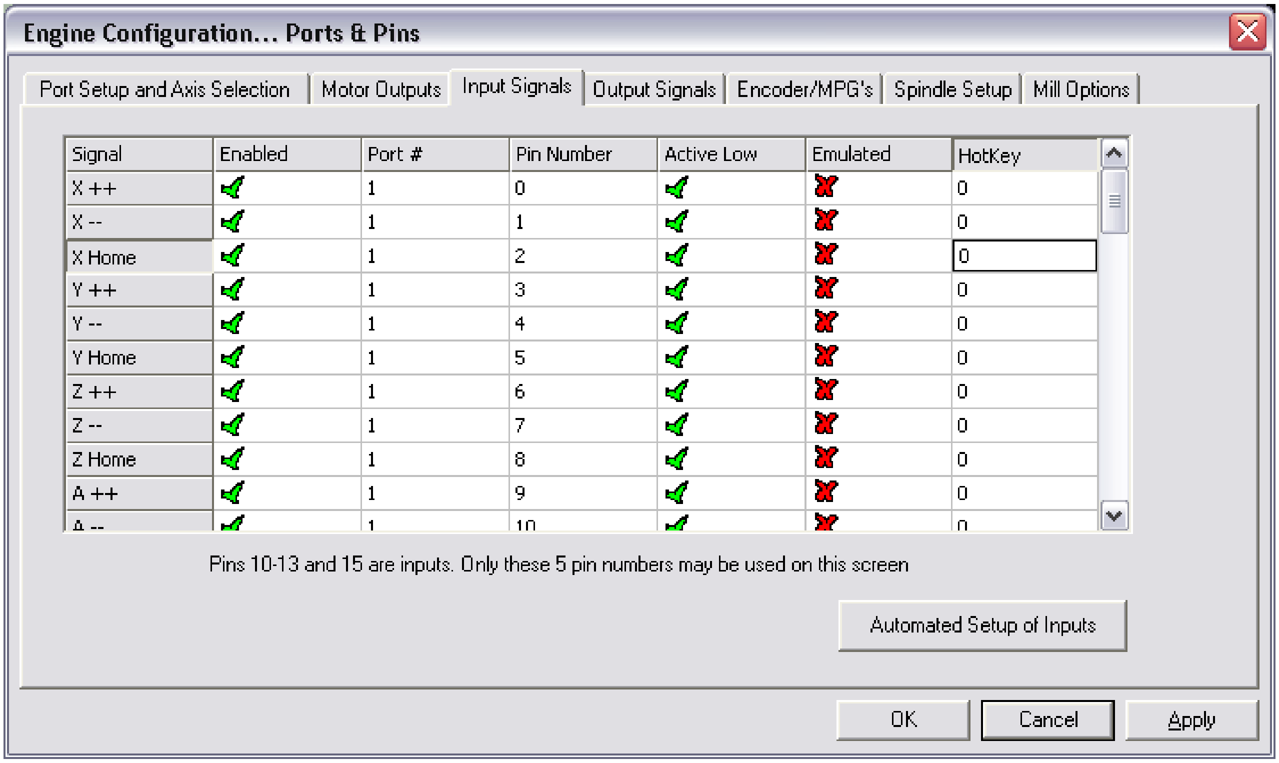
This screen is mostly used for diagnostics, but the user can also clear the Encoder counters, if required.

7.2 Mapping Mach Input Signals to dspMC Digital Inputs

The following table shows the mapping from Mach3 input pin numbers to the actual digital input pin numbers available on the dspMC board.

Mach3 *Input* ports and pins		**dspMC J4 and J5 pin assignments**		**Breakout board 7535 pin assignments**	
Port #	**Pin Number**	**J4 Pin number**	**J5 Pin number**	**Board #1**	**Board #2**
1	0	**18**	X	0	X
1	1	**6**	X	1	X
1	2	**19**	X	2	X
1	3	**7**	X	3	X
1	4	**20**	X	4	X
1	5	**8**	X	5	X
1	6	**21**	X	6	X
1	7	**9**	X	7	X
1	8	**22**	X	8	X
1	9	**10**	X	9	X
1	10	**23**	X	10	X
1	11	**11**	X	11	X
1	12	**24**	X	12	X
1	13	**12**	X	13	X
1	14	**25**	X	14	X
1	15	**13**	X	15	X
1	16	X	**18**	X	0
1	17	X	**6**	X	1
1	18	X	**19**	X	2
1	19	X	**7**	X	3
1	20	X	**20**	X	4
1	21	X	**8**	X	5
1	22	X	**21**	X	6
1	23	X	**9**	X	7
1	24	X	**22**	X	8
1	25	X	**10**	X	9
1	26	X	**23**	X	10
1	27	X	**11**	X	11
1	28	X	**24**	X	12
1	29	X	**12**	X	13
1	30	X	**25**	X	14
1	31	X	**13**	X	15

If using any of the Mach3 input signals, make sure the pins are 'Enabled' and set 'Active Low' as shown in the example figure below.



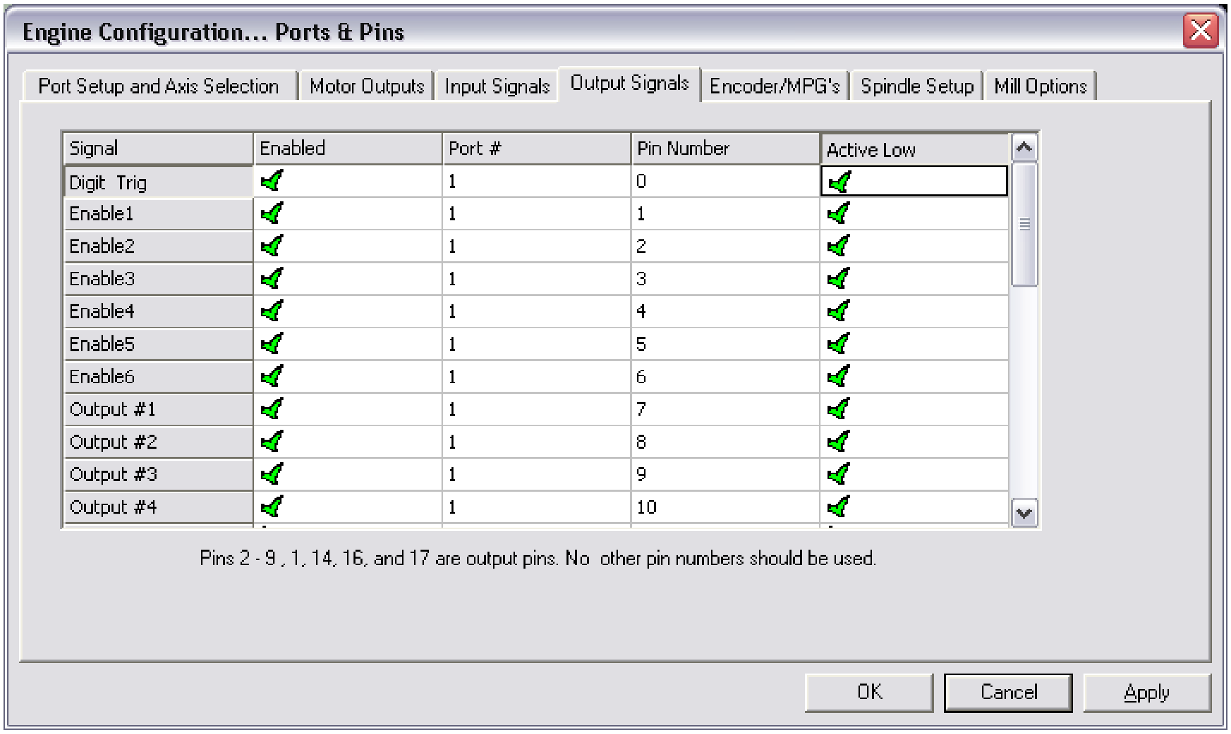
Ignore the line printed on the above window about pins 10-13 and 15 !@#$. This does not apply to DSPMC based system.

7.3 Mapping Mach3 output pins to dspMC Digital Outputs

The following table shows the mapping from Mach3 output pin numbers to the actual digital output pin numbers available on the dspMC board.

Mach3 *Output* ports and pins		**dspMC J4 and J5 pin assignments**		**Breakout board 7535 pin assignments**	
Port #	**Pin Number**	**J4 Pin number**	**J5 Pin number**	**Board #1 On J4**	**Board #2 On J5**
1	0	**14**	X	0	X
1	1	**2**	X	1	X
1	2	**15**	X	2	X
1	3	**3**	X	3	X
1	4	**16**	X	4	X
1	5	**4**	X	5	X
1	6	**17**	X	6	X
1	7	**5**	X	7	X
1	8	X	**14**	X	0
1	9	X	**2**	X	1
1	10	X	**15**	X	2
1	11	X	**3**	X	3
1	12	X	**16**	X	4
1	13	X	**4**	X	5
1	14	X	**17**	X	6
1	15	X	**5**	X	7

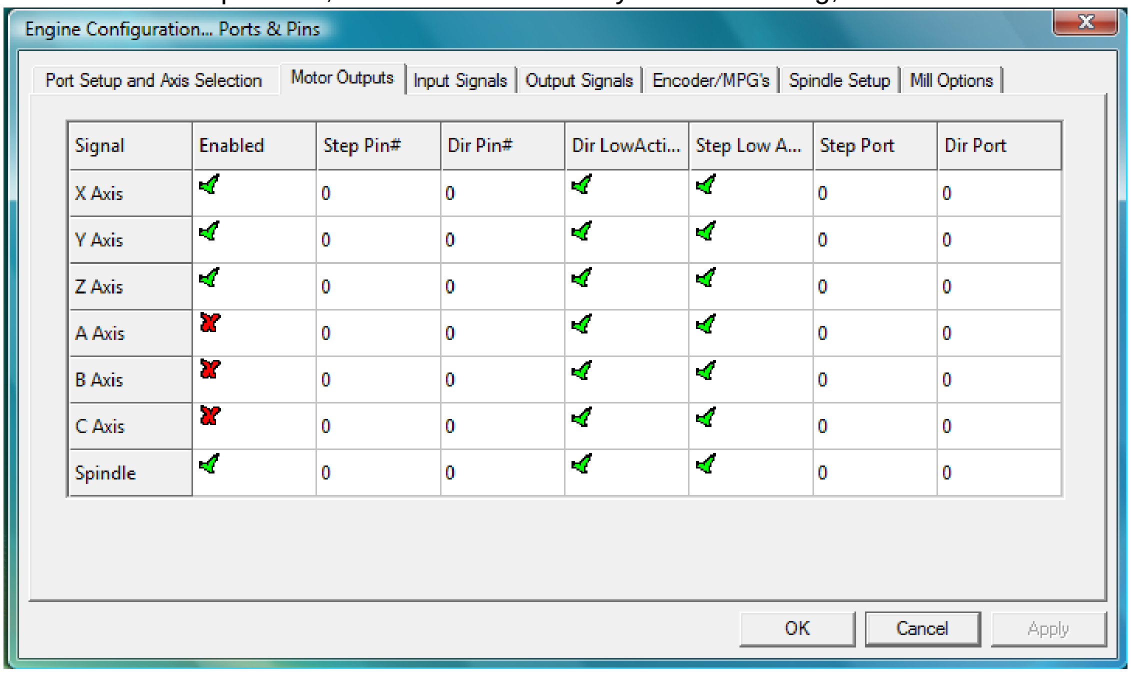
As with the input configuration, if using any of the Mach3 output signals, make sure the pins are 'Enabled' and set 'Active Low' as shown in the example figure below.



Ignore the line printed on the above window about pins 2-9,1,14... !@#$. This does not apply to DSPMC based system.

7.4 Motor outputs.

On the Motor Ouputs tab, enable the ones that you will be using, and set to active low.



7.5 Spindle Setup.

When using a VFD or other motor controlling device that uses 0-10v control the following steps are needed.

Make sure the spindle enable is checked and is set to active low.

Go to the Config tab and then spindle pulleys. Current pulley 1. For this example, set min speed to 0 and max speed to 100. Set Ratio 10. This will give a 0v output to the spindle at S0 (min speed) and a 10v at S100 (max speed).

This setting is great for testing. Without the VFD/Drive hooked up you can test your output with a digital volt meter to make sure you are getting 0-10volts for 0 to max speed.
When it all works then put in min 0 and max gets set to the max speed of your machine, eg, 5000. This will allow you to program S in the G-code in actual rpm, ie 0 … 5000.

7.6 Downloading the PID configuration

The *dspMC.xml* file contains the PID and system configuration data. This file is usually stored in C:\Mach3 folder. The user can modify this file for tuning the PID loop. To download new set of PID configuration onto the dspMC board, click on ‘PlugIn Control’ and select ‘Download Config File’ item.

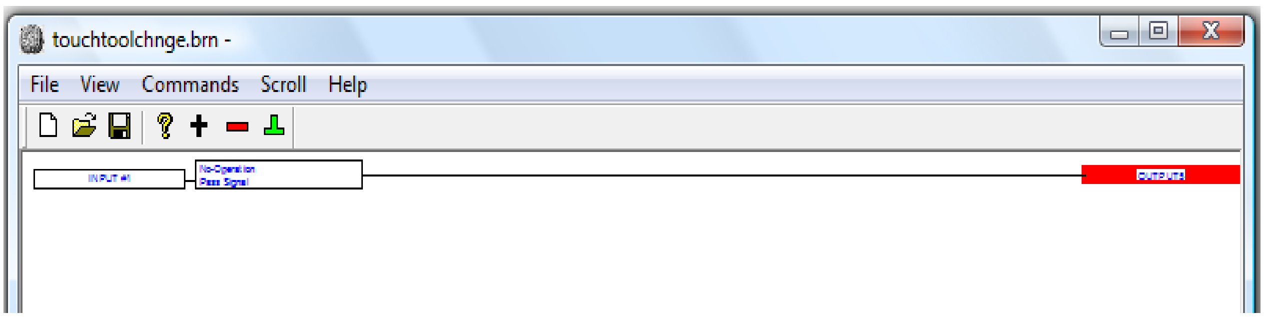
Note that the download is possible only when the Mach3 software is in ESTOP mode ('Emergency Mode Active…' message is displayed continuously when in ESTOP mode).

For more information on the *dspMC.xml* configuration file, please refer to *Appendix A*

7.7 Getting beyond the basic input/output with Mach3.

When your are done with limit switches and other basic I/O you will probably want to have several switches on you control panel next to the e-stop such as feed hold, stop, g-Code rewind and other things. To get this added functionality you will need to learn how to write brains in Mach3. Brains are used to get access to all of the extra I/O and to work tool changers and just about anything you can think of.

Here is a pretty basic brain to map input 1 to output 5.



You would setup what wire goes to input 1 and output 5 under the ports and pins tab of Mach3. After a few tries you will get the hang of it.

For more information: go to www.machsupport.com and then to the video section and look for Brains.

7.8 Axis Homing and Direction

In the Config menu, select Homing/Limits. You will see the following window.

Axis	Reversed	Soft Max	Soft Min	Slow Zone	Home Off.	Home Neg	Auto Zero	Speed %
X	✓	100.00	-100.00	1.00	1	✗	✓	20
Y	✗	100.00	-100.00	1.00	1	✓	✓	20
Z	✓	100.00	-100.00	1.00	.5	✗	✓	20
A	✗	100.00	-100.00	1.00	0.0000	✗	✓	20
B	✗	100.00	-100.00	1.00	0.0000	✗	✓	20
C	✗	100.00	-100.00	1.00	0.0000	✗	✓	20

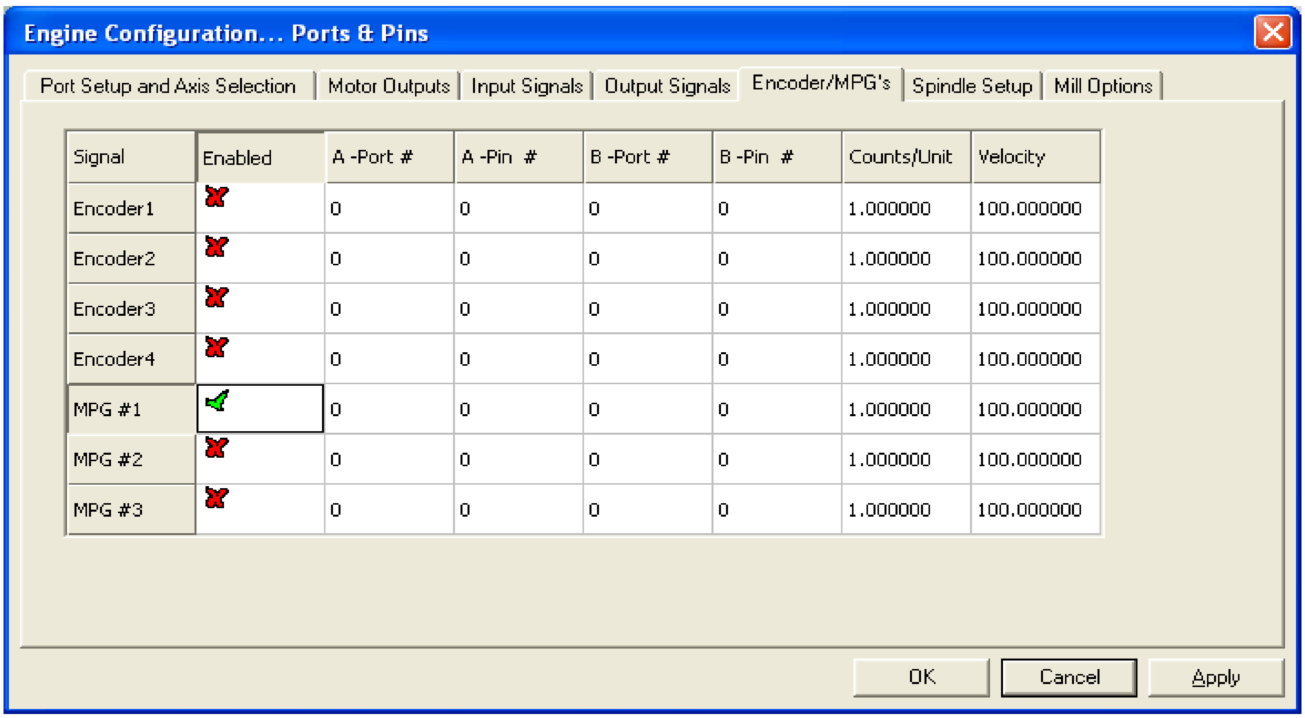
To change the axis direction, click on the Reversed colum for the axis you want to change the direction. A green check mark indicates the direction is now reversed.

Homing Offset lets you define the home position co-ordinates. When homing sequence is complete, the axis machine position is set to this value. Home Neg, changes the default homing direction.

The DSPMC offers a number of different homing types for each axis. Please review section A.15 Homing to select the correct homing sequence.

7.9 Manual Pulse Generation - MPG

DSPMC allows using a quadrature encoder as a MPG source. The encoder can be connected to the dedicated encoder inputs on J6 and J7 (HardEncoder), as well as to J4 and J5 digital inputs (*SoftEncoder*). In the dspmc.xml file you specify the MPG source wired.

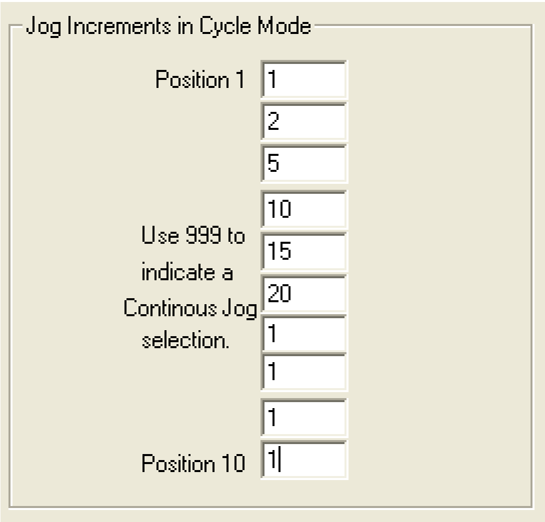


To turn on MPG feature, make sure MPG #1 is checked green as shown above in the Ports and Pins window. Enter the Encoder resolution in the Counts/Unit field. The rest of the fields in this window are not used.

The *SoftEncoder* are available on digital inputs of connector J4 and J5. The pin assignments are as follows:

SoftEncoder 0 : A = J4 Pin 24, B = J4 Pin 12
SoftEncoder 1 : A = J4 Pin 25, B = J4 Pin 13
SoftEncoder 2 : A = J5 Pin 18, B = J5 Pin 6

You set the encoder multiplier in the General Config setting as shown below. You can use your own multiplier values in this window as well as use the standard .1, .01, .001, etc values.



See section A.16 MPG Settings – MpgSource1 for the xml file settings for MPG.

When MPG mode is selected, and a G-Code file is run, the plugin will switch to jog mode automatically in order to run the file. Once the file is complete or stopped, the mode will revert back to MPG.

7.10 Lathe Threading

Threading using DSPMC does not require any special setup in Mach. The only configuration needed is part of the dspMC.xml file. See section A.17 Threading for details on the xml file threading configuration.

The Mach3 Software requires that the Spindle RPM to be constant before the threading G-Code command is executed. But there is always some variation in RPM calculation. As the DSPMC plugin has no knowledge of which G-Code command is going to run, the plugin has no way to know when to keep the RPM value constant. To solve this problem, the plugin looks at ***OEMLED 1999*** to know when to pass the calculated RPM to Mach3 (so user can see the actual spindle RPM), or to pass the fixed programmed (in G-Code) RPM value for Mach threading logic. The user can add a macro in the G-code program to turn on OEMLED 1999 before any threading command, and to turn it off after the threading command, to workaround this issue.

7.11 Homing Using Limit Switches

DSPMC allows Limit switches to be used as Home Sensors. There is a setting in xml file that disables scanning Limit switches during Homing sequence. This setting should be turned on to use Limit switches as Home sensor. See section A.18 Using Limit Switches as Home Sensors for more information.

7.12 OEMDROs

The following table lists the OEMDROs used by the DSPMC plugin. Firmware version 63 or newer is required.

OEMDRO Index	Description
1300 thru 1305	RPM for J6 and J7 Encoders 0..5. The dspMC.xml must have the ***CountsPerRev_x,y,a,b,c*** defined. These RPM values are not used by the Threading Logic.
1310 thru 1311	RPM for I/O Toggle Counters. **I/O Toggle Counter 0: Digital Input 18 (J5 Pin 19)** **I/O Toggle Counter 1: Digital Input 19 (J5 Pin 7)** The dspMC.xml must have the ***CountsPerRev_x,y*** defined. These RPM values are not used by the Threading Logic.
1320 thru 1327	Analog Input data from ADC channels 0..7. Available only with 7761-M model.
1330 thru 1332	RPM for Digital Input Encoders (SoftEncoder) on J4 and J5. **SoftEncoder 0** : A+ On J4 Pin 24, B+ On J4 Pin 12 **SoftEncoder 1** : A+ On J4 Pin 25 B+ On J4 Pin 13 **SoftEncoder 2** : A+ On J5 Pin 18 B+ On J5 Pin 6 The dspMC.xml must have the ***CountsPerRev_x,y,z*** defined. These RPM values are not used by the Threading Logic.
1340	Threading RPM. This RPM is calculated based on the parameters defined in the Threading section of dspMC.xml file, and is used by the Threading Logic.

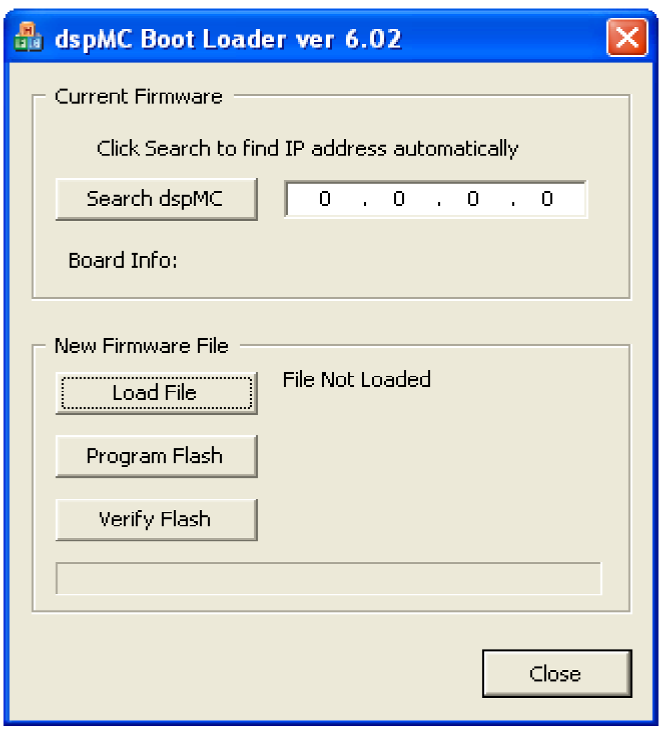
8. DSPMC Software Upgrade Tool

The Boot Loader software tool for DSPMC is used to re-program (Flash) the board software (Firmware). The latest Firmware file is available from the factory on request. The file is sent in compressed zip format, so you will need to unzip the file and extract the binary file before programming.

NOTE: Before programming the Firmware, please close all programs that are communicating with the DSPMC board. This includes programs running on other computers that are connected to the DSPMC board over Ethernet.

The following steps describe the procedure to upgrade the firmware.

1. Double Click the 'DSPMC Firmware Upgrade' icon that was installed during software installation. The following window appears:



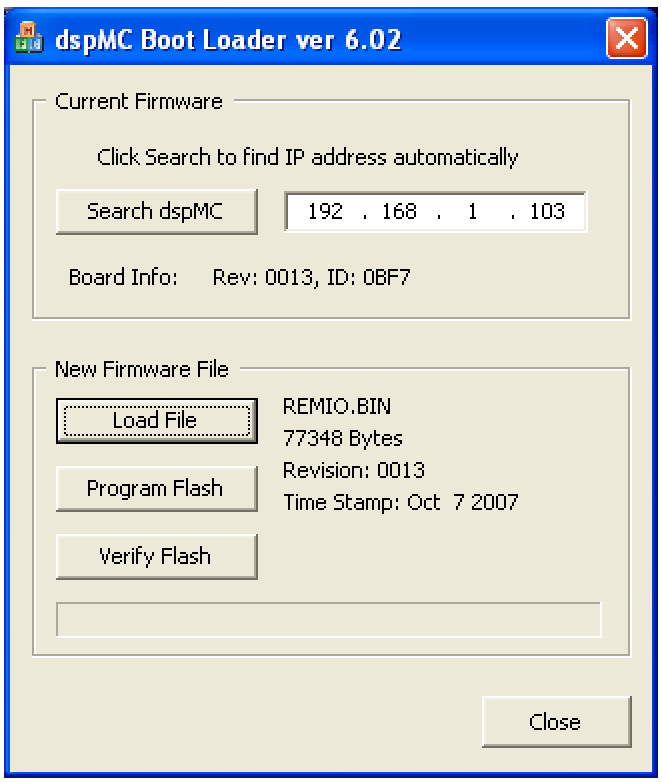
2. Click on the Search dspMC button to setup the connection.

This window shows that the connection is correctly established. Click OK. If the connection cannot be established, please refer to Network Connection Setup chapter to setup communications with the device.

Once connected, the main window shows the current board information, ie, the firmware revision and the board serial ID. If the firmware file you received from factory has the same revision number, then you do not need to re-program the board.

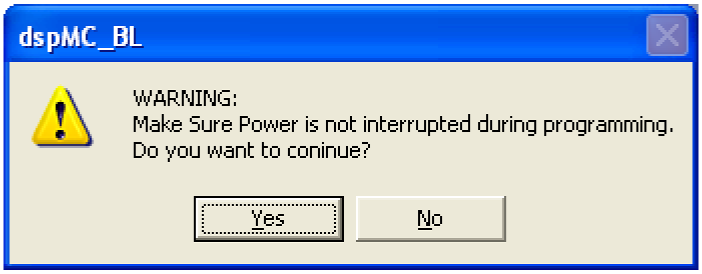
3. Click on the Load File button and search for the firmware (.bin) file that you received from factory.

Once the file is selected, a sample would look like this:

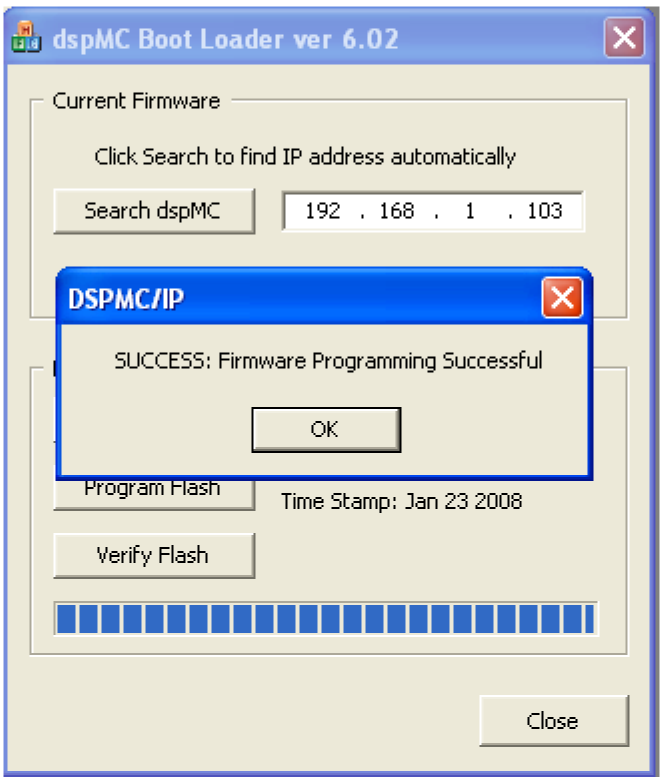


The version number and time-stamp of the firmware file you opened is shown on the main window. Please make sure this the correct version you intend to flash.

4. Click on the Program Flash Button. Please make sure that the board power supply does not get interrupted during this process. Click on the yes button to start. If you are not sure, click ‘No’ and provide a stable power supply before you try programming again.



Once you click 'Yes', the software will start programming the board. The progress bar will show the programming progress. After the programming is done, the software will verify if the programming is successful. A 'SUCCESS' message is displayed if firmware programming and verification is successful as shown below.



If an error is detected, it will show the location of the first data mismatch. You can try programming again by clicking on Program Flash Button to correct the error.

Note:
Note: If an error is detected during programming or verification, you can retry programming as many times as you like, as long as you do not turn off power to the DSPMC board. Once the board power is turned off, it will try to load the new firmware the next time you power up.

After a successful programming and verification, the DSPMC board must be restarted to execute the newly loaded program. This can be accomplished by turning off the power, and then turning back on again after 5 seconds.

Appendix A – dspMC.xml Format

XML format of DSPMC configuration is shown below. Only axis 0 is shown. All other axes follow the same format. At the end of the listing, detailed description if each parameter is described.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- This file represents configuration database for dspMC Motion Controller -->
<Mach3Config>

        <Spindle>
                <DAC>7</DAC>
        </Spindle>

        <MaxBufferLevel> 50 </MaxBufferLevel>   <!-- upto 100% for 4096 points/axis -->

        <BackLash>                              <!-- Raw Encoder Counts -->
                <Counts_x> 0 </Counts_x>
                <Counts_y> 0 </Counts_y>
                <Counts_z> 0 </Counts_z>
                <Counts_a> 0 </Counts_a>
                <Counts_b> 0 </Counts_b>
                <Counts_c> 0 </Counts_c>
        </BackLash>

        <RPM>
                <CountsPerRev_x> 2000 </CountsPerRev_x>
                <CountsPerRev_y> 2000 </CountsPerRev_y>
                <CountsPerRev_z> 2000 </CountsPerRev_z>
                <CountsPerRev_a> 2000 </CountsPerRev_a>
                <CountsPerRev_b> 2000 </CountsPerRev_b>
                <CountsPerRev_c> 2000 </CountsPerRev_c>
        </RPM>

        <HomingIgnoreLimitSw> 1 </HomingIgnoreLimitSw>

        <MpgSource1>
                <Type>HardEncoder</Type> <!—SoftEncoder or HardEncoder-->
                <Index>2</Index>
        </MpgSource1>

        <Threading>
                <RPM_Sync_Source> HardEncoder </RPM_Sync_Source> <!-- HardEncoder or DigitalInput -->
                <RPM_Sync_Index> 1 </RPM_Sync_Index>
                <RPM_Count_Per_Rev> 2000 </RPM_Count_Per_Rev>
                <RPM_Sampling_Milli_Second> 500 </RPM_Sampling_Milli_Second> <!-- 1ms thru 10000ms-->
        </Threading>

        <Probing>
                <ProbingType> DualCycle </ProbingType>
                        <!-- SingleCycle or DualCycle (coarse and fine moves) -->
                <ProbingMoveBack> 0.35 </ProbingMoveBack>
                        <!-- Distance in mm/inch to move back from touch point before starting the second (fine)
                        move -->
```

```
                <ProbingFeedRate> 15.5 </ProbingFeedRate>
                        <!-- Feedrate for the second cycle (fine move) in mm or inch per minute -->
        </Probing>

</Mach3Config>

<dspMCConfig>

        <HardEncoderPolarity>  <!-- 1 = as is, -1 = reversed  -->
                <Channel_0> 1 </Channel_0>
                <Channel_1> -1 </Channel_1>
                <Channel_2> 1 </Channel_2>
                <Channel_3> 1 </Channel_3>
                <Channel_4> 1 </Channel_4>
                <Channel_5> 1 </Channel_5>
                <Channel_6> 1 </Channel_6>
                <Channel_7> 1 </Channel_7>
        </HardEncoderPolarity>

        <Axis0>
                <PID_Filter>
                        <P>30000</P>
                        <I>1000</I>
                        <D>0</D>
                        <!-- Feed Forward Command Gain -->
                        <ffcGain>0</ffcGain>
                        <!-- Feed Forward Command Derivative Gain -->
                        <ffcdGain>0</ffcdGain>
                        <!-- Max Limit on Error (Command - Feedback) -->
                        <MaxError>0</MaxError>
                        <!-- Max Limit on Integral Error -->
                        <MaxError_I>200</MaxError_I>
                        <!-- Max Limit on Differential Error -->
                        <MaxError_D>0</MaxError_D>
                        <!-- Max Limit on Following Error -->
                        <MaxError_F>0</MaxError_F>
                        <!-- Range of Error where PID Loop does not respond -->
                        <Deadband>0</Deadband>
                        <!-- Max Limit on Command Derivative -->
                        <MaxCmd_D>0</MaxCmd_D>
                        <!-- Divisor for PID Calculation -->
                        <Scale>15000</Scale>
                        <!-- Constant to be added after PID calculation and scalling -->
                        <Offset>0</Offset>
                </PID_Filter>
                <ControlInput>
                        <!-- Source: TrajectoryPlanner, CommandPosition, Counter, A2D -->
                        <Source>CommandPosition</Source>
                        <Index>0</Index>
                        <Gain>1.000000</Gain>
                </ControlInput>
                <!-- Feedback: Counter, A2D -->
                <PID_Feedback>
                        <Source>Counter</Source>
                        <Index>0</Index>
                        <Gain>1.000000</Gain>
```

```
        </PID_Feedback>
        <ControlOutput>
                <!-- Output: DAC, Stepper -->
                <Type>DAC</Type>
                <Index>0</Index>
        </ControlOutput>
        <AmpEnable>
                <!-- Bit Index 0=undefined, 1..16 bit number - 1 -->
                <Index>0</Index>
        </AmpEnable>

        <Homing>

                <Type> HomeSensor </Type>     <!-- Options: HomeSensor, IndexPulseOnly -->

                <UseIndexPulse>1</UseIndexPulse>
                        <!-- Options: 1 = use index pulse, 0 = donot use index pulse -->
        </Homing>

    </Axis0>…..
```

Description of XML parameters:

A.1 Spindle DAC

The Spindle DAC configuration selects the analog output (DAC) for the spindle speed control. The DAC output voltage varies from 0 to 10volts. Any Analog output position can be selected from is 0 to 7. Make sure the selected DAC output number is not used anywhere else in the dspmc.xml file. If it is used, then enter **"Undefined"** in the ***ControlOutput Type*** section.

The following section defines DAC output 6 for spindle.

```
<Spindle>
    <DAC>6</DAC>
</Spindle>
```

A.2 MaxBufferLevel

This section defines how much command position buffering will be done inside the DSPMC controller. The total size of the buffer is 4096 points per axis. These points are consumed by the DSPMC at 1 KHZ. To get faster response time on feedrate changes, you may select a lower value, but the side effect is that if the PC software slows down and cannot sustain the motion data rate to the DSPMC, then the motion could be jerky.

The valid rage for this section is 1…100 percent.

eg,
MaxBufferLevel> 50 </MaxBufferLevel>

A.3 BackLash

The following example shows how to calculate backlash counts for x axis:

Example Backlash = 0.010" on x axis

Encoder = 4000 counts per revolution (1000 count encoder)
Lead Screw = 10 revolution per inch

You have 40000 counts per inch and backlash in encoder counts will be 40000 x 0.010 = 400

Therefore in the dspmc.xml you will enter 400 as follows:

```
<BackLash>
    <Counts_x> 400 </Counts_x>
</BackLash>
```

A.4 RPM

For hardware encoders on J6 and J7 and SoftEncoders on J4 and J5, The CountsPerRev_x value specifies the encoder counts per rev x 4. For example for a 500 count encoder you enter 2000.

```
<RPM>
        <CountsPerRev_x> 2000 </CountsPerRev_x>
        <CountsPerRev_y> 4000 </CountsPerRev_y>
</RPM>
```

See Section 7.12 OEMDROs for more information on how to use these RPM parameters.

A.5 PID Filter

These values define the co-efficient of PID filters for each axis. See Section 6.2 PID Filter Configuration for definition of these parameters. The PID filter is run at 5KHz for each axis.

More information on PID control is available at http://en.wikipedia.org/wiki/PID_control.

A.6 ControlInput Source

ControlInput Source defines the input type (or set-point) for the PID filter for a particular axis. The possible values are:

CommandPosition: This setting is used for PC applications that generate command position itself, e.g. Mach3. The DSPMC makes the axis follow these command positions. The command position data is provided to DSPMC at a rate of 1KHz for each axis.

TrajectoryPlanner: The DSPMC controller also has built in trajectory generator (or planner) for each axis. This allows motion profiles to be generated automatically, based on user provided distance, accel, and velocity settings.

Counter: Use a shaft encoder counter as the PID input.

A2D: Use one of the analog inputs as the PID input. This allows PID to be used for temperature and process control, in addition to motion control applications.

A.7 ControlInput Index

Defines the index of the PID input source. Normally this is equal to the axis number. For slave application, this defines the data index of the master.

A.8 ControlInput Gain

Defines the multiplier before applying the data to the PID input. E.g. if Analog Input 0 is the source, analog input 0 value is first multiplied by the Gain and then used in the PID calculation.

A.9 PID Feedback Source

PID Feedback Source defines the feedback type for the PID filter for a particular axis. The possible values are:

Counter: Use a shaft encoder counter 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.

A.10 PID Feedback Index

Defines the index of the PID feedback source.

A.11 PID Feedback Gain

Defines the multiplier before applying the data to the PID feedback. E.g. if Encoder Counter 0 is the feedback source, Encoder Counter 0 value is first multiplied by the Gain and then used in the PID calculation.

A.12 ControlOutput Type

ControlOutput 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 Servo amplifiers 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 when the axis is disabled and the index is used on some other axis. For example, if DAC 6 is used for Spindle speed and DAC 6 is also used as ControlOuput for the axis, but the axis is not part of the system, then the user must define the ControlOutput type as undefined. If the axis is part of the system, then some other index should be used for Spindle or axis output.

A.13 ControlOutput Index

Defines the index of the PID Output type.

A.14 AmpEnable Index

Defines the index of the digital output to be used as the enable signal for the output servo amplifier or drive. Valid range is 0 thru 16. A value of 0 means no output is used as Enable signal for this axis. Value of 1..16 means digital output 0 .. 15.

A.15 Homing

This section defines homing sequence for each axis. Two types of homing sequence are supported:

- **HomeSensor** (homing with or without Index Pulse)
- **IndexPulseOnly** (Use only the Index pulse to Home)

For *HomeSensor* method, the axis moves in configured direction until home sensor is found. It then moves in the opposite direction for configured length. If *UseIndexPulse* is set to 1, after finding the home sensor, the axis moves in opposite direction to locate the index pulse and then continue moving for the configured length.

For *IndexPulseOnly*, the axis moves in the configured direction to locate the index pulse to home the axis. It then moves in the opposite direction for the configured length.

A.16 MPG Settings – MpgSource1

This section defines MPG (Manual Pulse Generation) Quadrature encoder source. Both Differential and Single Ended Encoder types are supported. Differential encoder can be hooked up to any of the six encoder channel available on connectors J6 and J7. These encoders are defined as ***HardEncoder*** in the xml file. Single Ended encoders (defined as ***SoftEncoder***) can be hooked up to the Digital Inputs available on Connector J5.
The pin assignments for available *SoftEncoder* are as follows (***Require Firmware 63 or newer***):

SoftEncoder 0 : A+ On J4 Pin 24,
B+ On J4 Pin 12
SoftEncoder 1 : A+ On J4 Pin 25
B+ On J4 Pin 13
SoftEncoder 2 : A+ On J5 Pin 18
B+ On J5 Pin 6

Example 1: Connect a differential encoder for MPG on Encoder Channel 5 (J7)

```
<MpgSource1>
    <Type> HardEncoder </Type>
    <Index>5</Index>
</MpgSource1>
```

Example 2: Connect a single-ended encoder for MPG on J5 Digital Inputs, pin 18 and 6.

```
<MpgSource1>
    <Type> SoftEncoder </Type>
    <Index>1</Index>
</MpgSource1>
```

A.17 Threading

This section defines parameter for CNC threading. The following is an example configuration:

```
<Threading>
        <RPM_Sync_Source> HardEncoder </RPM_Sync_Source>
                        <!-- HardEncoder or DigitalInput -->

        <RPM_Sync_Index> 1 </RPM_Sync_Index>

        <RPM_Count_Per_Rev> 2000 </RPM_Count_Per_Rev>

        <RPM_Sampling_Milli_Second> 100 </RPM_Sampling_Milli_Second>
                        <!-- 1ms thru 10000ms-->
</Threading>
```

A.17.1 RPM_Sync_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.

The two possible values for **RPM_Sync_Source** parameter are: "***HardEncoder***" and "***DigitalInput***".

1. When ***HardEncoder*** is selected, the spindle feedback encoder must be connected to one of the encoder inputs on J6 and J7 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.

2. When ***DigitalInput*** is selected, the spindle feedback is generated by a single line pulse train. The pulse train is used to calculate the spindle RPM as well as used for Sync pulse to launch the threading cycle.
   There are two dedicated inputs on J5 for the spindle pulse train. These are also called I/O Toggle Counters.

   - I/O Toggle Counter 0:   Digital Input 18 (J5 Pin 19)
   - I/O Toggle Counter 1:   Digital Input 19 (J5 Pin 7)

A.17.2 RPM_Sync_Index   This parameter defines the encoder index for Spindle speed feedback. Below is the range for this index:
HardEncoder: index range is 0...5.
DigitalInput: index range is 0...1.

A.17.3 RPM_Count_Per_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.

A.17.4 RPM_Sampling_Milli_Second 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.

A.18 Probing

This section defines parameter for CNC Probing feature. For the probing cycle, the axis, probing feedrate, and the probe switch are set by the PC software, eg Mach3.

The following is an example configuration:

```
<Probing>
        <ProbingType> DualCycle </ProbingType>
                <!-- SingleCycle or DualCycle (coarse and fine moves) -->

        <ProbingMoveBack> 0.35 </ProbingMoveBack>
                <!-- Distance in mm/inch to move back from touch point before
                                                starting the second (fine) move -->

        <ProbingFeedRate> 15.5 </ProbingFeedRate>
                <!-- Feedrate for the second cycle (fine move) in mm or inch per minute -->
</Probing>
```

A.18.1 ProbingType This parameter defines the probing method:

SingleCycle: Axis starts the probing move. As soon as the probe switch is on, the current position is captured and the probing sequence is complete.

DualCycle: Axis starts a first probing move (called coarse move). As soon as the probe switch is on, the axis stops, and backs off distance specified in the **ProbingMoveBack** parameter. The axis then starts the second move, called fine move. The direction is same as the coarse move. The feedrate for the fine move is defined by the third parameter **ProbingFeedRate.**

A.18.2 ProbingMoveBack This parameter defines the distance to move back to start the fine move. It is applicable only in the DualCycle mode.

A.18.3 ProbingFeedRate This parameter defines the feedrate for the fine move. It is applicable only in the DualCycle mode.

A.19 Using Limit Switches as Home Sensors

When Homing using Limit switches, the ***HomingIgnoreLimitSw*** parameter should be set to 1 as follows:

```
<HomingIgnoreLimitSw> 1 </HomingIgnoreLimitSw>
```

For Home sensors using dedicated inputs, this parameter should be set to 0, or removed from the xml file.

A.20 Hardware Encoder Counter Polarity

The ***HardEncoderPolarity*** parameter is used to reverse the direction of the encoder counters. If A/B signals are connected such that it does not match the PID control direction, the system will not be able to arm. To fix this issue, the hardware A and B signals can be reversed, or simply use this section to change the direction in the software.

```
<HardEncoderPolarity>                <!-- 1 = as is, -1 = reversed  -->
        <Channel_0> 1 </Channel_0>
        <Channel_1> -1 </Channel_1>
        <Channel_2> 1 </Channel_2>
        <Channel_3> 1 </Channel_3>
        <Channel_4> 1 </Channel_4>
        <Channel_5> 1 </Channel_5>
        <Channel_6> 1 </Channel_6>
        <Channel_7> 1 </Channel_7>
</HardEncoderPolarity>
```

Note that this software encoder polarity setting only applies to the encoder counter. The Index pulse signal polarity is not affected by this setting.

License Agreement

Before using the DSPMC/IP 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/IP 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.