HICON Mach4
Software Integration

HiCON Ethernet Motion Controller

Document Revision 1.9.5

(Updated Jan 19, 2026)

© 2026 Vital Systems Inc

Atlanta, GA USA

For more information, please visit the product web page:

www.vitalsystem.com/HiCON

Contents

LICENSE AGREEMENT **3**

INTRODUCTION **4**

INITIAL SETUP **5**

- Identify the Controller 5
- Mach4 Setup 5
- HICON Mach4 Plugin Setup 6
- MachPro Setup 8
- HICON MachPro Plugin Setup 8
- Starting Mach4/MachPro with HICON 11
- Mach4/MachPro Configuration 13
- Digital I/O 14
  - On Board Digital I/O 14
  - Mapping Mach Inputs to HICON Digital Input Pins 15
  - Mapping Mach Outputs to HICON Digital Output Pins 17
  - EtherCAT Digital I/O 18
- Motor Parameters 19
- Mach Axis Mapping 21
- Axis Homing Setup and Soft Limits 22

HICON SYSTEM CONFIGURATION **23**

- Communication 24
- Spindle Config 25
- Misc. Config 26
- Hardware Encoder Polarity 26
- Encoder Debounce 26
- Update HICON (System Tab) 26
- Analog Outputs 27
- Analog Inputs 27
- EtherCAT 27

MPG SETUP **28**

- MPG Encoder Selection 28
- MPG Jog Increments 29

HICON MOTOR CONFIGURATION **30**

- Control Input Parameters 31
- Control Output Parameters 31
- Feedback Parameters 31
- Backlash Compensation 33
- PID Filter 33
- Homing Parameters 33
- Test Motion 35

HICON STATUS WINDOW 37

ETHERCAT CONFIGURATIONS 39

Mapping Additional EtherCAT I/O with Mach: 39

Ethercat Homing (Methods Built-in in EtherCAT Drive) 42

Ethercat Status Window 44

ADDITIONAL SETUP 45

Master-Slave Axis Setup 45

Dependent Homing 45

Independent Homing 45

Spindle Setup 46

Analog Voltage Spindle 46

Setup Step/Dir or EtherCAT Motor as a spindle 48

Synced Analog Outputs - M67 51

Counts Per Unit Calibration 52

Scripting and Mach HICON Registers 53

I/O Expansion Using Smart3G cards over Modbus/Tcp 55

License Agreement

Before using the HICON 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 HICON 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 HICON board and its software. I understand that the HICON 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 HICON, 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 HICON 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 HiCON motion controller, you agree to the license agreement.

Introduction

IMPORTANT

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

This document **DOES NOT** serve as a primer or tutorial for the use of Mach. As such, readers without basic understanding of Mach, 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 HiCON Controller with Mach Software.

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

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

Initial Setup

Identify the Controller

Over the decade Vital System has made the HiCON platform reliable and has developed multiple motion controllers for multiple applications. It is important to identify your controller as some controllers have additional capabilities that other controllers may not have. This document covers all the settings across all of our motion controllers.

List of currently supported Motion Controllers.

- **HiCON Integra 7866** (Industrial Applications)
- **HiCON DSPMCv3** (Flagship - Industrial Applications with analog and stepper capability with 96 I/O points)
- **EtherCAT EC01 Motion Controller** (Industrial Applications)
- **HiCON Integra 7766** (Industrial Applications)
- **HiCON Mini 77E4** (low-cost Hobby Use and Industrial Applications)

We also have 3 other motion controller that we customize for our OEM partners. Any advanced feature that we implement or a bug we fix gets propagated across all our motion controller for maximum compatibility and reliable motion control system.

Mach4 Setup

To install Mach4 on the target machine, please follow the steps below.

1. Download and install the Mach4 application from the machsupport website.

2. If a Mach4 Hobby or Industrial license was purchased from Vital System Inc. You should have received an email containing a registration code to redeem the license.

3. Further instructions for licensing your Mach4 copy may be found within the aforementioned email. Licenses may be registered on up to 5 PCs.

4. Once Mach4 has been successfully installed, the HiCON plugin must be installed to facilitate communication with the HiCON unit.

> **NOTE**: *Documents and User Guides for Mach4, supported GCode commands, and lua scripting may be found in the "Docs" folder in the Mach4 install directory (usually C:/Mach4Hobby/Docs).*
>
> *Additionally, the Machsupport forum may also be consulted for information. Product feedback and suggestions may also be posted on the forum.*

HICON Mach4 Plugin Setup

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

1. Download the Mach4 HICON plugin available on the controller page.

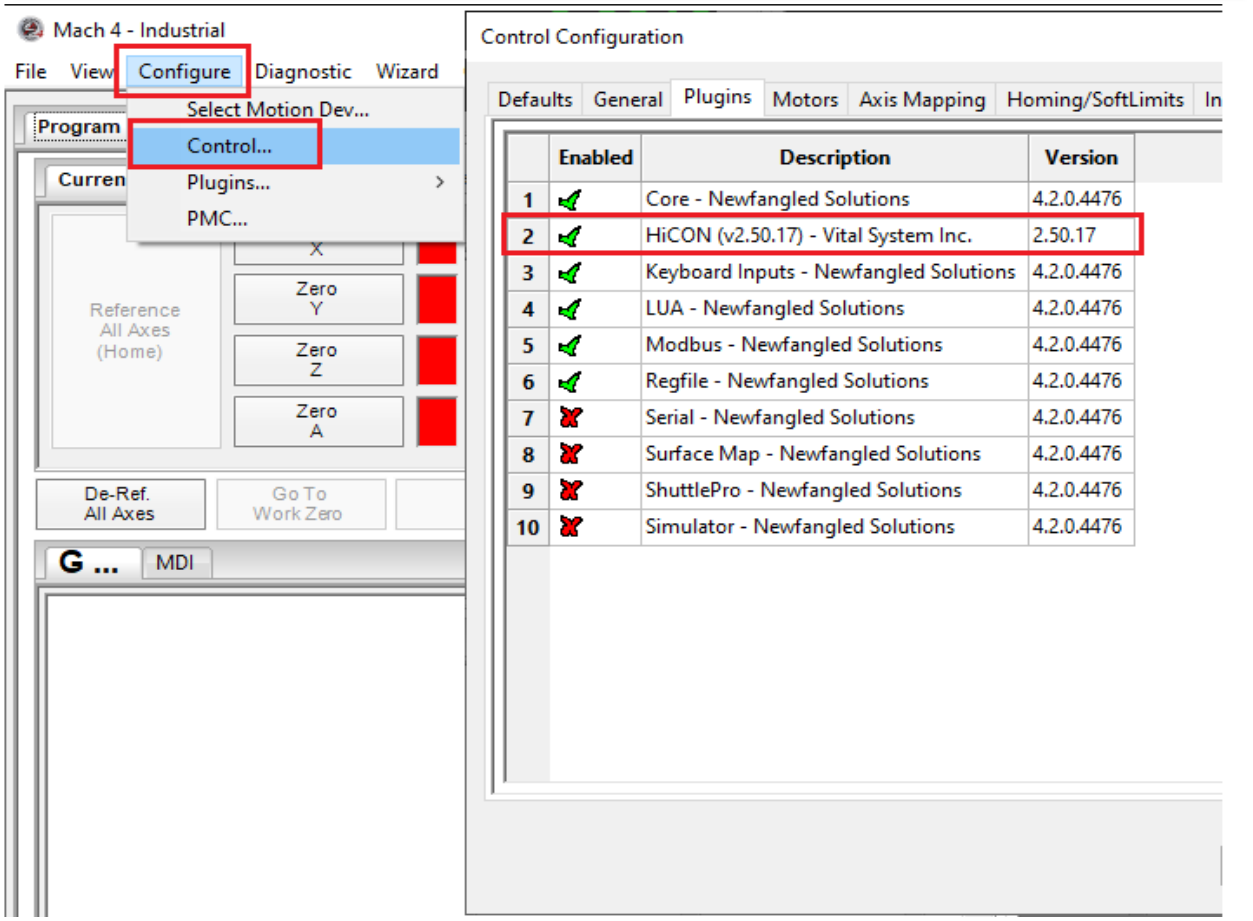
2. Open the zip file and the plugin files "*HiCONPlugin.mpw" and "HiCONPlugin.sig"* can be found within.

3. To use the HICON plugin for Mach4, extract or copy the Content of the Zip File to the Plugins folder in the Mach4 install directory (usually *C:/Mach4Hobby/Plugins*).

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

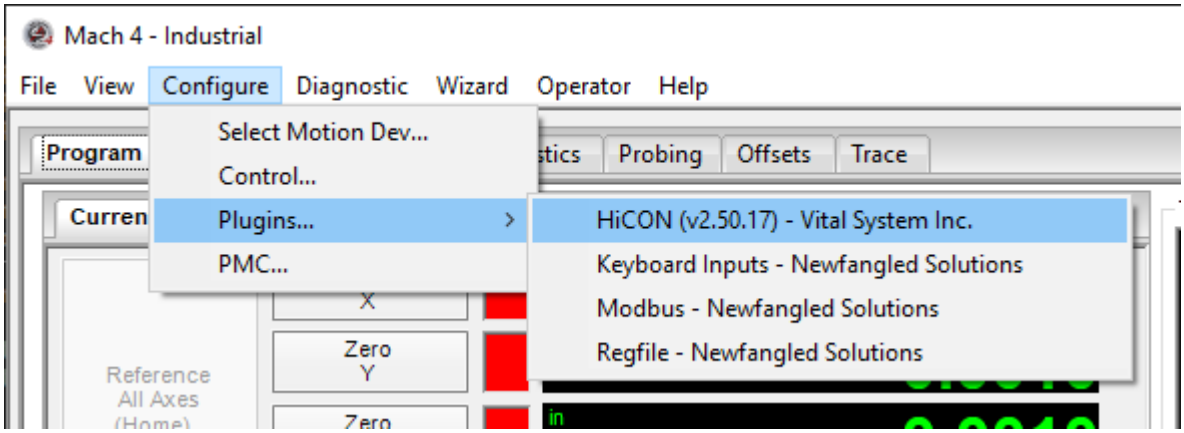
5. Enable the HICON plugin by opening the "*Mach4 Configuration*" window ("*Configure->Control*" at the top of the Mach4 main window). The list of currently installed plugins can be viewed under the "*Plugins*" tab. Make sure the HiCON plugin is enabled (green check as shown below). Click "Apply", then "OK".

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



NOTE: *Some motion device plugins may interfere with each other. It is advisable to disable, or even remove other motion device plugins that are not being used.*

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



NOTE: *Take note of this procedure to access the HiCON Plugin Configuration as a large amount of operational parameters are configured from the HiCON plugin screen.*

MachPro Setup

To install MachPro on the target machine, please follow the steps below.

1. Download and install the MachPro application from the machsupport website.

2. If a MachPro Hobby or Industrial license was purchased from Vital System Inc. You should have received an email containing a registration code to redeem the license.

3. Further instructions for licensing your MachPro copy may be found within the aforementioned email. Licenses may be registered on up to 5 PCs.

4. Once MachPro has been successfully installed, the HiCON plugin must be installed to facilitate communication with the HiCON unit.

> **NOTE**: *Documents and User Guides for MachPro, supported GCode commands, and lua scripting may be found in the "Docs" folder in the MachPro install directory (usually C:/MachPro/Docs).*
>
> *Additionally, the Machsupport forum may also be consulted for information. Product feedback and suggestions may also be posted on the forum.*

HICON MachPro Plugin Setup

To setup the HICON plugin with MachPro, please follow the steps below.

1. Download the MachPro HICON plugin available on the controller page.

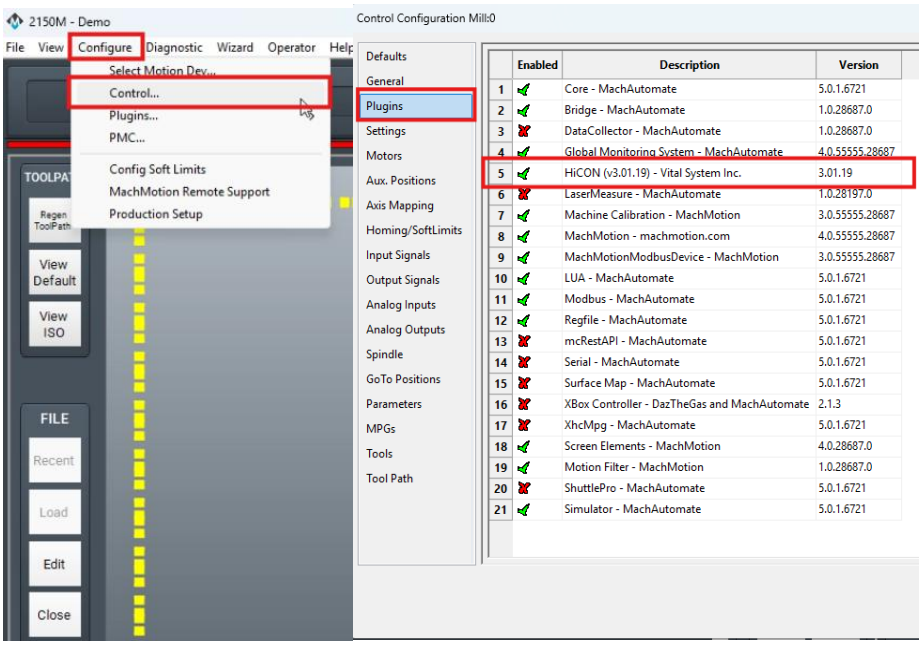
2. After opening the ZIP file, you will find the plugin files "*HiCONPlugin.mpw" and "HiCONPlugin.sig*", along with the required dependency folder.

7. To use the HICON plugin for MachPro, extract or copy the Content of the Zip File to the Plugins folder in the MachPro install directory (usually *C:/MachPro/Plugins*).

3. To launch MachPro with HICON plugin, double-click on the "*MachGUI.exe*" software icon on the desktop (or MachPro install folder).

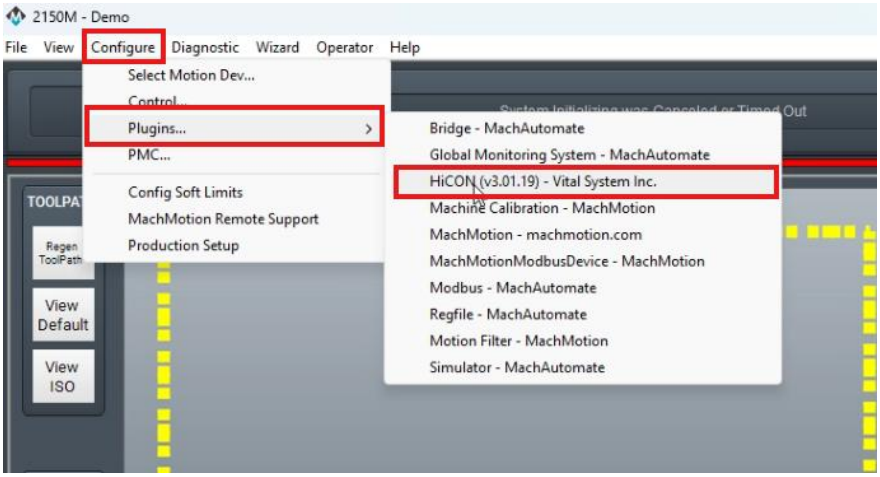
4. Enable the HICON plugin by opening the "*MachPro Configuration*" window ("*Configure->Control*" at the top of the MachPro main window). The list of currently installed plugins can be viewed under the "*Plugins*" tab. Make sure the HiCON plugin is enabled (green check as shown below). Click "Apply", then "OK".

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

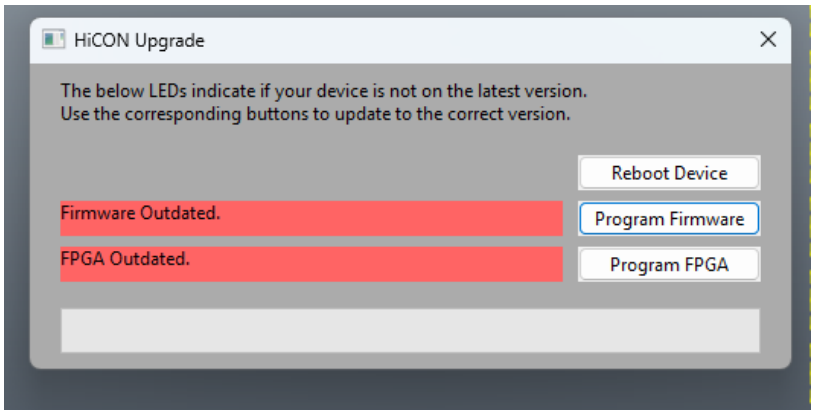


NOTE: *Some motion device plugins may interfere with each other. It is advisable to disable, or even remove other motion device plugins that are not being used.*

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



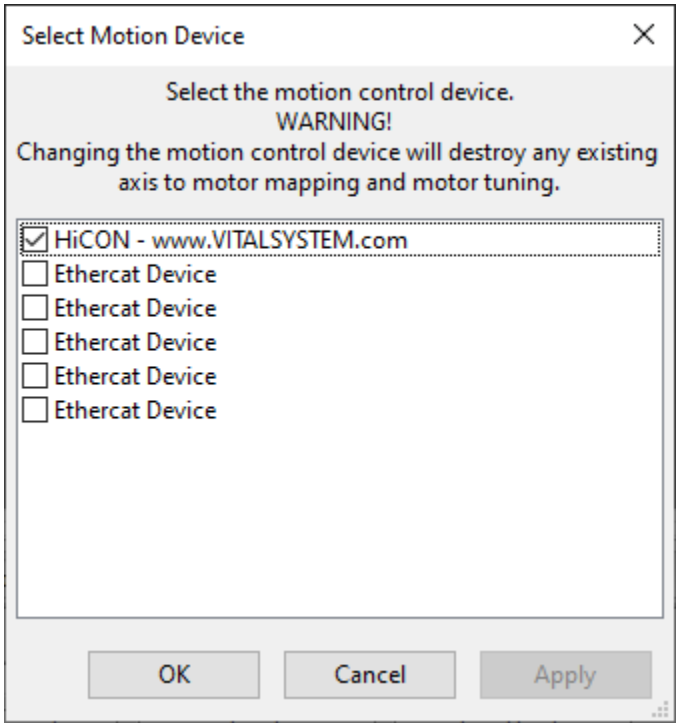
6. After successfully enabling the plugins, a firmware-upgrade popup will appear the next time MachPro starts. From there, you can update the device to the latest firmware version



NOTE: *Take note of this procedure to access the HiCON Plugin Configuration as a large amount of operational parameters are configured from the HiCON plugin screen.*

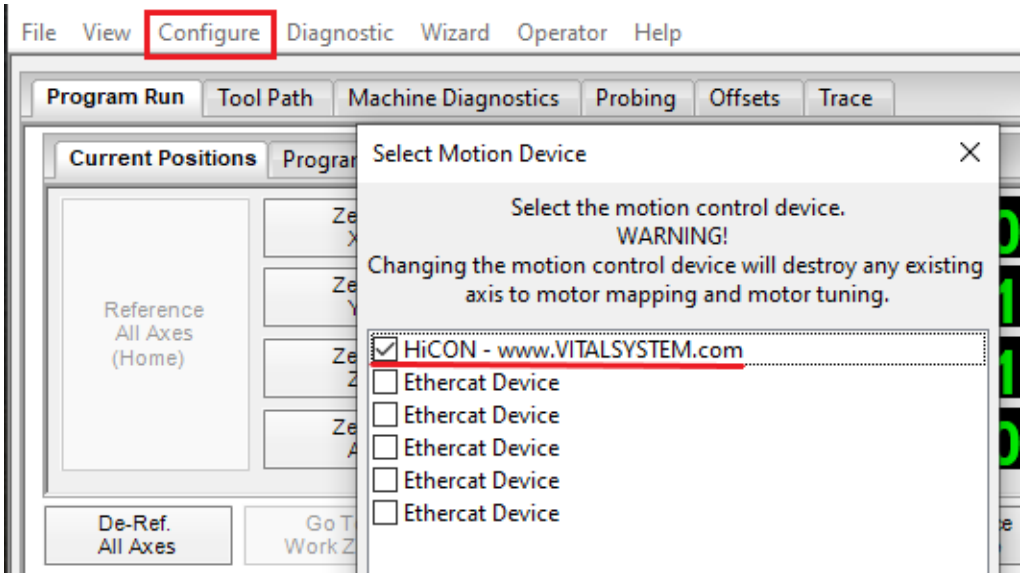
Starting Mach4/MachPro with HICON

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 HICON as an option on Mach startup. Make sure this plugin is selected and click 'OK'. (*See image to the right*)

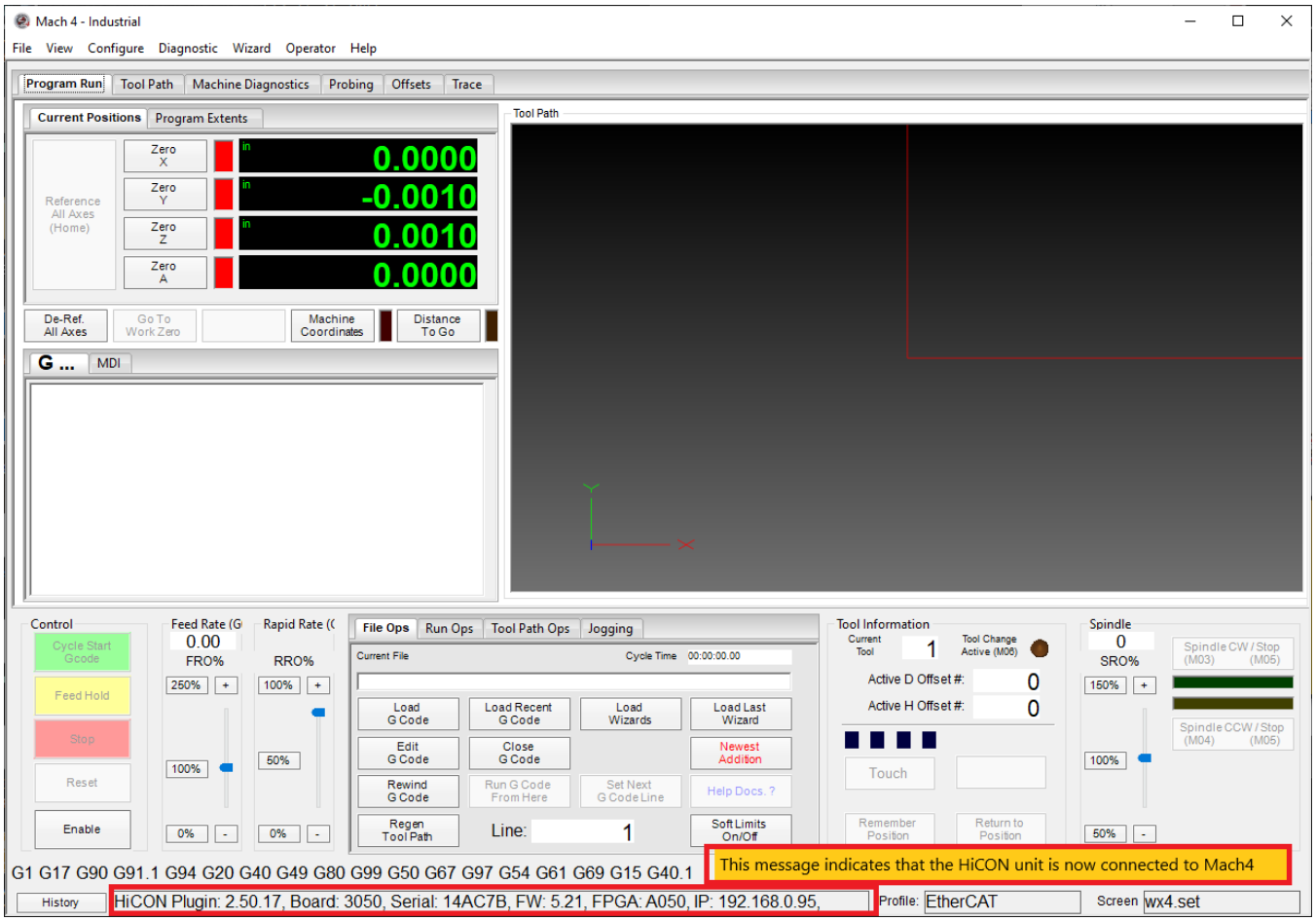


NOTE: *Mach must be restarted when the current motion device is changed or a new one is selected.*

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 HICON is powered up and properly connected on the network with a good quality CAT5e Ethernet cable. Mach will automatically search through all networks for any HICON and, if successful, will display a status message containing information for the currently connected HICON. (*See image below*).

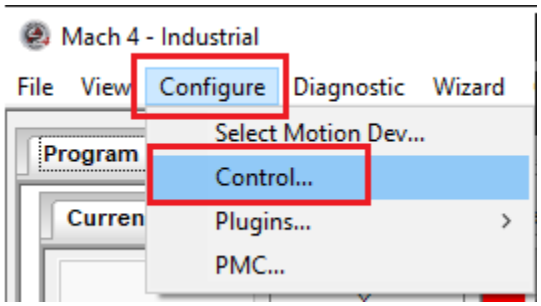


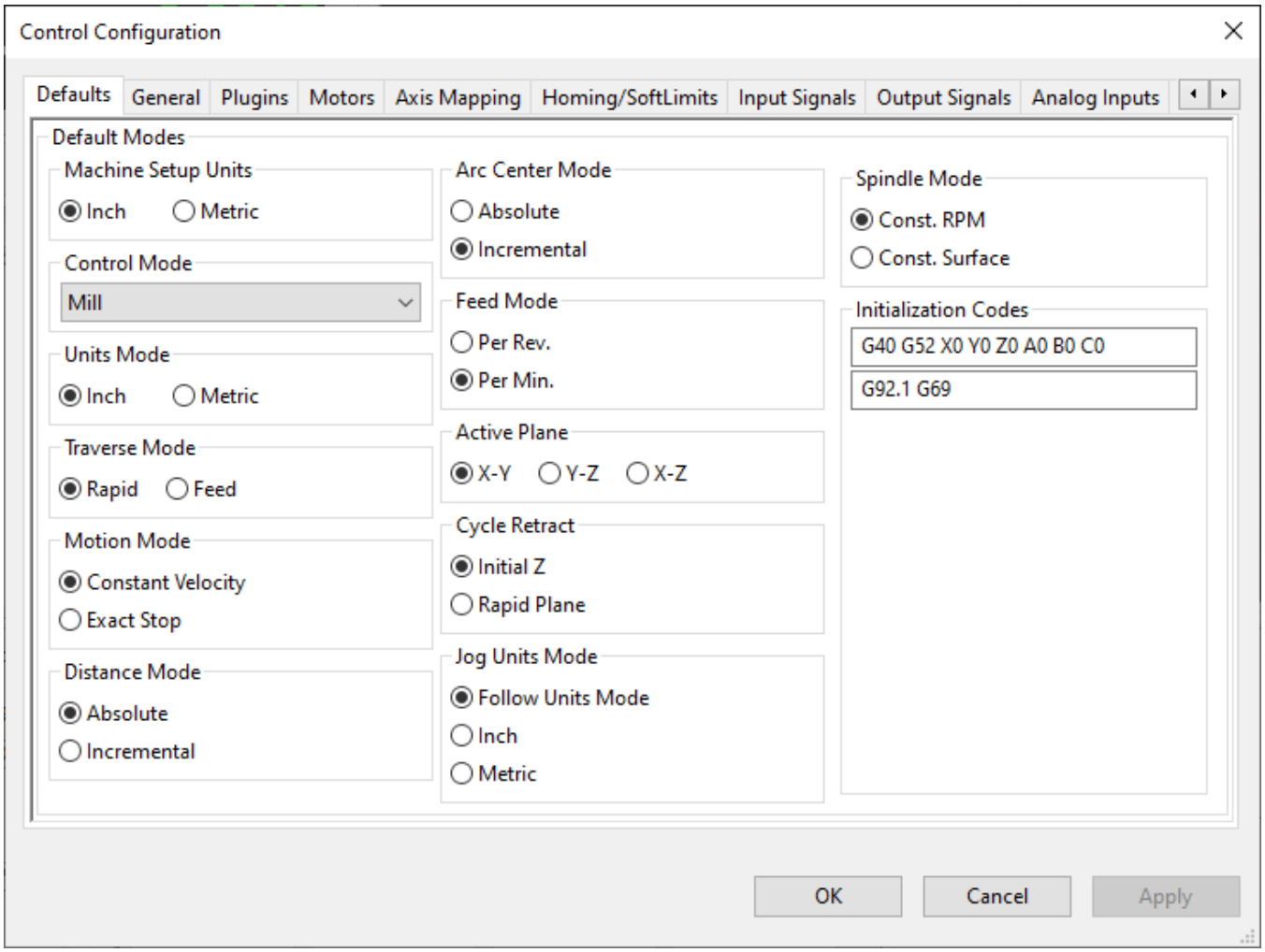
NOTE: *If you do not see the status message in the image below, then the plugin has not connected with any HICON on the network.*

- *Refer to the controllers user manual to resolve network connection issues or how to change the PC's network adapter IP address.*
- *Refer to the "**VSI Device Manager User Manual**" to change the HICON IP address.*
- *The HICON and PC IP addresses must be located on the same network.*

Mach4/MachPro Configuration

Open the Mach Configuration window click on *"Configure"*, and then *"Control…"* from the drop-down menu at the top of the Mach4/MachPro main window.





From here, click on the "Input Signals" tab, or the "Output Signals" tab to configure the Digital I/O Settings.

NOTE: *It is imperative to setup the necessary ESTOP and limit switches before arming the machine*. Failure to follow correct safety guidelines may result in personal injury or damage your machine in the event of a fault.

Digital I/O

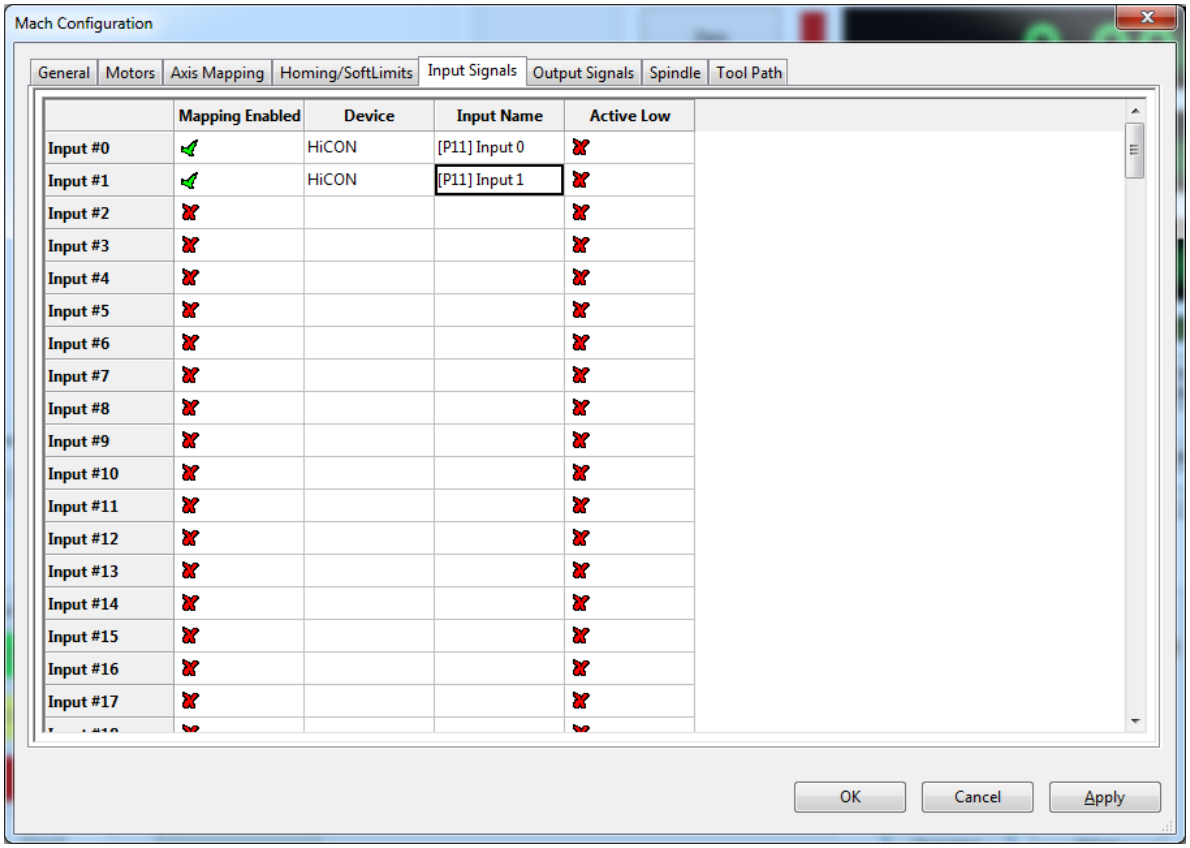
When it comes to mapping Digital I/O you can map either to on board I/O by selecting the device HiCON. For EtherCAT motion controller (EC01) you can map to one of the digital I/O from the EtherCAT device along with onboard I/O.

On Board Digital I/O

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

NOTE: *It is highly recommended to first set up the ESTOP and limit switches before attempting jogging (or motion in general) on the machine.* ***Global ESTOP mapping is P14 Input 15.***

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



Mapping Mach Inputs to HICON Digital Input Pins

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

Mach Input	HICON OEM J8 Input Pin	HiCON Integra Onboard Input	EtherCAT EC01 Onboard Input J8	HiCON BASIC Macro GetPin() Index
[P11] Input 0	10	J14, INP0	J8, INP0	11, 0
[P11] Input 1	11	J14, INP1	J8, INP1	11, 1
[P11] Input 2	12	J14, INP2	J8, INP2	11, 2
[P11] Input 3	13	J14, INP3	J8, INP3	11, 3
[P11] Input 4	14	J14, INP4	J8, INP4	11, 4
[P11] Input 5	15	J14, INP5	J8, INP5	11, 5
[P11] Input 6	16	J14, INP6	J8, INP6	11, 6
[P11] Input 7	17	J14, INP7	J8, INP7	11, 7
[P11] Input 8	18	J13, INP8		11, 8
[P11] Input 9	19	J13, INP9		11, 9
[P11] Input 10	20	J13, INP10		11, 10
[P11] Input 11	21	J13, INP11		11, 11
[P11] Input 12	22	J13, INP12		11, 12
[P11] Input 13	23	J13, INP13		11, 13
[P11] Input 14	24	J13, INP14		11, 14
[P11] Input 15	25	J13, INP15		11, 15

Mach Input	HICON OEM J9 Input Pin	HiCON Integra J7 Input Pin	7535 Breakout Board Input	HiCON BASIC Macro GetPin() Index
[P12] Input 0	10	10	0	12, 0
[P12] Input 1	11	11	1	12, 1
[P12] Input 2	12	12	2	12, 2
[P12] Input 3	13	13	3	12, 3
[P12] Input 4	14	14	4	12, 4
[P12] Input 5	15	15	5	12, 5
[P12] Input 6	16	16	6	12, 6
[P12] Input 7	17	17	7	12, 7
[P12] Input 8	18	18	8	12, 8
[P12] Input 9	19	19	9	12, 9
[P12] Input 10	20	20	10	12, 10
[P12] Input 11	21	21	11	12, 11
[P12] Input 12	22	22	12	12, 12
[P12] Input 13	23	23	13	12, 13
[P12] Input 14	24	24	14	12, 14
[P12] Input 15	25	25	15	12, 15

Mach Input	HICON OEM J11 Input Pin	HiCON Integra J8 Input Pin	7535 Breakout Board Input	HiCON BASIC Macro GetPin() Index
[P13] Input 0	10	10	0	13, 0
[P13] Input 1	11	11	1	13, 1
[P13] Input 2	12	12	2	13, 2
[P13] Input 3	13	13	3	13, 3
[P13] Input 4	14	14	4	13, 4
[P13] Input 5	15	15	5	13, 5
[P13] Input 6	16	16	6	13, 6
[P13] Input 7	17	17	7	13, 7
[P13] Input 8	18	18	8	13, 8
[P13] Input 9	19	19	9	13, 9
[P13] Input 10	20	20	10	13, 10
[P13] Input 11	21	21	11	13, 11
[P13] Input 12	22	22	12	13, 12
[P13] Input 13	23	23	13	13, 13
[P13] Input 14	24	24	14	13, 14
[P13] Input 15	25	25	15	13, 15

Mach Input	HICON OEM J12 Input Pin	HiCON Integra J10 Input Pin	HiCON BASIC Macro GetPin() Index
[P14] Input 0	10 (J7, pin14)	8	14, 0
[P14] Input 1	11 (J7, pin15)	15	14, 1
[P14] Input 2	12 (J7, pin16)	7	14, 2
[P14] Input 3	13 (J7, pin21)	14	14, 3
[P14] Input 4	14 (J7, pin22)	6	14, 4
[P14] Input 5	15 (J6, pin23)	13	14, 5
[P14]Input 6	16 (J6, pin24)	9	14, 6
[P14] Input 7	17		14, 7
[P14] Input 8	18		14, 8
[P14] Input 9	19		14, 9
[P14] Input 10	20		14, 10
[P14] Input 11	21		14, 11
[P14] Input 12	22		14, 12
[P14] Input 13	23		14, 13
[P14] Input 14	24		14, 14
[P14] Input 15	25	*7737 Estop Status*	14, 15

© 2026 Vital Systems, Inc.

Mapping Mach Outputs to HICON Digital Output Pins

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

Mach Output	HICON OEM J8 Output Pin	HiCON Integra onboard Output	EtherCAT EC01 onboard Output	HiCON BASIC Macro SetPin() Index
[P11] Output 0	2	J15, OUT0	J6, OUT0	11, 0
[P11] Output 1	3	J15, OUT1	J6, OUT1	11, 1
[P11] Output 2	4	J15, OUT2	J6, OUT2	11, 2
[P11] Output 3	5	J15, OUT3	J6, OUT3	11, 3
[P11] Output 4	6	J15, OUT4	J6, OUT4	11, 4
[P11] Output 5	7	J15, OUT5	J6, OUT5	11, 5
[P11] Output 6	8	J15, OUT6	J6, OUT6	11, 6
[P11] Output 7	9	J15, OUT7	J6, OUT7	11, 7

Mach Output	HICON OEM J9 Output Pin	HiCON Integra J7 Output Pin	7535 breakout board output	HiCON BASIC Macro SetPin() Index
[P12] Output 0	2	2	0	12, 0
[P12] Output 1	3	3	1	12, 1
[P12] Output 2	4	4	2	12, 2
[P12] Output 3	5	5	3	12, 3
[P12] Output 4	6	6	4	12, 4
[P12] Output 5	7	7	5	12, 5
[P12] Output 6	8	8	6	12, 6
[P12] Output 7	9	9	7	12, 7

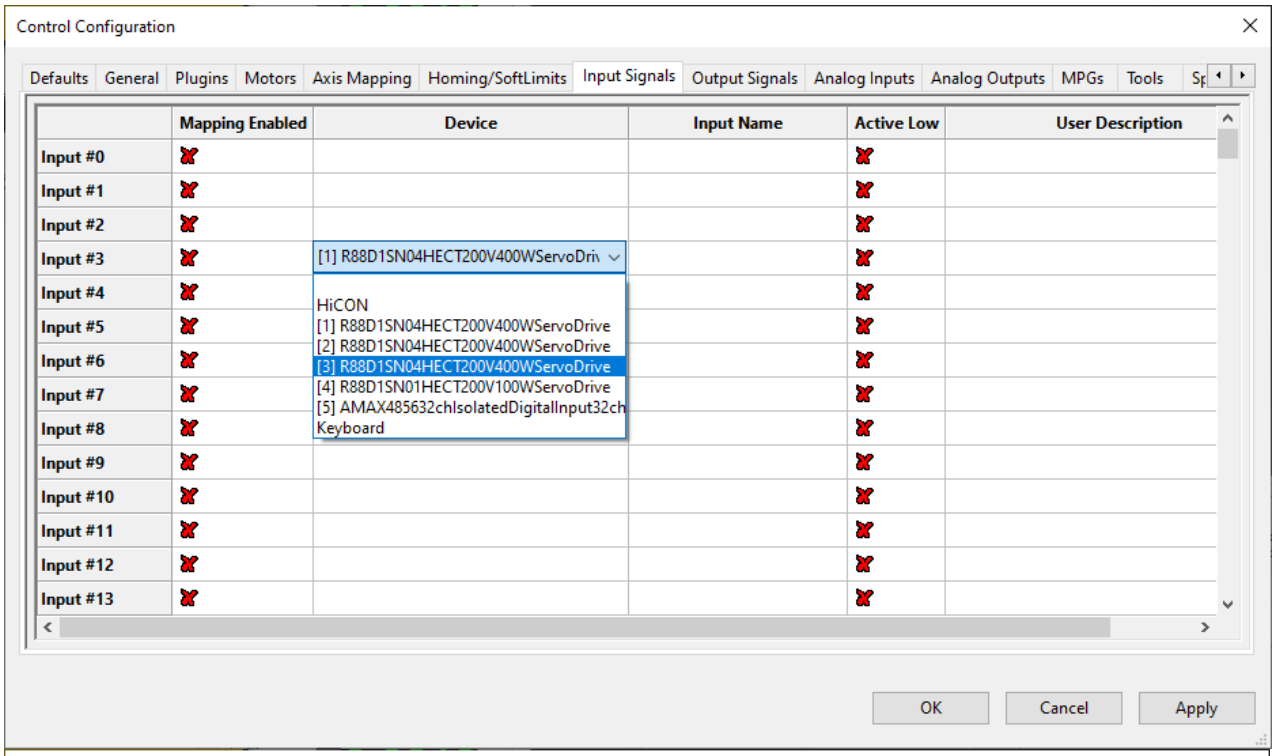
Mach Output	HICON OEM J11 Output Pin	HiCON Integra J8 Output Pin	7535 breakout board output	HiCON BASIC Macro SetPin() Index
[P13] Output 0	2	2	0	13, 0
[P13] Output 1	3	3	1	13, 1
[P13] Output 2	4	4	2	13, 2
[P13] Output 3	5	5	3	13, 3
[P13] Output 4	6	6	4	13, 4
[P13] Output 5	7	7	5	13, 5
[P13] Output 6	8	8	6	13, 6
[P13] Output 7	9	9	7	13, 7

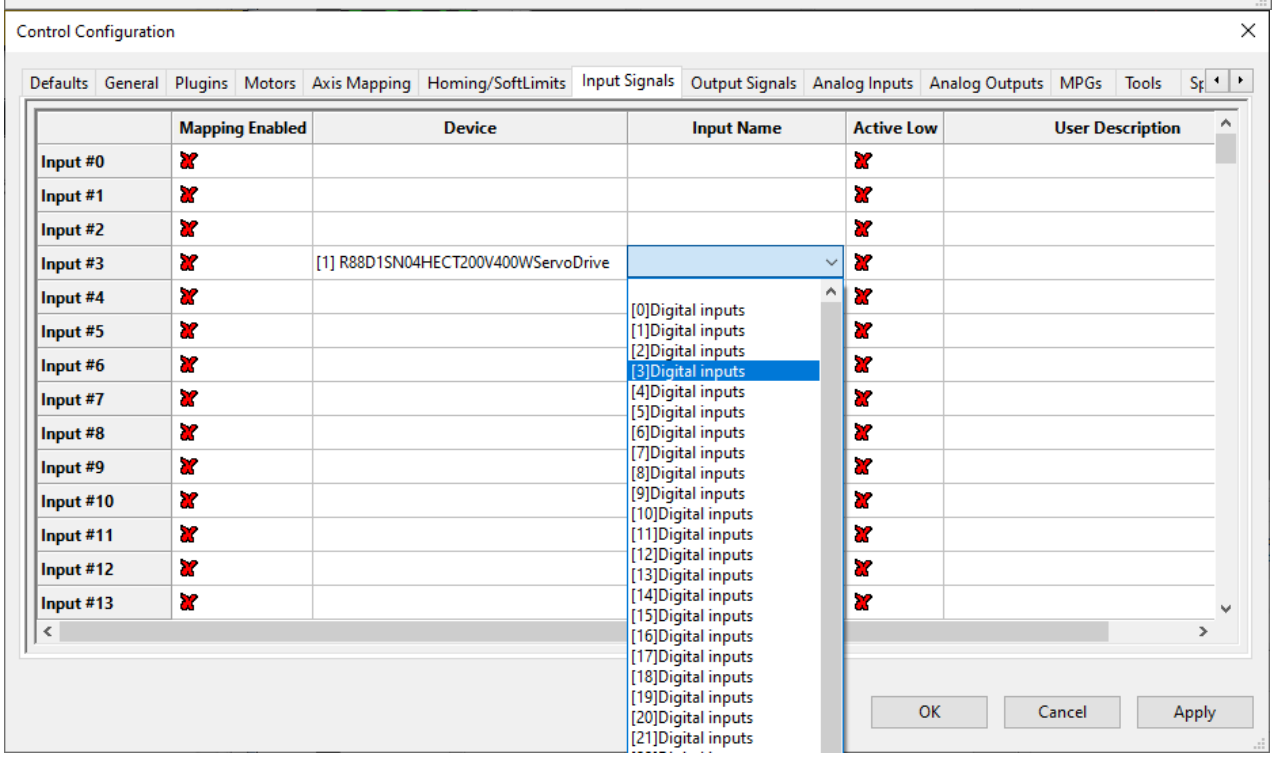
Mach Output	HICON OEM J12 Output Pin	HiCON Integra Relay Outputs	HiCON BASIC Macro SetPin() Index
[P14] Output 0	2		14, 0
[P14] Output 1	3		14, 1
[P14] Output 2	4		14, 2
[P14] Output 3	5		14, 3
[P14] Output 4	6		14, 4
[P14] Output 5	7	J12, RL1	14, 5
[P14] Output 6	8	J12, RL2	14, 6
[P14] Output 7	9	J12, RL3	14, 7

EtherCAT Digital I/O

To map EtherCAT I/O first make sure the relevant Digital I/O PDOs are selected and downloaded to the controller along with importing the ECL project file to the plugin.

Each EtherCAT device which has its I/O mapped will be available under the drop down as shown below.





Motor Parameters

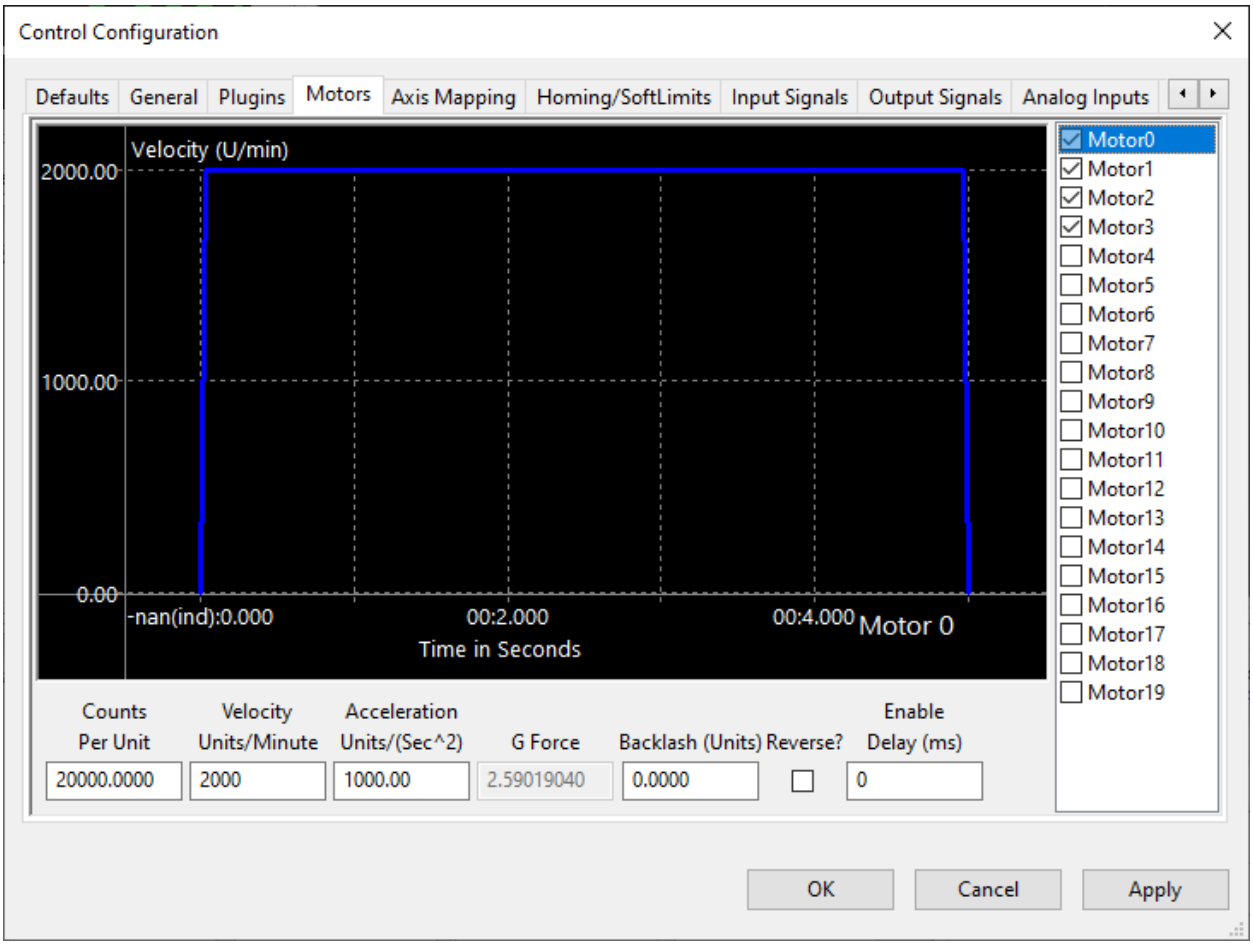
On the “Motors” tab, enable the motors that will be used by checking the checkbox of the corresponding motor to the upper-right of the window.

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

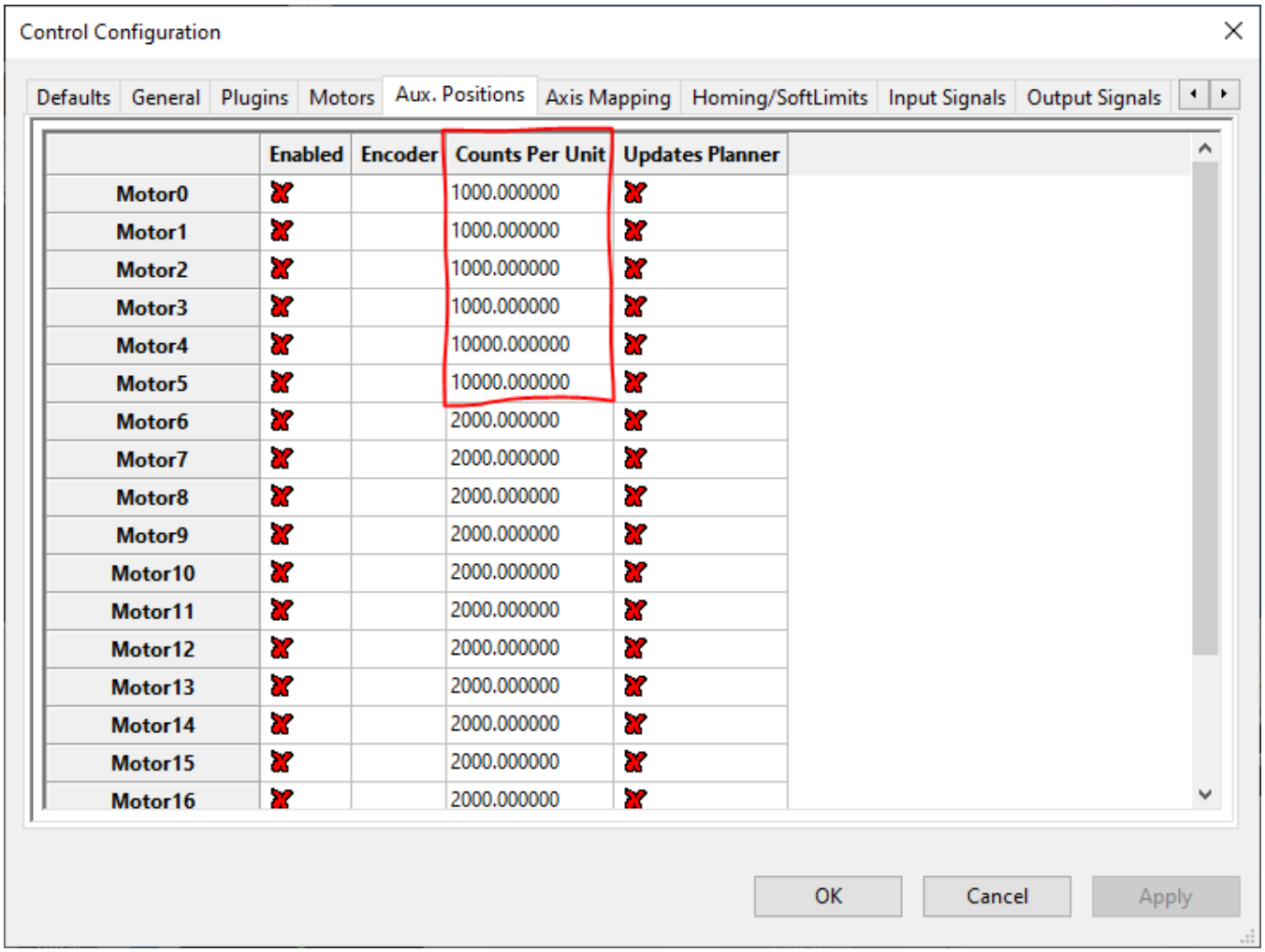
Because of the flexible nature of Mach, the number of usable motors is determined by the number of activated motors your controller has. EC01 can have up to 12 motors activated. You can check the number of motor activations under VSI device manager -> Activations tab. Additional motors can be purchased for your controller.

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

The HiCON will only arm a motor if it is enabled from this window (checked, as shown below).
Also activate only the number of motors you need.



NOTE In the latest Mach version you also need to key in your counts per unit in the Aux Positions. Does not matter if it is enabled or not.

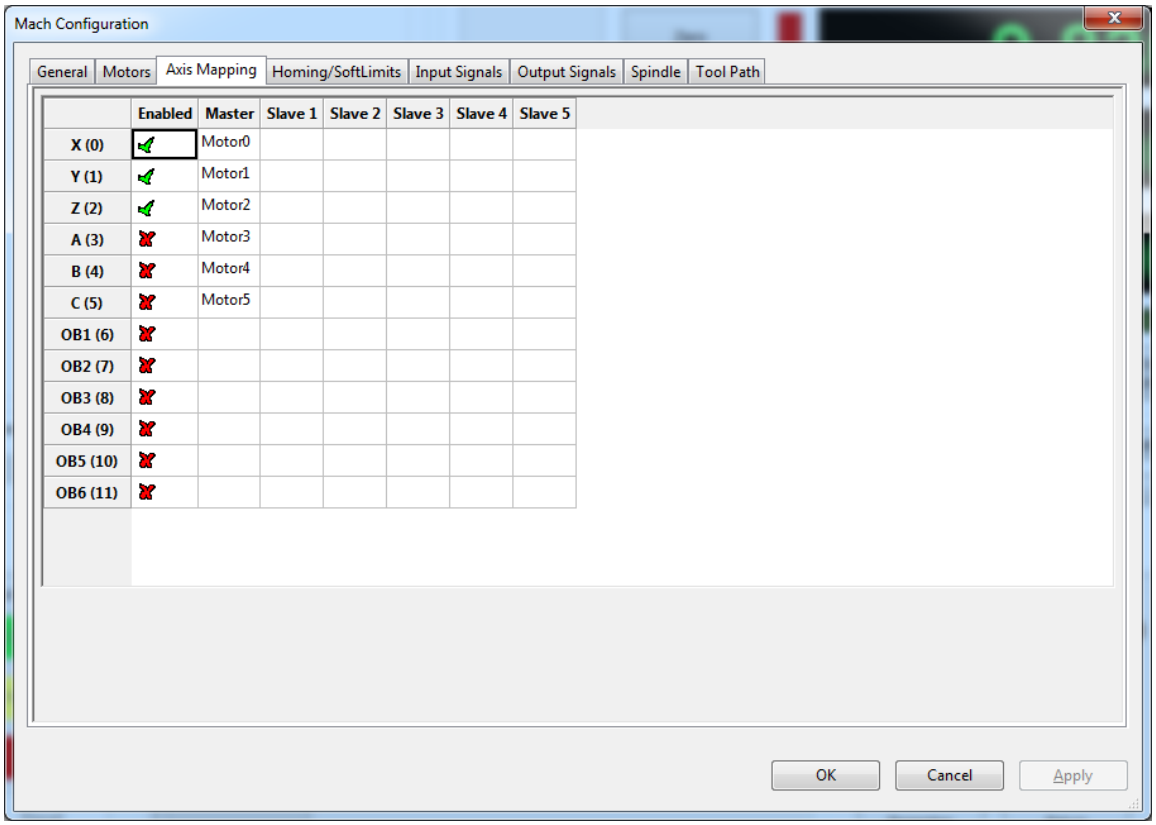


Mach Axis Mapping

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

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

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



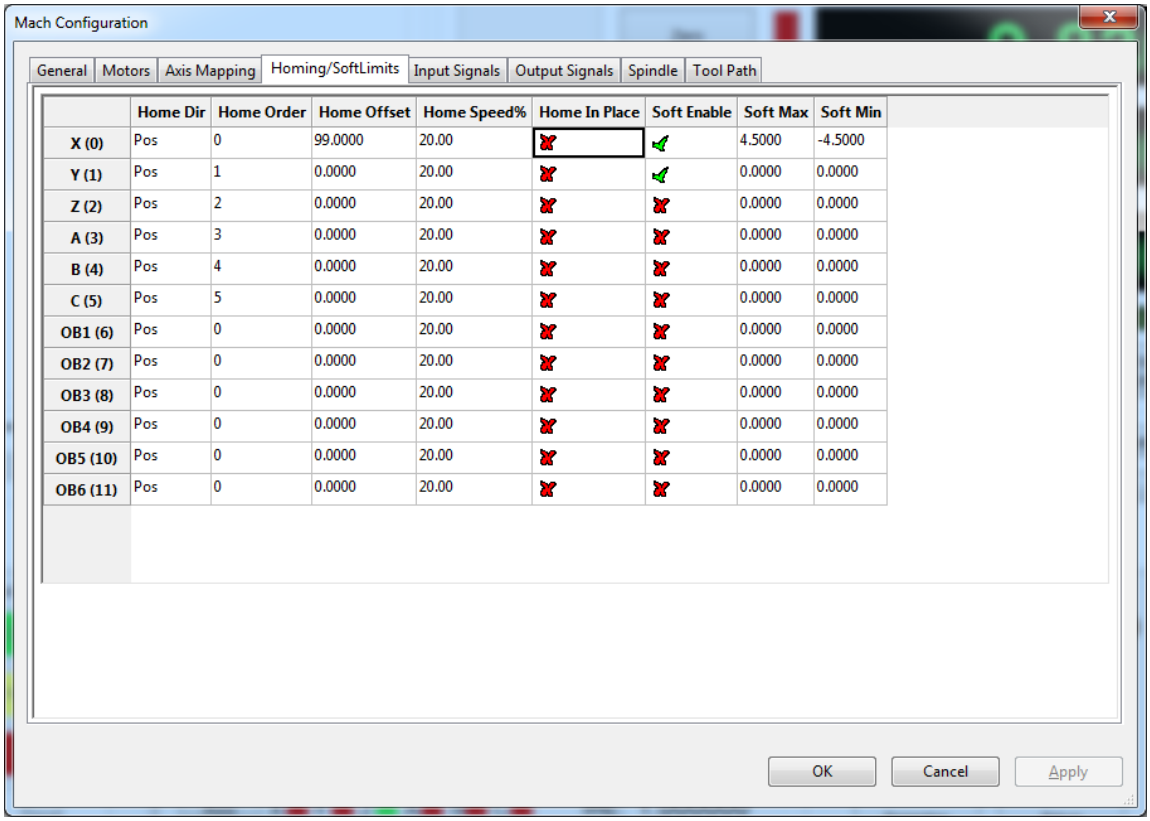
From this window, an axis can be quickly enabled/disabled, motors can be assigned as master or slave on any axis, and motors can be reassigned to operate on different axes without having to modify the motor configuration in the HiCON plugin.

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

Axis Homing Setup and Soft Limits

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

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

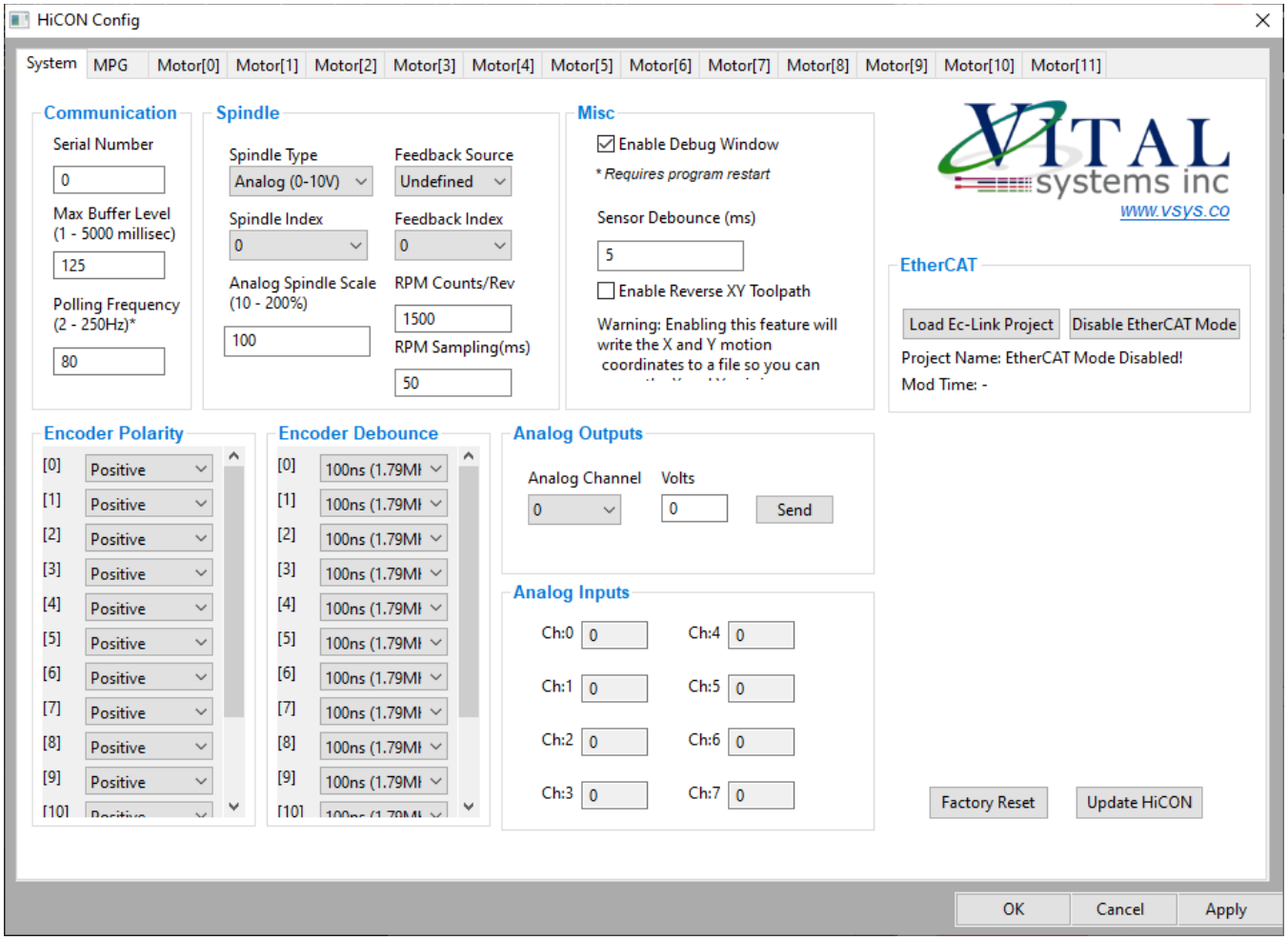


	Home Dir	Home Order	Home Offset	Home Speed%	Home In Place	Soft Enable	Soft Max	Soft Min
X (0)	Pos	0	99.0000	20.00	✗	✓	4.5000	-4.5000
Y (1)	Pos	1	0.0000	20.00	✗	✓	0.0000	0.0000
Z (2)	Pos	2	0.0000	20.00	✗	✗	0.0000	0.0000
A (3)	Pos	3	0.0000	20.00	✗	✗	0.0000	0.0000
B (4)	Pos	4	0.0000	20.00	✗	✗	0.0000	0.0000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

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

HICON System Configuration

NOTE: The first time this window is opened on each Mach Profile, you **must** click the 'Factory Reset' button to ensure that all settings are properly initialized in a clean state.



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

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

NOTE: "Factory Reset" will revert all the settings to default. Use this feature to start over.

Communication

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

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

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

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

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

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

Spindle Config

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

Undefined	Select this if a spindle is not used.
Analog (0-10V)	Most common for spindles driven by a VFD. Direction is determined by the Spindle Relay Outputs. *(Spindle Index should be set to 0)*
Analog (±10V)	For special VFDs or custom analog output that requires negative voltages. (Available in DSPMCv3)

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

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

- ***Spindle Index*** *– In Controllers that support multiple Analog Outputs you can choose which output the spindle is connected to. By default, it is 0 for controllers with one analog output.*

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

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

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

Misc. Config

- ***Enable Debug Window –*** Selecting this option will have Mach 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 Mach.

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

Hardware Encoder Polarity

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

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

Encoder Debounce

The ***Encoder Debounce*** field is used to filter noise from the hardware encoder signals. Higher debounce values filter more noise, but reduce the maximum frequency of the encoder signals. If this value is too low, it will cause the encoder readings to be inconsistent, resulting in following errors. The recommended value is 100ns, but higher values may be used if the reading is experiencing a lot of electrical noise.

Update HICON (System Tab)

This button downloads all system configuration parameters to the HICON.

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

Analog Outputs

Test the analog outputs of your controller.

- ***Analog Channel*** *– Select the Analog Channel to set the voltage of. By default, it is 0 for controllers with one analog output.*
- ***Volts*** – Enter the value in volts to send to the analog output. Usually the range is 0-10V but the DSPMC is capable of applying -10 to +10V.
- ***Send*** – Press the button to set the voltage of the analog output.

Analog Inputs

View the Analog Input values here.

EtherCAT

Here you will find the settings related to the EtherCAT motion controllers

- ***Load Ec-Link Project –*** Load the Ec-Link Project file that was created using the Ec-Link application so you can start configuring the Ethercat inputs and outputs. **Mach must be restarted after loading the project file every time.**
- ***Disable EtherCAT Mode –*** If you were using an EtherCAT motion controller like EC01 and want to switch to a non EtherCAT motion controller then you will have to disable the EtherCAT Mode and restart Mach in order to use it.

MPG Setup

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

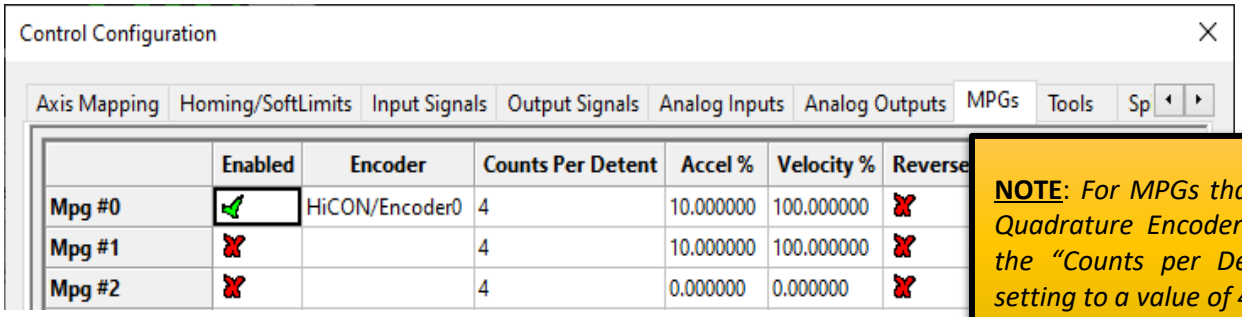
NOTE: *For USB based MPGs (e.g. Shuttle Pro), this section can be disregarded, but ensure the "Enable VSI MPG I/O" setting is unchecked.*

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

It is always recommended to first configure the MPG ESTOP button (if available) before attempting to perform axis jogging with an MPG Pendant.

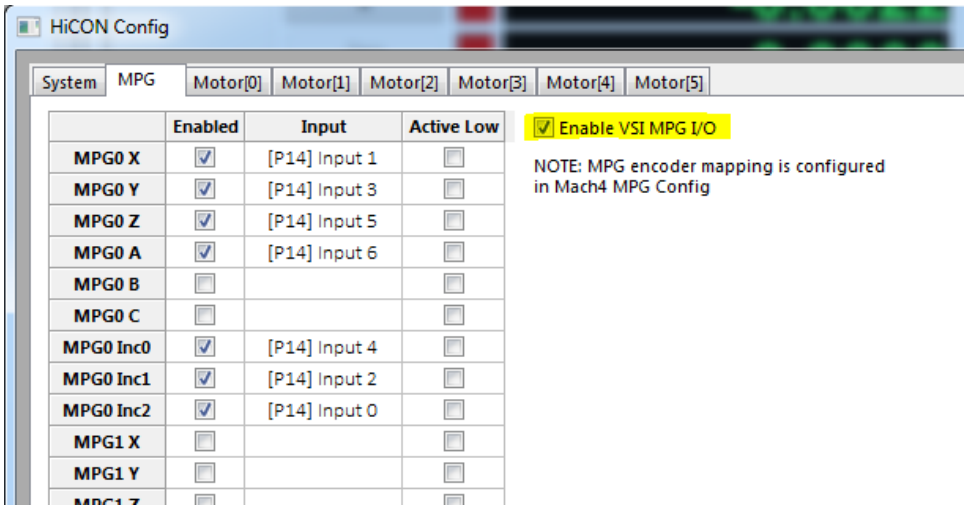
MPG Encoder Selection

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



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

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

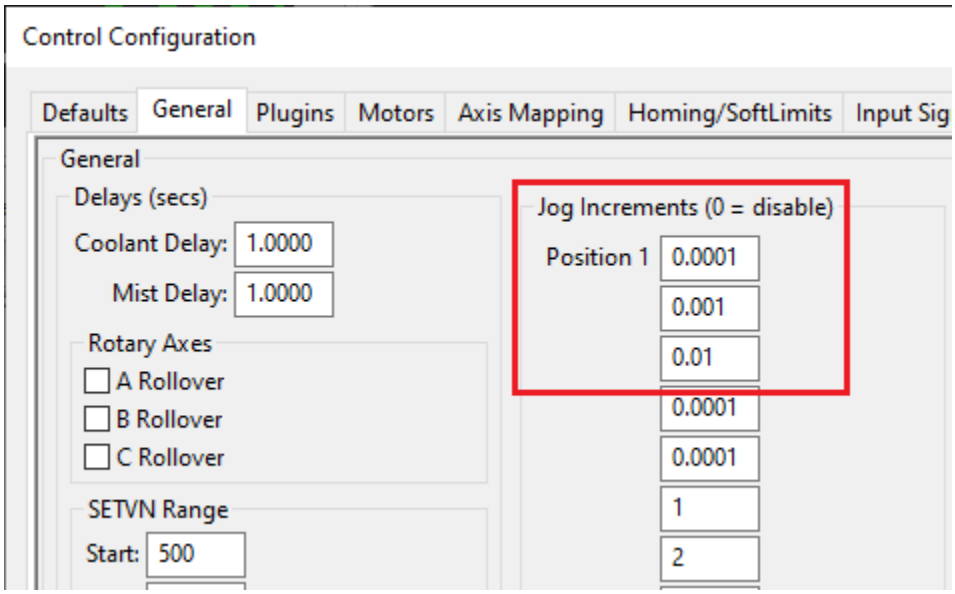


NOTE: *Digital I/O and Encoder mapping in the above MPG config is for the **(pn7766 & 7754) HiCON Integra** and the MPG06ED available from the Vital System Inc. website. For other Motion Controllers, select the appropriate input signals.*

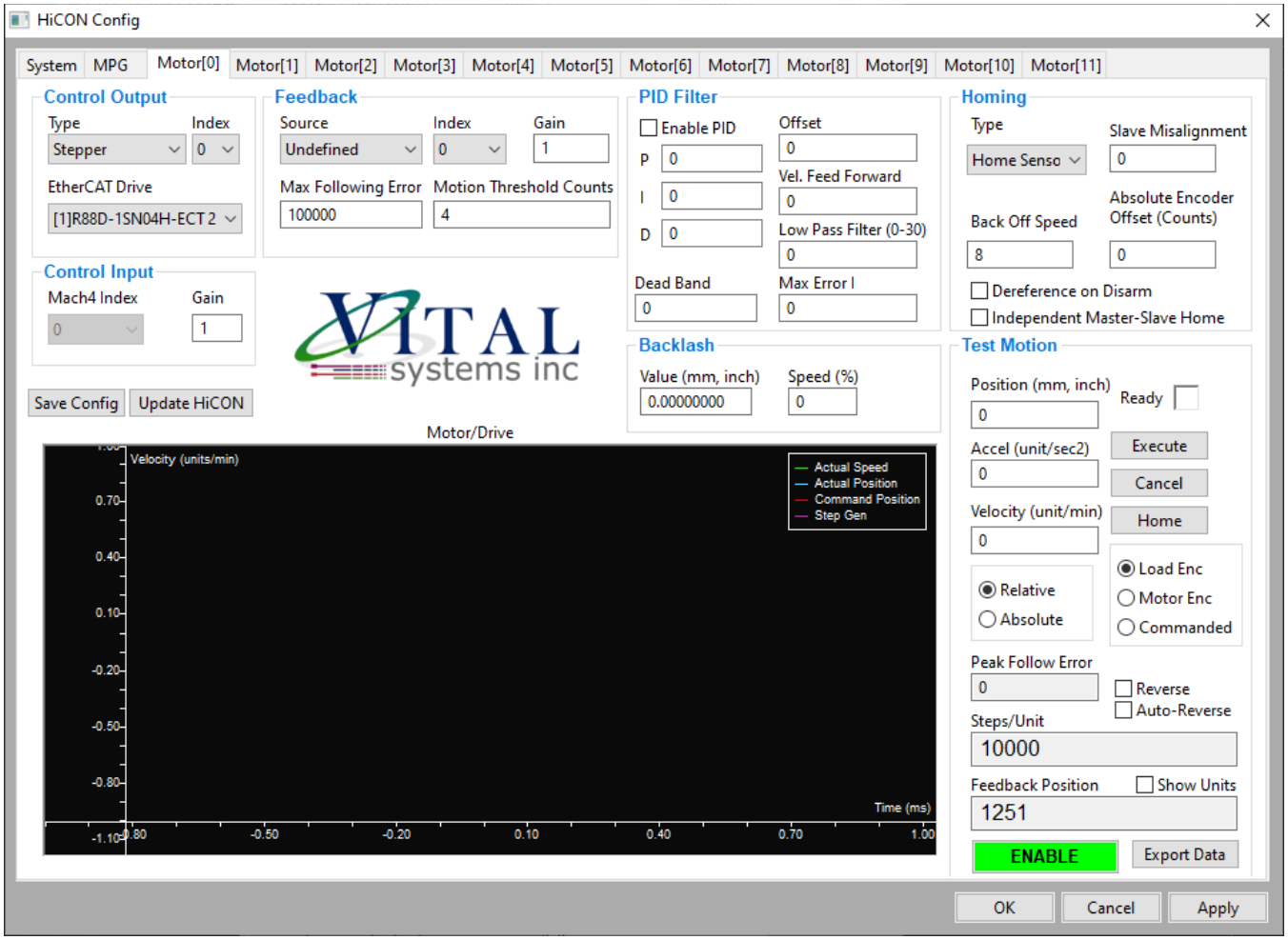
The “Enable VSI MPG I/O” checkbox enables or disables the HiCON plugin MPG I/O behavior (axis and increment selector buttons).

MPG Jog Increments

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



HICON Motor Configuration



The Motor tabs provide configuration settings that are directly related to each physical 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.

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 Mach such as jogging, MDI, or GCode Files.

Control Input Parameters

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

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

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

Control Output Parameters

- ***Control Output Type –*** This field determines the output signal type for the selected motor. The possible values are:
  - ***Stepper:*** Use one of the dedicated Step and Direction channels (0 – 5) as the output for position control (2 MHz max frequency). *This is the recommended setting*.
  - ***Analog:*** On supported boards like the DSPMCv3 use one of the analog outputs to control an analog motor.
  - ***EtherCAT:*** On supported boards like the EC01 you can set the output as EtherCAT.
  - ***Undefined:*** This setting causes the motor to be disabled to ignore the control output index.

- ***Control Output Index*** - The output channel number where the commanded motion will be issued.
- ***Control Output EtherCAT Drive –*** On supported boards like EC01 set the name of the EtherCAT drive that is connected to the controller. When this selection is enabled the Index represents the index of the motor within a single EtherCAT drive. The HiCON firmware supports up to 3 motors per drive (Index 0, 1, 2).

Feedback Parameters

- ***Feedback Source –*** This determines the source of the feedback position for the selected motor. The possible values are:
  - ***Undefined:*** No feedback source is selected. The outputted step pulses are used instead as the feedback source. In case of EtherCAT the CiA402 "Actual Position" is used.
  - ***Encoder:*** Use one of the Differential Quadrature hardware encoder channels (0 – 7) as the PID feedback (4MHz max frequency).

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

- ***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.
- ***StepGen:*** Use the internal hardware step pulses as feedback. You can use the step pulses generated for another axis as feedback in complex and advance machines.

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

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

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

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

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

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

- **Motion Threshold Counts** – This represents the maximum number of counts a motor can move before being detected as being "in motion". Too low values can result in the system never detecting the motor in "stopped" state. Some high-resolution feedback which is in tens of thousands can have a large idle count variation. Set this value to fall within the observed variation in the raw feedback count which can be seen in the HiCON status window when the motor is in idle state.

Backlash Compensation

- ***Backlash Value –*** The amount of position compensation to apply in response to motor backlash.
- ***Backlash Speed*** – The speed at which the backlash compensation is applied. This value is represented as a percent of the maximum motor velocity defined in the Mach Motors configuration.

> **NOTE**: *The performance of these values can be verified by performing test motion runs from the plugin motor config tabs.*

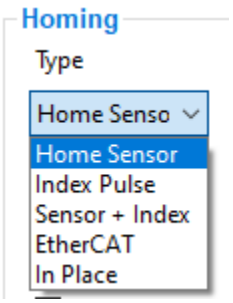
PID Filter

Follow the PID setup Guide to use the PID function. The PID function works on both Stepper and Analog drives. This feature required the Closed Loop Software activation and a hardware encoder feedback from the motors of linear scales.

Homing Parameters

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

- ***Homing Type*** – Dropdown that defines the homing sequence for each axis:
  - **In Place –** The current position of the axes will be set as their home position.
  - ***Home Sensor -*** *T*he motor is commanded to move in the configured homing direction (from Mach) until the motor's home sensor (mapped from the Mach inputs tab) is activated. It then moves in reverse at the specified ***Back Off Speed*** **(see below),** until the sensor is released.



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

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

- **Sensor + Index –** The same as the Home Sensor method, except it will continue to back off until it has found an Index Pulse, to guarantee the most consistent home position.
- **Ethercat –** The homing will be executed by the EtherCAT drive itself. This will require several config steps:
  - In ECLink project of your EtherCAT network, for each drive, add Mode of Operation to RxPDO, and Mode of Operation Display to TxPDO. Configure the homing SDOs starting at 0x6098, e.g., Homing Method, Velocity, etc. The actual function of the Homing Method SDO depends entirely on the drive. You will need to refer to the Drive's documentation from this point onward. Refer to Ethercat Homing section for more information.

- ***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 Mach Axis Mapping.*

- ***Back Off Speed*** – Sets the speed at which the motor will move away from the home sensor once it is triggered while referencing. The value is in units/minute.
- ***Dereference on Disarm*** – selects whether or not the axis reference flag will be reset when controller is disabled. When using encoder feedback this option should be left unchecked as the reference position is not lost when the machine is disabled.
- ***Absolute Encoder Offset*** – For use with EtherCAT Drives with absolute or battery backed incremental encoders. When this value is set to non-zero, the motor is understood to use an absolute encoder. The behavior of this parameter depends if the motor is a master or slave as configured in Mach Ax
  - ***Master Motor***: The value provided will be applied as an offset to the actual position of the drive. This is useful to set a reference position on the table.
  - ***Slave Motor:*** The offset is used to maintain a physical separation between the master and slave motor. This separation remain in place during the motion as well. Homing the axis will not be possible while this value is non-zero. This offset is important for having a squared gantry when using Absolute Encoders.

- ***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 Mach Axis Mapping. This option also requires that a slave motor have its own home sensor mapped in the Mach Input Signals and wired to one of the HiCON digital inputs.*

Test Motion

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

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

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

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

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

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

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

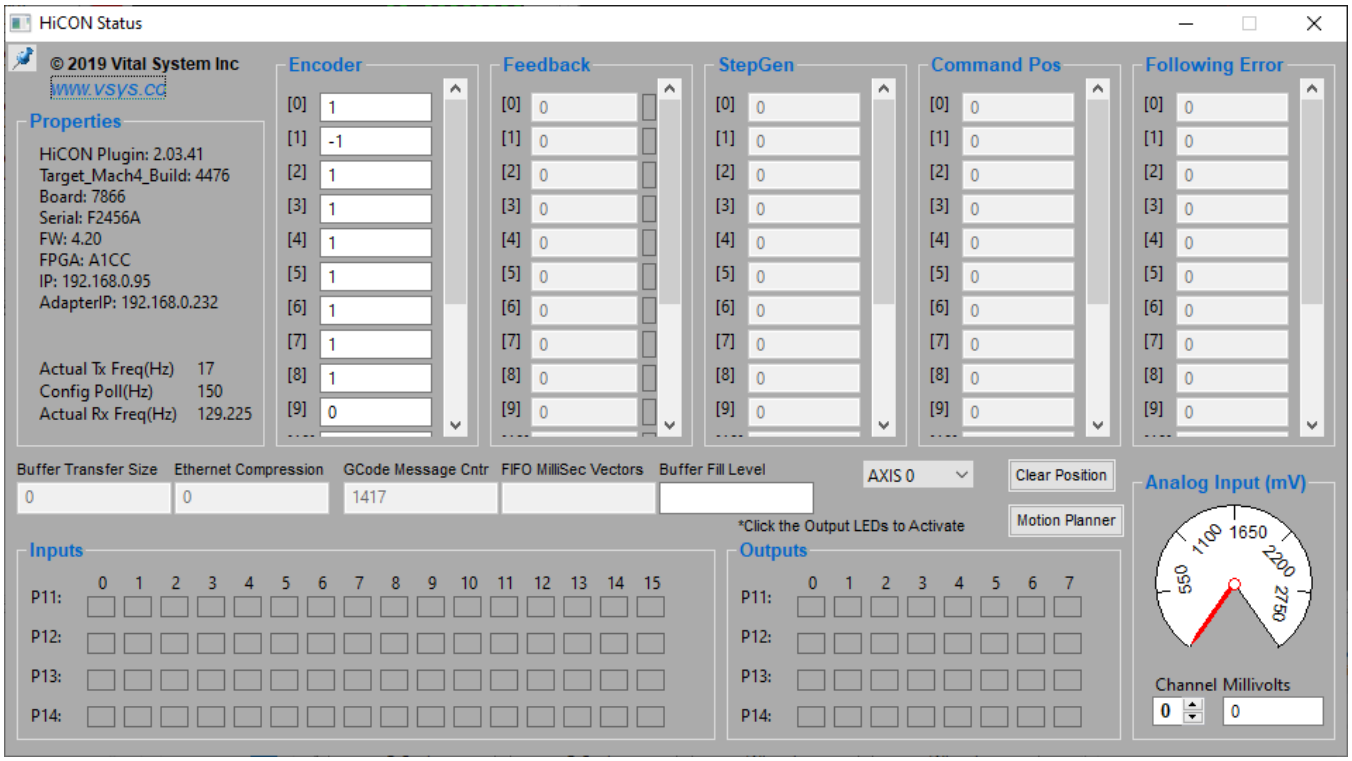
  NOTE: The system must be armed before performing motion.

- ***Position*** – Final/Target position or displacement in terms of Position Units, (e.g. 1.5, 10.093, mm or inches etc.)
- ***Acceleration*** – Acceleration value in terms of units per second squared, (e.g. inches/second^2, mm/sec^2 etc.)
- ***Velocity*** – Motion velocity value in terms of Units per minute, e.g. inches/minute, mm/minute etc.
- ***Relative/Absolute*** – These check boxes indicate whether the value in the ***Position*** field is either the distance to travel (relative) or the final position (absolute).

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

HICON Status Window

The HICON Status Window can be accessed from the Mach main menu (top of the main window), click on *"Diagnostic"*, then *"HICON…"* from the drop-down menu. The following window will be displayed.



This window contains information about the current status of the HICON such as:

- ***Properties*** – Displays the device information of the HICON controller.
- ***Encoder*** – Displays the current encoder position. These fields are helpful for indicating if the encoders are properly connected, and are reading correctly.

> **NOTE**: *The encoder value displays can be clicked to manually clear the encoder position. It is recommended to disarm the system first, otherwise a following error will be detected.*

- ***Feedback*** – Displays the current axis feedback. The value of this field depends on the currently selected feedback for the given motor (e.g. Encoder counts in closed loop, commanded step pulses in open loop). A red indicator is also present to the right of the value display which indicates if the motor
- ***StepGen*** – Displays the counter for outputted step pulses. When using encoder feedback, this value should ideally be equal to the "Encoder" counts, although current standards dictates a maximum difference of 4 counts between the Stepper and Encoder counts is within normal operating range.

- ***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 HICON immediately triggers an Emergency Stop which disarms the whole system.
- ***Analog Inputs*** – Displays the voltage reading (in millivolts) on an analog input.

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

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

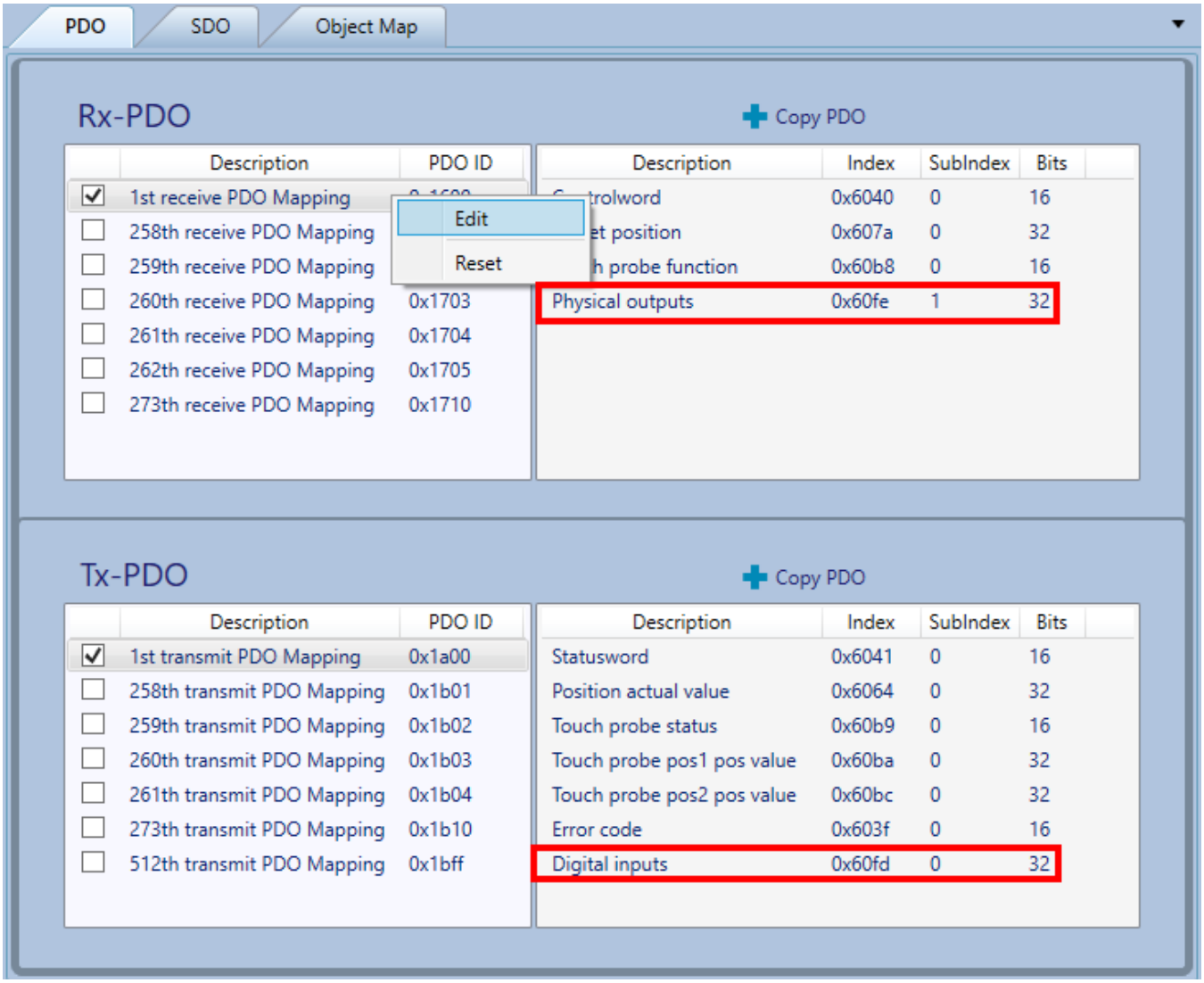
NOTE: *Digital Output states can be manually toggled by clicking on the displayed LED.*

EtherCAT Configurations

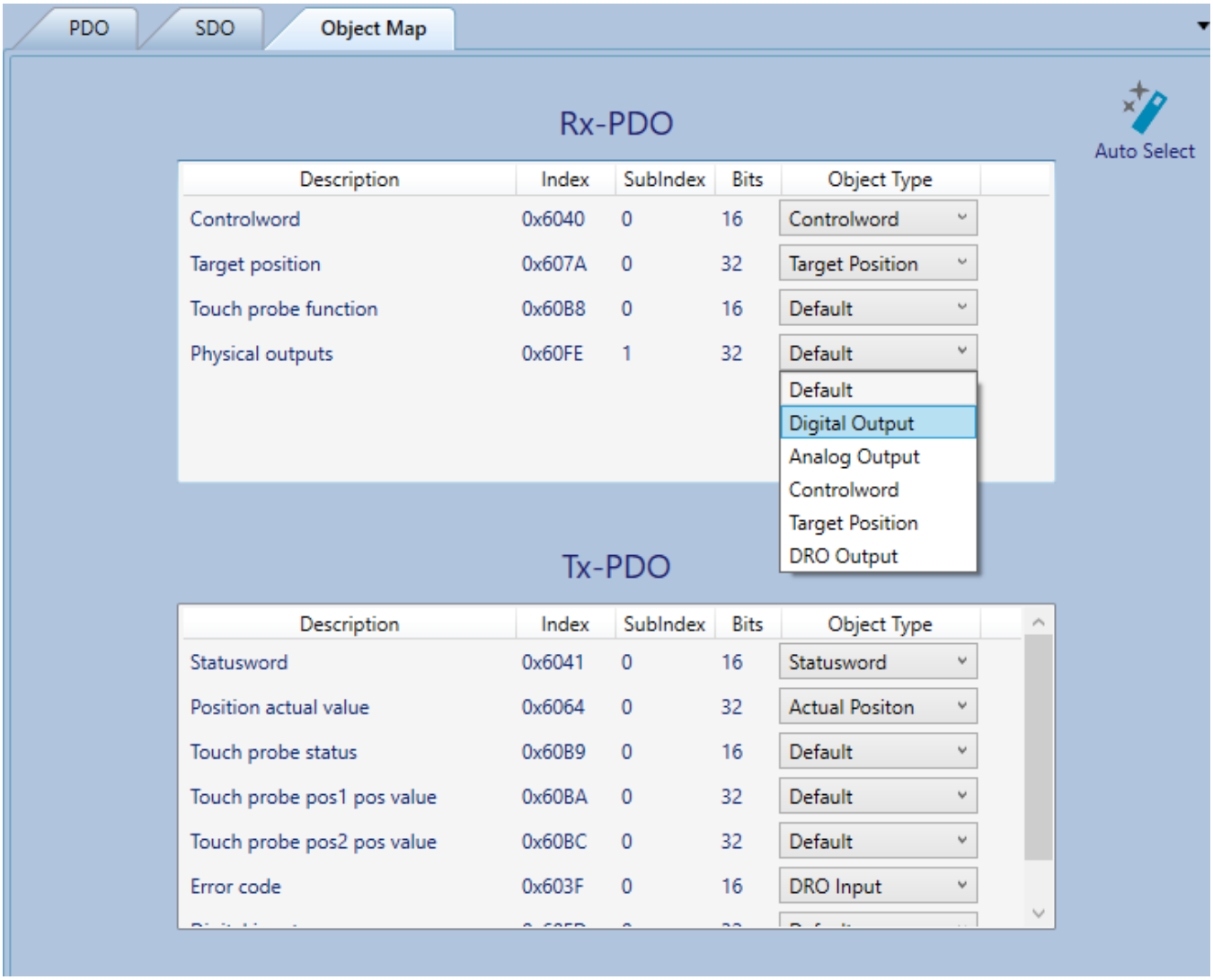
Mapping Additional EtherCAT I/O with Mach:

In ECLink you will need to do a few steps:

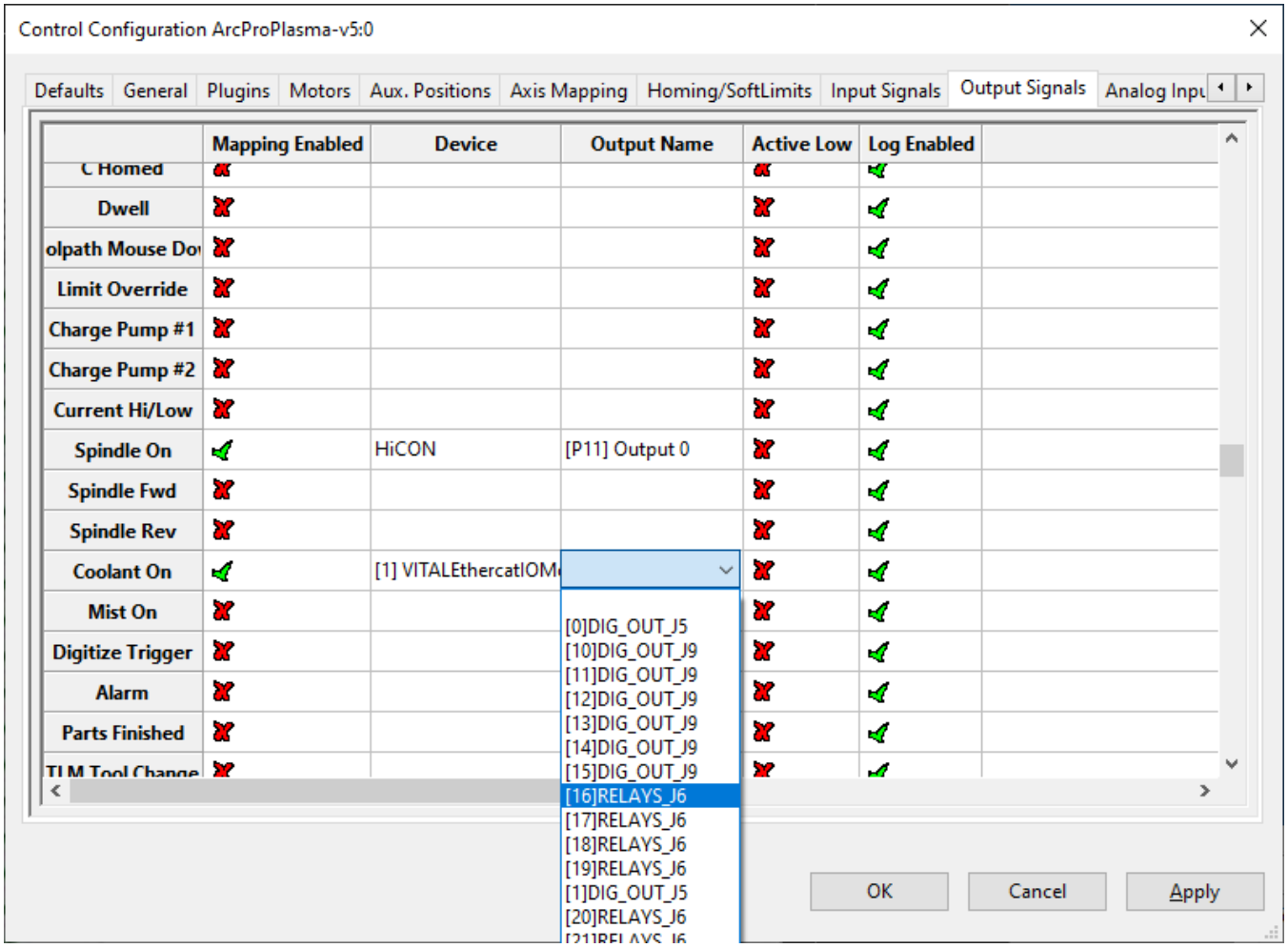
1. Add whatever IO you want to the PDO. Usually the IO will be included in one of the default PDOs. If it is not, you can right click a PDO and select Edit.



2. In Object Map tab, click on Auto Select. If it does not auto assign object type to the I/O, select Digital Input or Digital Output options:



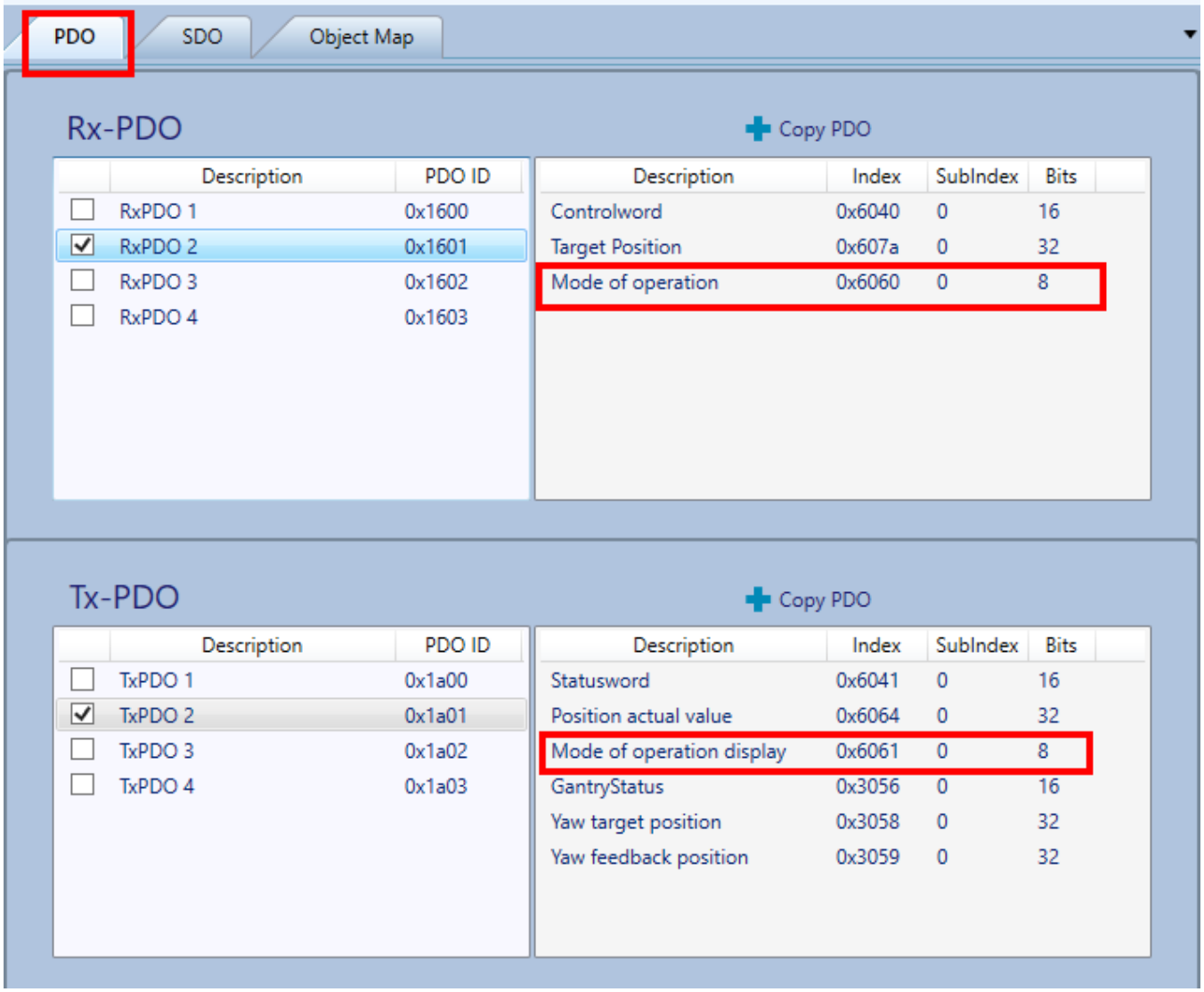
3. Download the Project to your EC01, and close Mach if it's open. Upon reopening Mach, the IO will be mappable in the Input and Output Signals tab of Mach



Ethercat Homing (Methods Built-in in EtherCAT Drive)

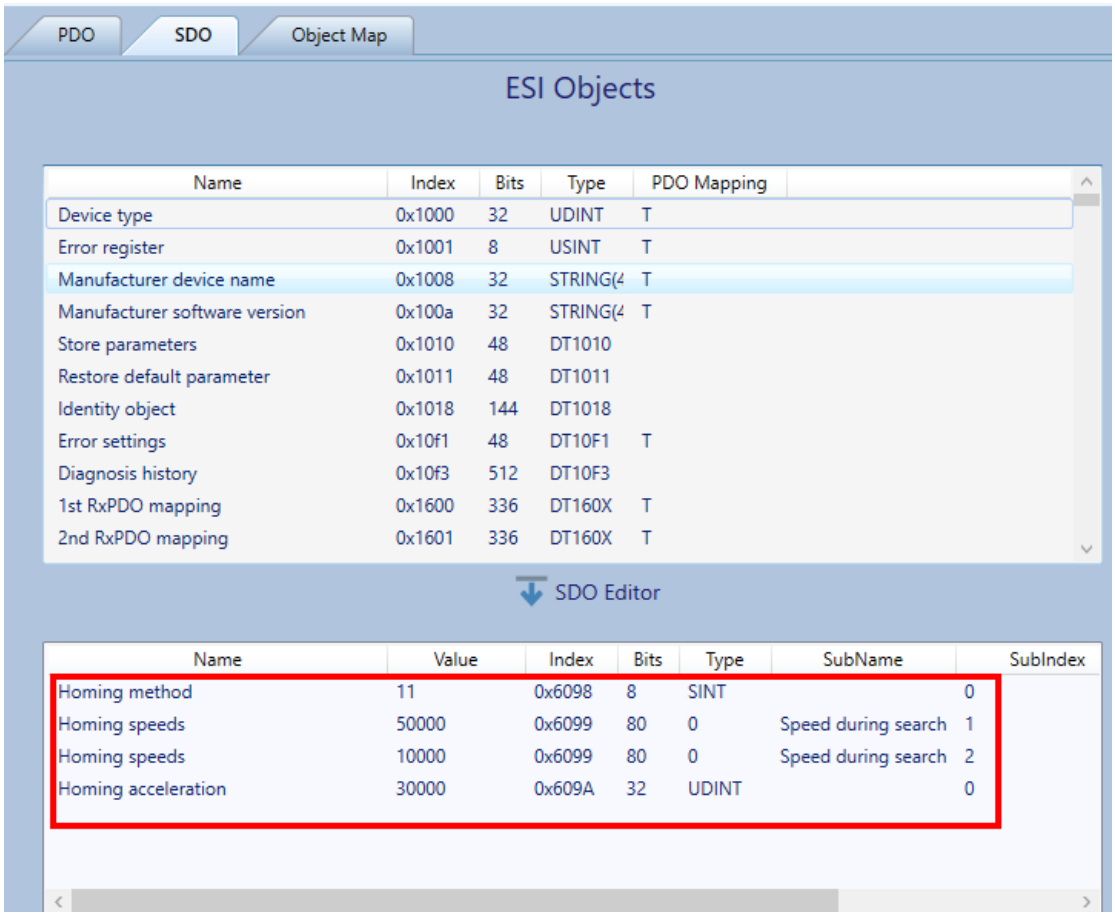
Enabling the use of Homing method directly on the EtherCAT drive can lead to more precise results when homing. You will need to perform the following steps for each motor/drive that you wish to enable this functionality for:

- In ECLink, add Mode of Operation to your selected RxPDO, and Mode of Operation Display to the selected TxPDO.



- Configure the homing SDOs starting at 0x6098:
    - Homing Method – This determines the method used when doing EtherCAT Homing. This is dependent on the Drive itself. You will need to refer to your Drive's manual in order to determine what value to select for this SDO.
    - Homing Speeds – This SDO contains two sub-values that must both be set.
        - Subindex 1 - Speed during search for Switch
        - Subindex 2 - Speed during search for Zero (Backoff)

      You can start with the default values for these SDOs as indicated in your Drive's manual, and adjust from there. If you do not add the SDO, then the drive will use the default value.
    - Homing Acceleration – Acceleration to use while performing Homing operations on the drive. You can start with the default value specified in your Drive's manual and adjust from there. If you do not add this SDO, then the drive will use the default value.



- Select EtherCAT Homing as your homing type in the Motor tab of the HiCON Plugin window.

Ethercat Status Window

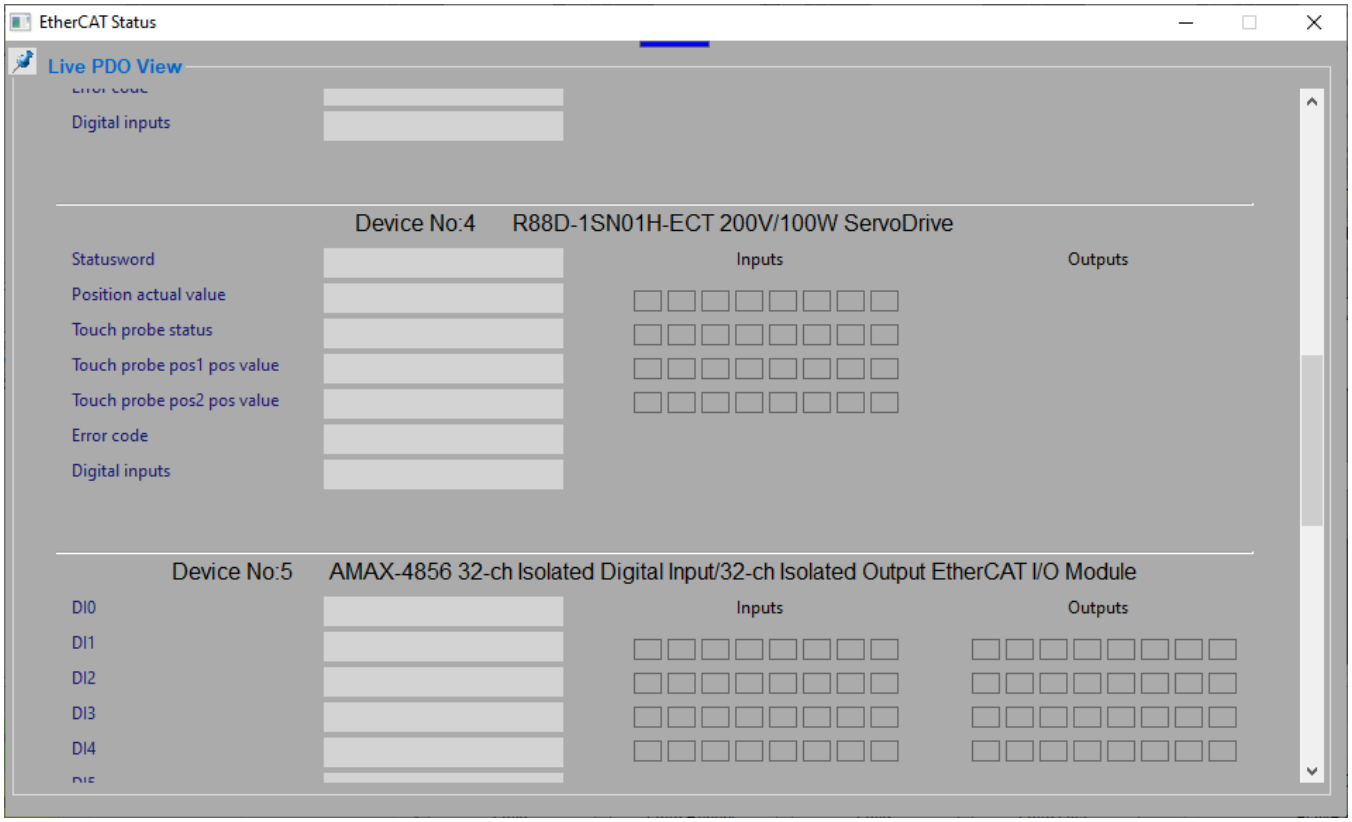
To open this window, you have to open the Status window then click on EtherCAT Status button located on the right side. This window is independent of the HiCON status and is only enabled with an EtherCAT motion Controller. Please follow the EC01 guide for further details on how to set up EtherCAT.

The EtherCAT status window is dynamic and is generated based on the EC-Link project configuration. Below is an example of how it will generate the PDOs

The LED on the top of the window flickers when it is receiving EtherCAT status updates from the controller.

EtherCAT outputs can be manually operated from this window.

You can click on the Thumb Tack to keep this window open on top of all other windows.



Additional Setup

Master-Slave Axis Setup

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

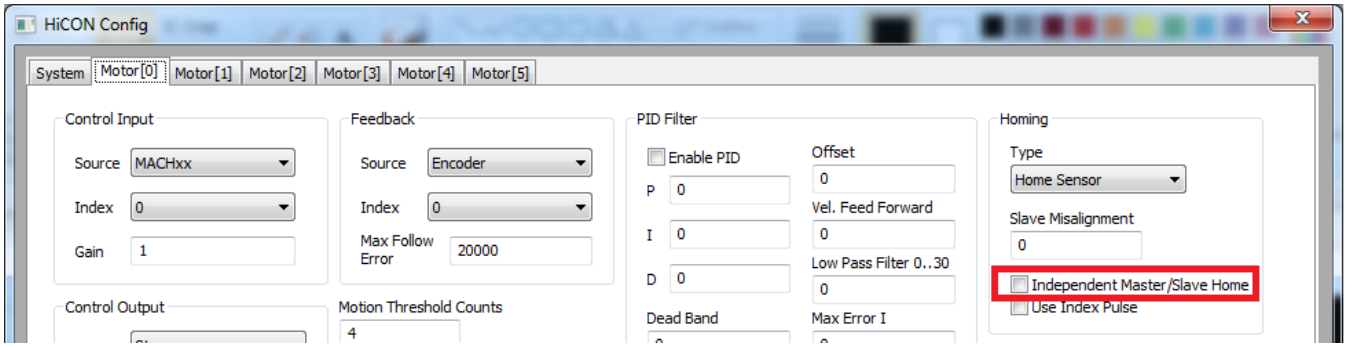
	Enabled	Master	Slave 1	Slave 2	Slave 3	Slave 4	Slave 5
X (0)	✓	Motor0	Motor1				
Y (1)	✗						
Z (2)	✓	Motor2					

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

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

Dependent Homing

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



Independent Homing

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

The following picture shows motor home sensors mapping:

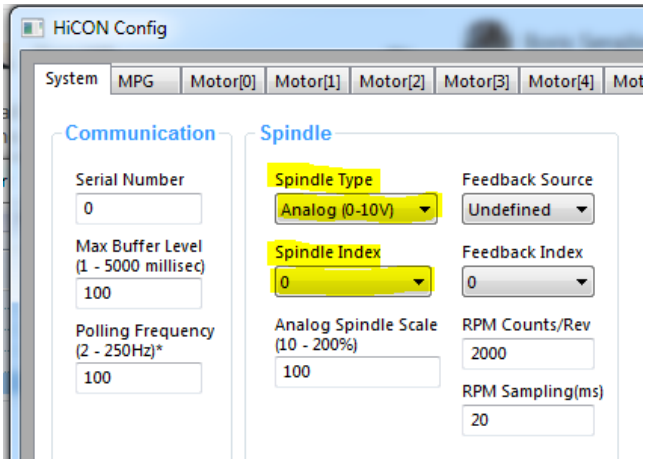
Motor 0 Home	✓	HiCON	[P11] Input 13	✓
Motor 1 Home	✓	HiCON	[P11] Input 14	✗
Motor 2 Home	✓	HiCON	[P11] Input 15	✗

Spindle Setup

Refresher:
You can set the spindle speed with the 'S' Command in your GCode, and enable the Spindle with the M3 Command. The S command sets the spindle speed in revolutions per minute (RPM). An example of a valid S command: S10000

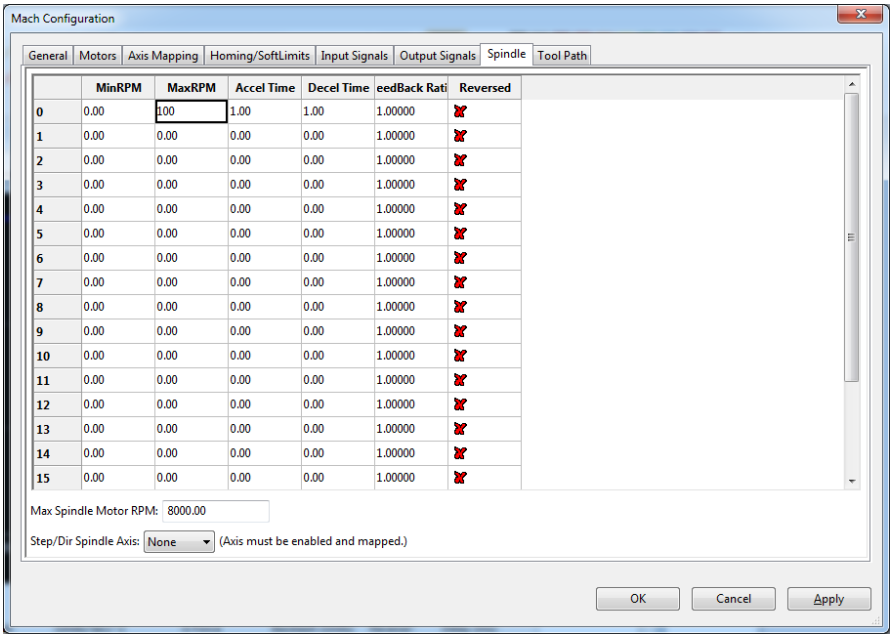
Analog Voltage Spindle

When using a VFD or any other motor-controlling device that uses 0–10V (such as a servo amplifier), the following steps are needed:



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

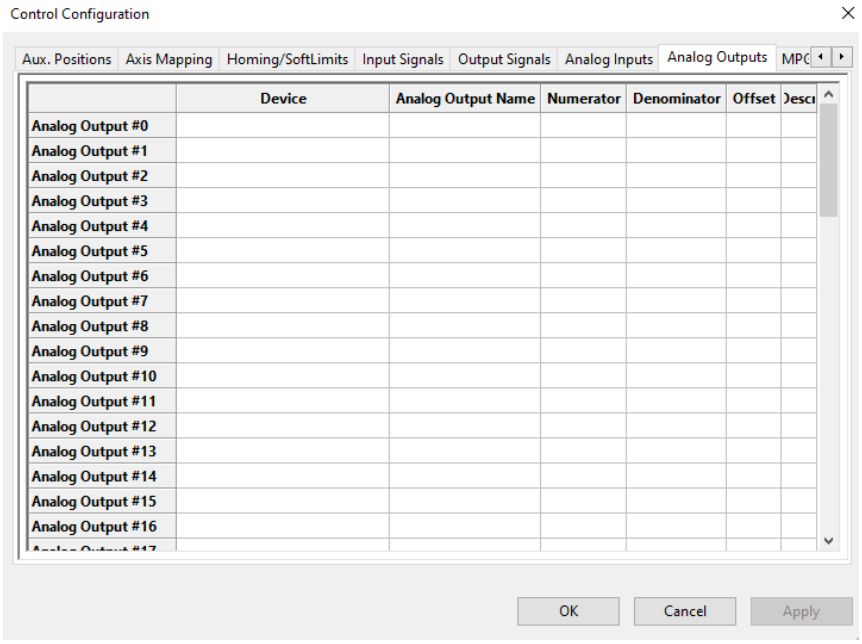
2. In the Mach Config window, go to the "***Spindle***" Tab. Set "***MinRPM***" to 0 and "***MaxRPM***" to the desired maximum RPM of the spindle. This will output a value of 0V to the spindle at the min RPM and 10V at the max RPM. Remember that the S command in GCode selects the speed of the spindle in RPM.



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

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

3. Make sure that the analog output for the spindle is **NOT** mapped in the Analog Outputs tab. If it is mapped in this tab, it will not be able to function properly as a spindle.

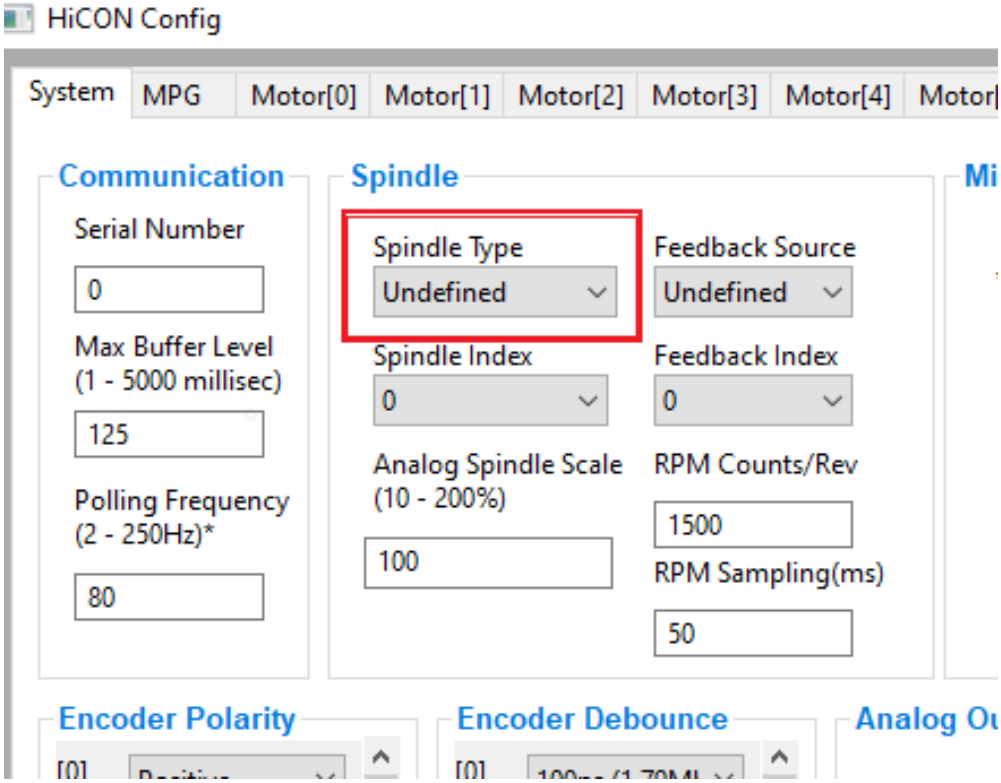


NOTE: *Do **NOT** map the spindle in the Analog Outputs tab. This is not only unnecessary, but will also prevent the spindle from functioning.*

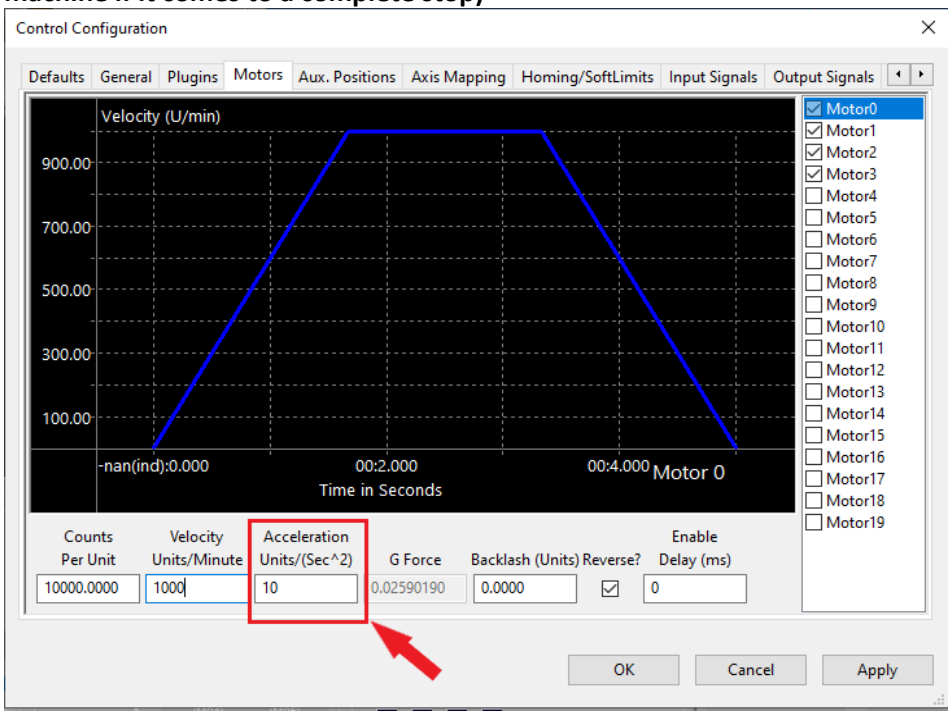
Setup Step/Dir or EtherCAT Motor as a spindle

Instead of using 0-10V analog signal to a VFD, one of the Step/Dir motors can be setup as a spindle. This is achieved using OB axis feature of Mach.
Setup the Motor as you would do it normally for an axis. E.g. in Motor[] tabs set the output type to Stepper/Analog/Ethercat and set its velocity / counts per unit/ acceleration in “Mach Control”

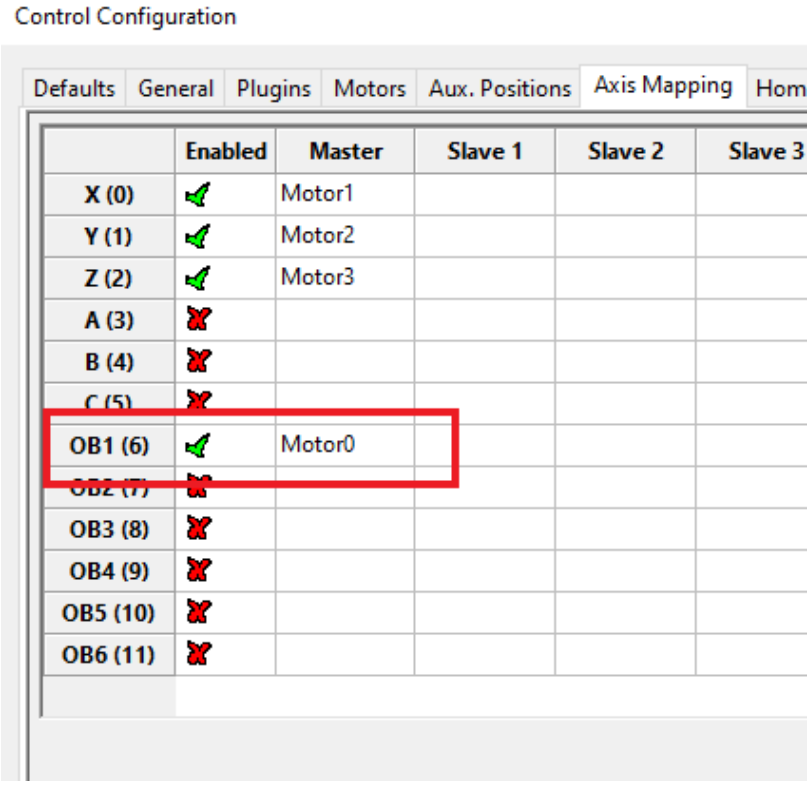
1. Set the Spindle Type in System Tab of HiCON Config to Undefined



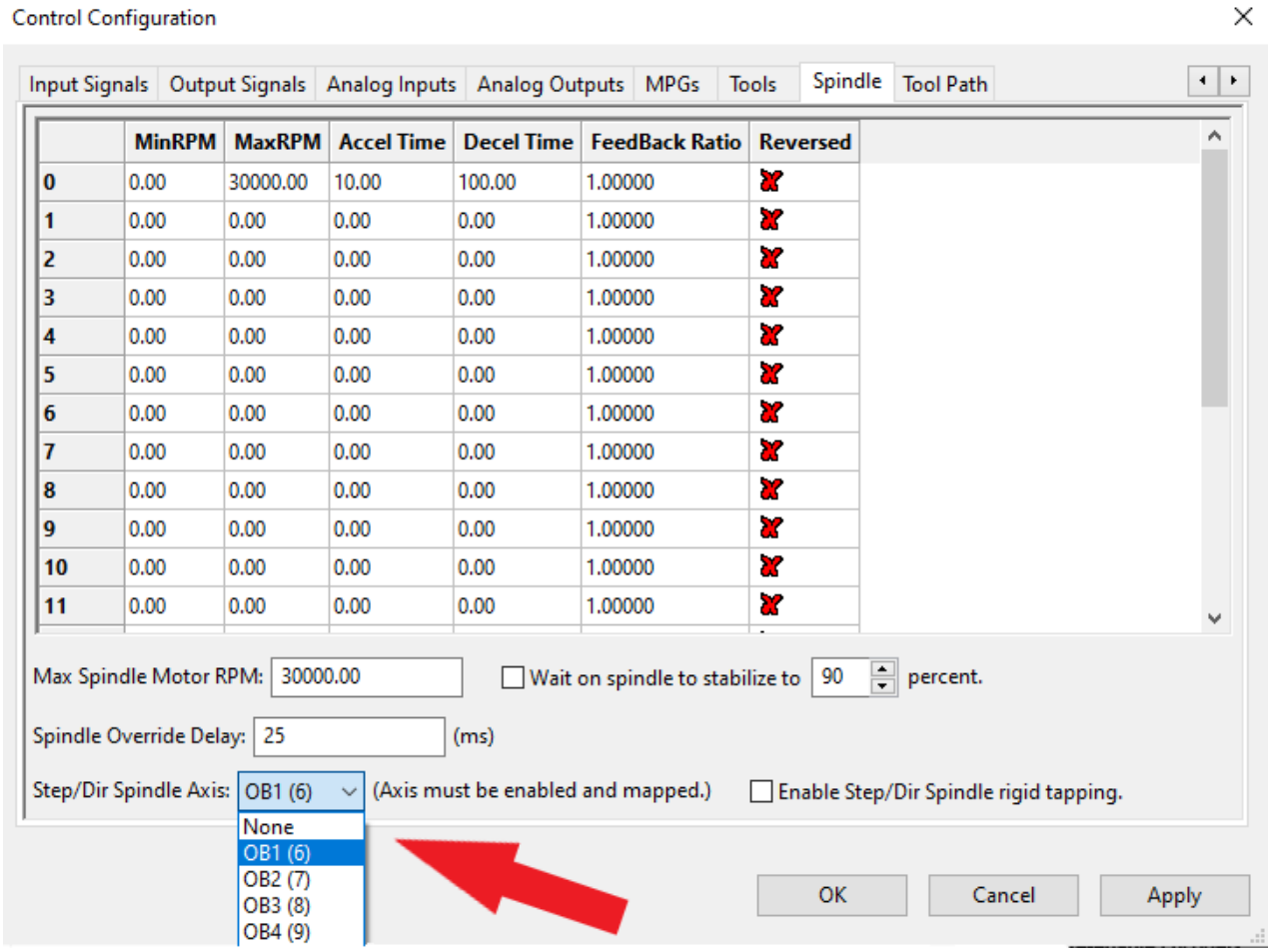
2. Make sure the acceleration of the motor spindle is set to a very low value. **(Setting it to a high value will cause motor to seize immediately when changing speeds and may cause permanent damage to the machine if it comes to a complete stop)**



3. Map the motor spindle to OB1(6) and enable it.



4. Under the Spindle Tab Select Step/Dir Spindle Axis to OB1 (6) as shown below.



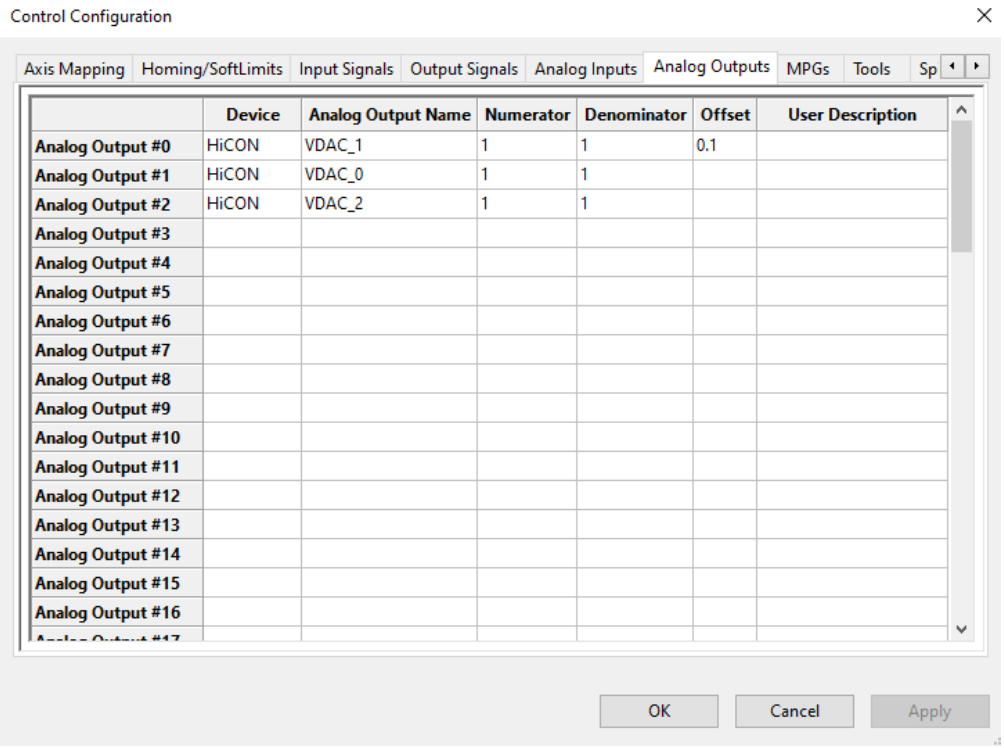
5. Setup MinRPM and MaxRPM as you would do for a normal spindle.

6. Now you can use M3 and S Gcode in Mach just as you would for a normal spindle.
   Note: Start off with low S values when using for the first time.

Synced Analog Outputs - M67

Synced analog outputs using the M-Code M67 is available for use with HiCON-EXTIO (Extended I/O) activation. This feature comes in handy if you want o synchronize analog output with your Gcode Eg. Laser engraving application. Depending on the motion controller you have up to 8 analog outputs available for use.

Map the analog registers in Mach Control and Analog Outputs tab as shown below.



	Device	Analog Output Name	Numerator	Denominator	Offset	User Description
Analog Output #0	HiCON	VDAC_1	1	1	0.1	
Analog Output #1	HiCON	VDAC_0	1	1		
Analog Output #2	HiCON	VDAC_2	1	1		
Analog Output #3						
Analog Output #4						
Analog Output #5						
Analog Output #6						
Analog Output #7						
Analog Output #8						
Analog Output #9						
Analog Output #10						
Analog Output #11						
Analog Output #12						
Analog Output #13						
Analog Output #14						
Analog Output #15						
Analog Output #16						

The Numerator and Denominator must be 1 by default but these are configurable. Offset is optional.

M67 E- Q-
E is analog output number in Mach control Q is voltage (float datatype) – refer the controller documentation for analog range.

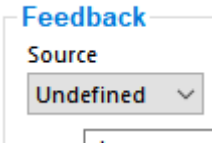
Here is an example Gcode to use 3 spindles simultaneously.
analog output #0 is @5.5V
analog output #1 is @6.7V
analog output #2 is @-4V
All three voltages are applied simultaneously at g0 x10

g0 x0
m67 e0 q5.5
m67 e1 q6.7
m67 e2 q-4
g0 x10

Counts Per Unit Calibration

You can use the HiCON plugins to get an accurate counts/unit without having to measure the teeth, belt or gears on the axis.

For each one of the motors in the HiCON plugin config axis tabs, disable the encoder feedback. Do not change the feedback index number, just set the source to 'Undefined' in the Feedback area. **Set Backlash value to zero and backlash speed to zero.**



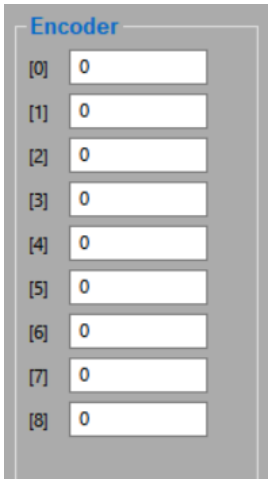
NOTE: *If your system does not have encoder feedback, use the StepGen counter instead of the encoder in the HiCon Status window.*

Initial Steps

- Mark where the machine's axis is on the table.
- Using the MDI tab for custom gCode, command the machine to move the x-axis a short measurable distance, like 10 inches. (g0 x10)
- Then measure the actual distance that the axis moved, and using the ratio method, update counts per unit in Mach motor tab to make sure that the commanded movement is close to the actual.
- As long as the commanded motion is relatively close to the actual distance travelled then you can continue to calibrate the counts per unit.

Counts per Unit

- Move the axis all the way to one end of the table
- Mark where the machine's x-axis is on the table
- In the HiCon Diagnostic/Status window for the Encoder's, click on the encoder channel field for the desired axis to clear the value.
- Using the MDI tab for custom gCode, command the machine to move the axis a long measurable distance, like 60 inches. (g0 x60)
- Then measure the actual distance that the axis moved on the table.
- Take the encoder value that is displayed in the HiCon status window for the selected axis and divide that by the actual distance travelled. This is your counts per unit for the axis. Enter this value in mach config motors tab.
- Repeat the above steps for each axis.



Re-enable Encoders

For each one of the motors in the HiCon config, reenable the encoder feedback. Do not change the index just set the source to 'encoder' in the Feedback area.

If your system does not have encoder feedback, use the StepGen counter instead of the encoder in the HiCon Status window.

Scripting and Mach HICON Registers

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

This section can be skipped if your application does not require any custom functionality through lua scripting, or any screen editing.

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

Encoder (0 – 8) – These ***read-only*** registers display the current encoder counter position for each encoder channel.

VADC (0 – 1) – These ***read-only*** registers display the voltage readings (0 – 3300 milliVolts) for each analog input channel.

VDRO (2000 – 2049) – These ***writable*** registers are general-purpose numerical values that are sent to the HICON 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 HICON 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 HICON and used in the HICON 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 HICON and used in the HICON BASIC Macro Program *(i.e. DROs 2050 – 2099).*

Accessing register files from a lua script may be accomplished with the lua command:

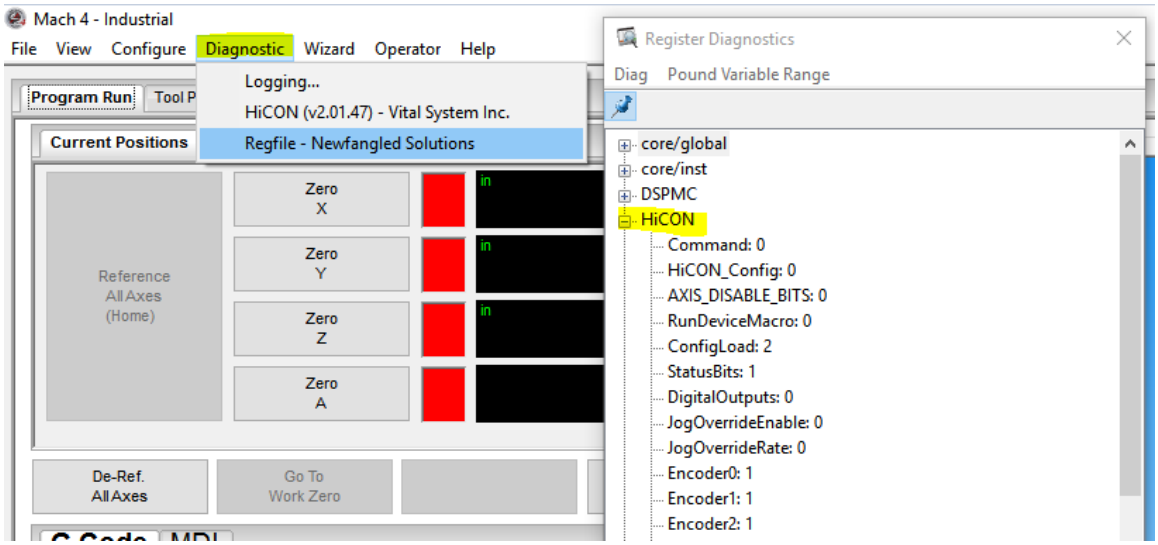
```
mc.mcRegGetHandle(m4Instance, registerPath);
```

Where the register path is '*HiCON/<registerName>*'.

The example below reads Analog Input 0 in order to scale the Feedrate override setting.

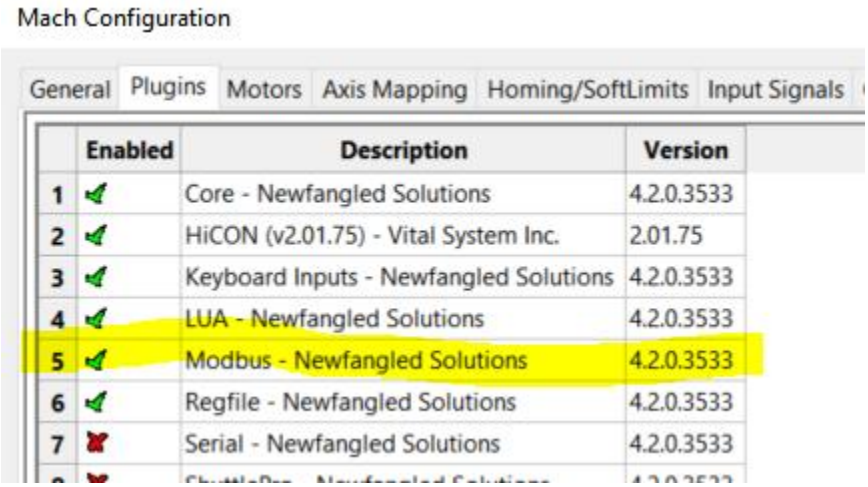
```
inst = mc.mcGetInstance();
regHandle = mc.mcRegGetHandle(inst, 'HiCON/VADC_0');
regValue = mc.mcRegGetValue(regHandle);
mc.mcCntlSetFRO(inst, (regValue * 250) / 3300);
```

Registers may also be viewed and modified via the Regfile Diagnostic Utility in Mach. Use this utility to view the available registers, their names, and current values. Values may be modified by double-clicking a register entry.

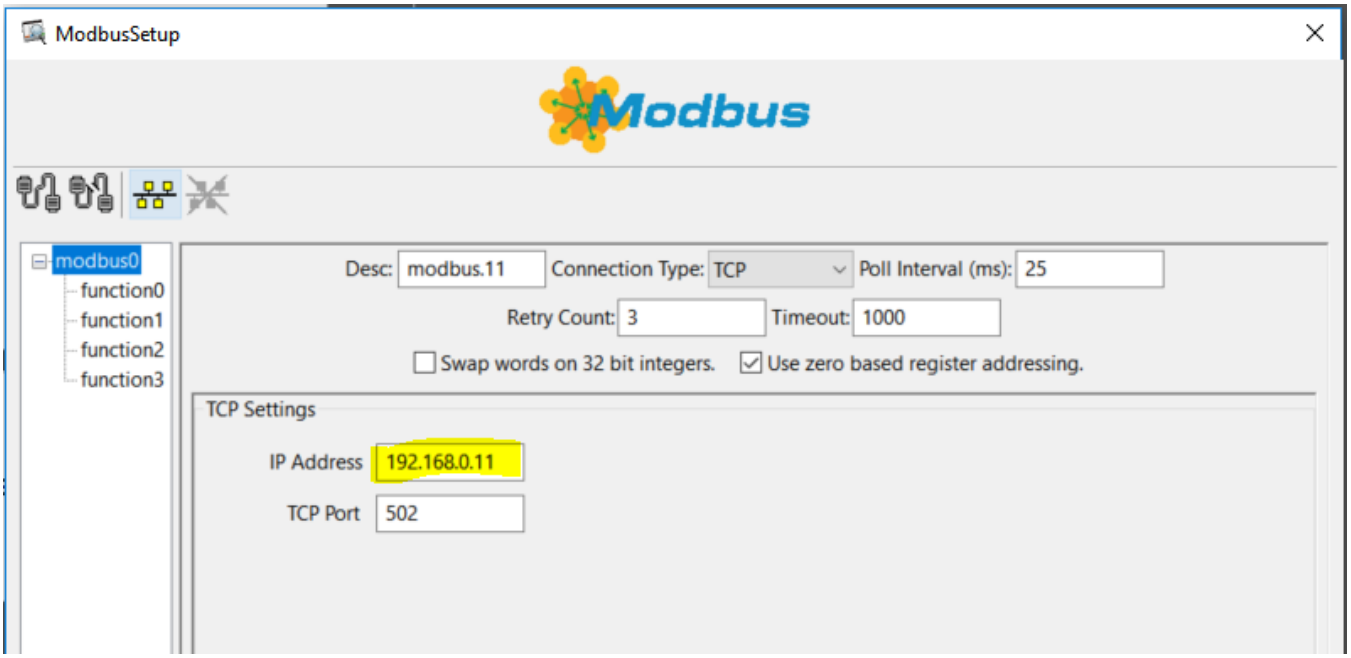


I/O Expansion Using Smart3G cards over Modbus/Tcp

Activate the ModBus plugin in the Mach configuration window and restart Mach.

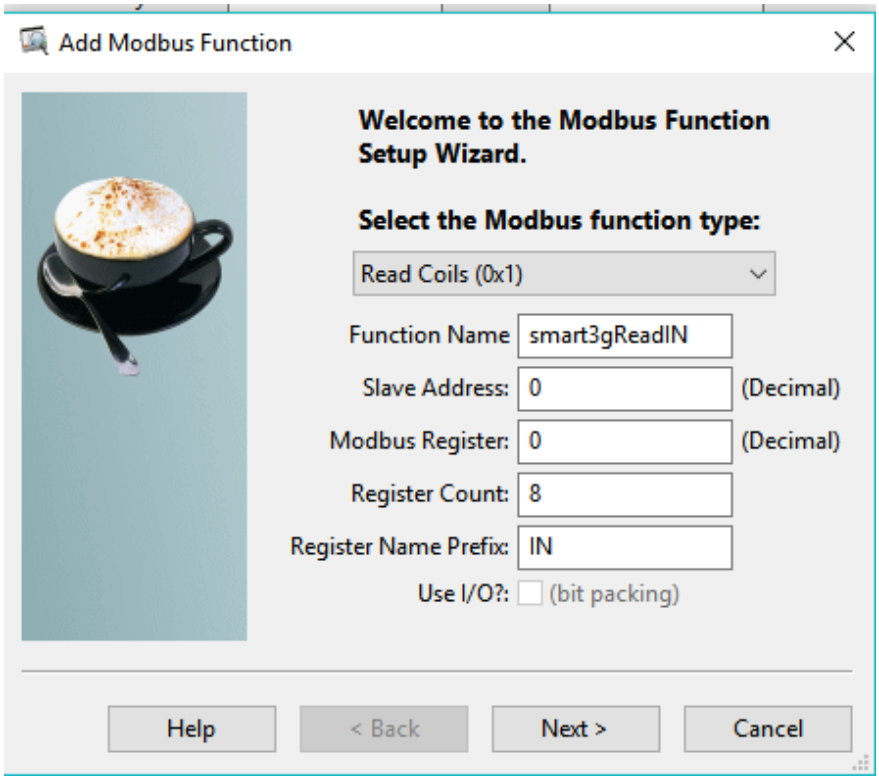


Create a new Modbus connection and configure the settings like in the following image. The last octet in the IP address field will be the value of the rotary switch on the Smart3G card.

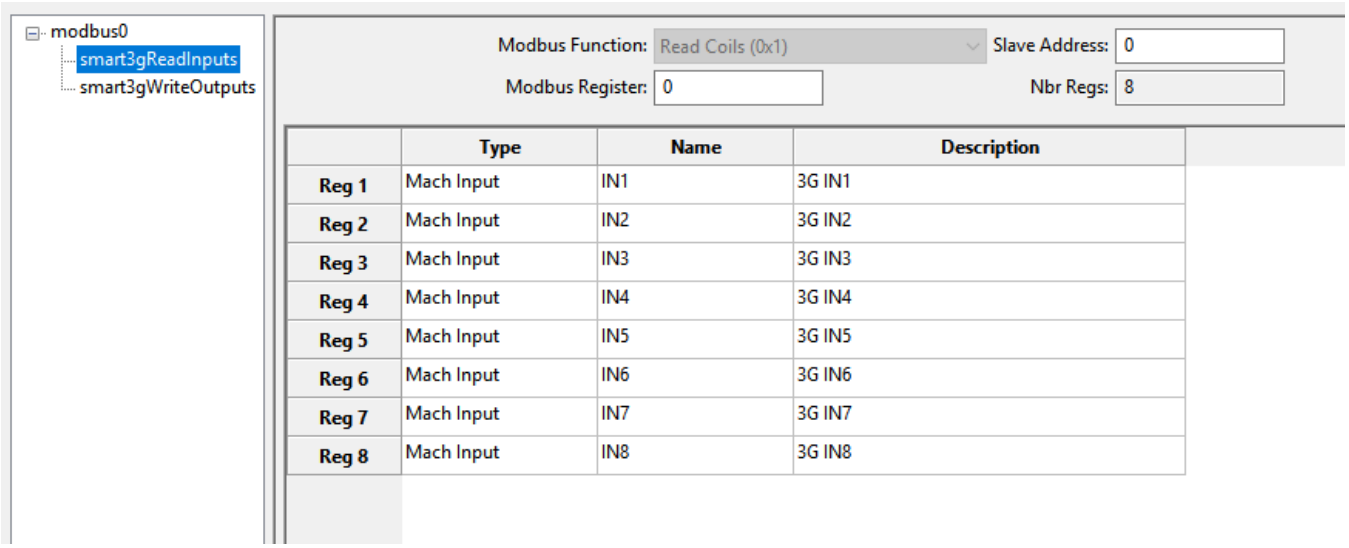


Inputs Setting:

Create a new function for the connection and configure it like the following image.



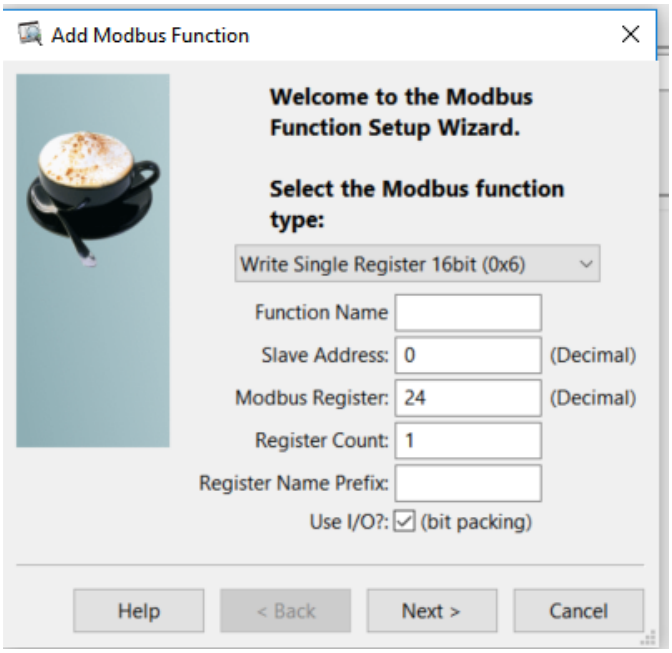
Label the input register bits like in the following image.



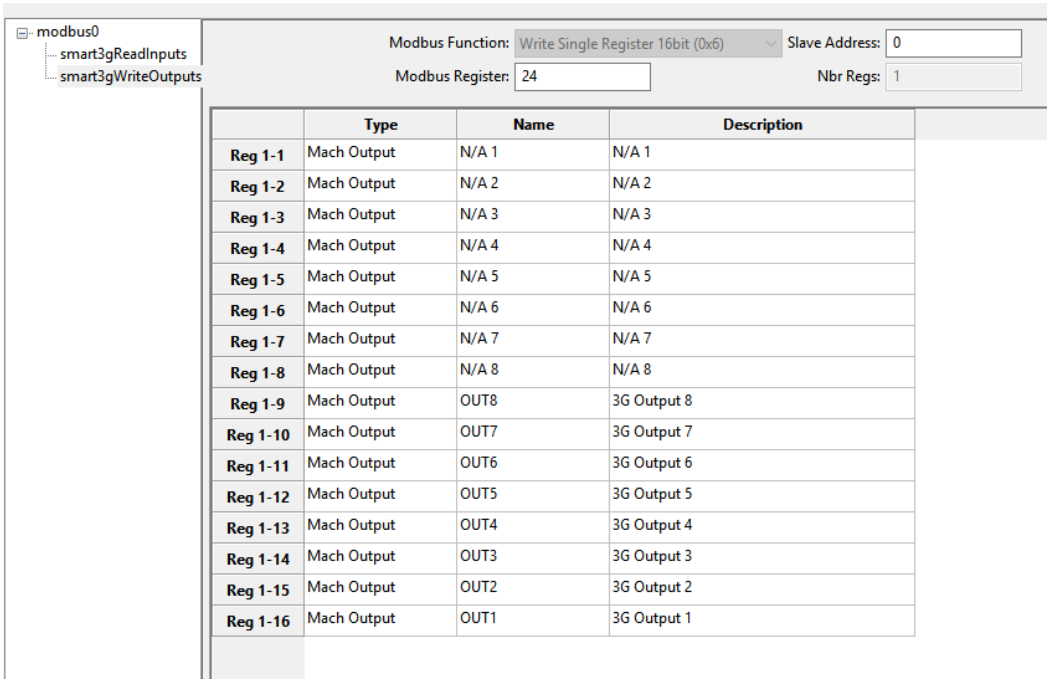
	Type	Name	Description
Reg 1	Mach Input	IN1	3G IN1
Reg 2	Mach Input	IN2	3G IN2
Reg 3	Mach Input	IN3	3G IN3
Reg 4	Mach Input	IN4	3G IN4
Reg 5	Mach Input	IN5	3G IN5
Reg 6	Mach Input	IN6	3G IN6
Reg 7	Mach Input	IN7	3G IN7
Reg 8	Mach Input	IN8	3G IN8

Outputs Setting:

Create a new function for the connection and configure it like the following image.

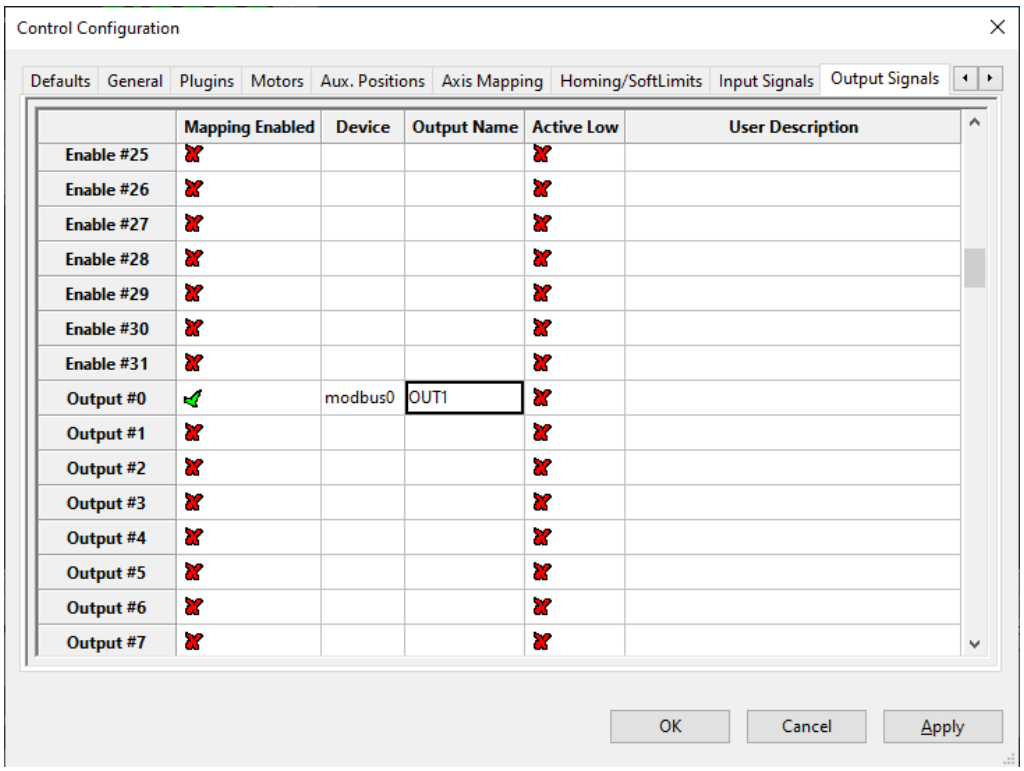


Label the output register bits like in the following image.



	Type	Name	Description
Reg 1-1	Mach Output	N/A 1	N/A 1
Reg 1-2	Mach Output	N/A 2	N/A 2
Reg 1-3	Mach Output	N/A 3	N/A 3
Reg 1-4	Mach Output	N/A 4	N/A 4
Reg 1-5	Mach Output	N/A 5	N/A 5
Reg 1-6	Mach Output	N/A 6	N/A 6
Reg 1-7	Mach Output	N/A 7	N/A 7
Reg 1-8	Mach Output	N/A 8	N/A 8
Reg 1-9	Mach Output	OUT8	3G Output 8
Reg 1-10	Mach Output	OUT7	3G Output 7
Reg 1-11	Mach Output	OUT6	3G Output 6
Reg 1-12	Mach Output	OUT5	3G Output 5
Reg 1-13	Mach Output	OUT4	3G Output 4
Reg 1-14	Mach Output	OUT3	3G Output 3
Reg 1-15	Mach Output	OUT2	3G Output 2
Reg 1-16	Mach Output	OUT1	3G Output 1

In the Input/Output Signal tabs of the control window, map the desired Modbus output/input bits to the desired Mach Signal.



Blink and Connectivity Example Test

You can test the configuration by editing the PLC script. First go to Operator->Edit Screen. Then click the Screen name in the hierarchy on the left (root node), and click the Events button in the property window (thunderbolt icon.) Click PLC Script, then the … button that appears. This will open the PLC script.

The below code will blink the Mach output signal Output #0. Additionally, it will attempt to reconnect to the Modbus device if the connection is lost:

```
local hSig = mc.mcSignalGetHandle(inst, mc.OSIG_OUTPUT0)      --Output mapped in Mach to
                                                              --Modbus device modbus0 OUT1
mc.mcSignalSetState(hSig, testcount & (1 << 2) ~= 0)          --Blink output

local hReg = mc.mcRegGetHandle(inst, "modbus0/smart3gWriteOutputs/rc")      --Check device error state
local devError = mc.mcRegGetValue(hReg)

if (devError ~= 0) then    --Modbus Device is offline, try to reconnect

        if (testcount % 20 == 0) then      --Perform once a second
                mc.mcCntlSetLastError(inst, "Smart3G device disconnected! Attempting to reconnect.")
                hReg = mc.mcRegGetHandle(inst, "mbcntl/command")
                mc.mcRegSetValueString(hReg, "")        --Clear Command Register

        elseif (testcount % 20 == 1) then          --Give time for modbus to react

                hReg = mc.mcRegGetHandle(inst, "mbcntl/command")
                mc.mcRegSetValueString(hReg, "restart modbus0")       --Set Command Register to attempt
                                                                      -- to reconnect

        end
end
```

Insert the code near the very bottom of the PLC Script, right before these lines:

```
------------------------------------------------------
--This is the last thing we do.  So keep it at the end of the script!
machStateOld = machState;
machWasEnabled = machEnabled;
```

If you are polling multiple Modbus devices, you will need to duplicate the above code for each connected device, and use the corresponding device name.